

# Ordinary Differential Equations and Numerical Analysis

**Petr Habala**

FEL, ČVUT Praha

CTU Prague, FEE

<http://math.feld.cvut.cz/habala/>

## Introduction

If someone made a survey among scientists and engineers about which part of mathematics they use the most, two names are bound to come on the top: Differential equations and numerical analysis.

Differential equations can be seen as a language that allows us to capture precisely the relationship between a cause and its consequence. Given that pretty much everything we see is the outcome of some causes, it is no surprise that differential equations feature large in all serious attempts to describe (and understand) this or that particular facet of the world around us, from movement of planets to the weather, to changes in populations, to combustion in engines, to elementary particles.

Unfortunately, they are also extremely hard to solve. When we describe some process using differential equations, then the more interesting it is, the smaller the chance that we will be able to solve those equations. Thus in most cases engineers and scientists turn to computers to give approximate solutions.

However, solving problems on computers brings its own challenges. Sometimes calculations—without making a programming error—provide wrong answers due to limitations of computers. Sometimes the answer is correct, but we would have to wait a few months or years for it. Numerical analysis is a field that derives various ways of calculating things and then studies those ways so that we know as much as we can about their reliability and how long they will take. Not surprisingly, we want the fastest and most reliable algorithms, and numerical analysis tries to make us happy.

Many students take courses on differential equations. Perhaps less students take courses on numerical analysis. Traditionally, these two are studied independently. This book grew out of the author's conviction that there is a benefit in studying differential equations and numerical analysis together. The former cannot be done without the latter in real life applications. On the other hand, it makes studying the latter easier when we see why it is needed. As we will see, most of the topics in numerical analysis are directly related to differential equations.

Thus in this book we always first cover some topic theoretically, which works fine as long as the teacher serves students carefully prepared problems. In those, all polynomials are of the second or third degree and they have nice roots, and all functions can be easily integrated. Then we become spoilsports and stop being this nice (aka real life), immediately run into trouble with our theoretical methods and turn to numerical methods for help.

Thus in the book we have two parallel threads, chapters switch focus. The advantage of this is a mutual synergy, but not surprisingly this comes at a price. It may be a bit harder for the reader to build a cohesive picture of these two theories, as we sometimes start on some topic, then interrupt it, then return to it. The author (naturally) believes that benefits outweigh the costs, but readers who would like to see a whole picture without side trips need not despair. Some chapters are marked with the symbol (A) as in "analytic approach". If the reader reads only these chapters in sequence, he will find a fairly standard exposition of differential equations, similar to those in traditional courses. Similarly, by reading chapters marked with (N) as in "numerical approach", she will get a standard course on numerical analysis. We also added some sections that aim to put the bits of knowledge together and put some system into it for the reader.

There is another aspect in which the book is somewhat different. A typical textbook aims to provide a comprehensive theory. We do not have time for that in this book, so we do not go very deep, but prefer to give a broad overview and entry-level knowledge. This makes sense, as this book is aimed at freshmen and it is based on a one-semester course. The aim of this book is to help the reader in understanding concepts, developing intuition and imagination, and seeing connections between mathematical language and pictures or processes. We do this by including extensive examples, illustrative pictures, and fairytales.

Generally speaking, when the author pictures a student who went through this book, he does not see someone well-versed in differential equations and numerical methods. After all, it would be

naive to expect that by taking such a combined course one can become as good as those who take a specialized course on differential equations or on numerical analysis. Instead, the author sees someone who understands the substance of these two fields, someone who understands concepts and the mathematical language used to work with them. The author believes that every engineer and scientist can benefit from being aware of possibilities of these two fields, and a one-semester course will provide just that. And the author also believes that if the reader one day finds out that a deeper knowledge of a certain topic is needed, then the insight gained in this book will make further study easier.

### Notation

We use mostly a standard notation as known from introductory calculus and linear algebra courses. There are two symbols that should be clarified.

We use  $\approx$  for “almost equality”. This notion does not have any precise meaning, it is used to show approximate values of precise numbers. For instance,  $\pi \approx 3.14$  and  $\sqrt{2} \approx 1.4$  are good examples. They are never used in precise calculations or theory, rather we use them to get a basic feeling for results or situations, for instance if we want to compare positions of some numbers on a real line.

Conversely, the notation  $\sim$  has a precise meaning. For two functions  $f, g$  we write  $f \sim g$  if they are asymptotically equal at infinity, which by definition means that  $\lim_{x \rightarrow \infty} \left( \frac{f(x)}{g(x)} \right) = 1$ . For practical purposes this means that the values of these two functions are the same as we approach infinity, and we use this notion to isolate the dominating term in a formula. For instance, we could say that  $x^2 + 13x \sim x^2$  as  $x \rightarrow \infty$ , and the reader can check that a typical calculator cannot see any difference between these two formulas for  $x$  as small as  $10^{15}$ .

We included a section on this notion in our background chapter, see 38b.

## Table of Contents

- 0. Introduction
- 1.(A) Meeting differential equations
- 2.(N) Meeting numerical mathematics
- 3.(N) Errors in calculations
- 4.(N) Approximating derivative
- 5.(N) Approximating integral
- 6.(A) Formal introduction to ordinary differential equations
- 7.(A) Separable differential equations
- 8.(A) Analyzing solutions
- 9.(A) First order linear ODEs (variation)
- 10.(A) Applications
- 11.(N) Euler method (introduction to numerical view of ODEs)
- 12.(N) Basic numerical methods for ODEs
- 13.(N) More numerical methods for solving ODEs
- 15.(A) (Homogeneous) linear ordinary differential equations
- 16.(A) Non-homogeneous linear ODEs
- 17.(A) Oscillations and other applications
- 18.(N) Numerical methods for higher order ODEs
- 19.(N) Finding roots numerically
- 20.(N) Finding fixed points numerically
- 22.(N) Elimination and related methods for matrices
- 22.(N) Solving systems of linear equations by elimination
- 24.(N) Error in matrix calculations, condition number
- 25.(N) Solving systems of linear equations by iteration
- 26.(A) (Homogeneous) systems of linear ODEs
- 27.(A) Non-homogeneous systems of linear ODEs
- 31.(N) Finding eigenvalues and eigenvectors
- 34. Transformations
- 38. Appendix: Taylor expansion, asymptotic growth
- 39. Appendix: Proof of Picard's theorem

## 1. Meeting differential equations

In this chapter we will get familiar with the essence of differential equations.

Informally, a differential equation is an equation that features a function as its unknown, and not only that, it also features some of its derivatives. If the function has more variables, then the derivatives are in fact partial derivatives. The resulting equation is then called a partial differential equation (PDE) and there are whole libraries of books about them. Here we prefer to talk about **ordinary differential equations** (ODE) that feature an unknown function of one variable and ordinary derivatives.

A typical ODE therefore sports two letters—one letter denotes the function and the other denotes its variable. Traditionally we use  $y(x)$  for the function and variable, but other choices are not uncommon; in particular, if the variable is time then we often use  $t$ . In fact, thinking of  $x$  as time is a good way to appreciate workings of differential equations, for instance the function  $y(x)$  may describe how temperature at my working desk changes throughout the day. But  $x$  may also stand for position and  $y$  may describe how temperature changes as I am walking in front of the blackboard from left to right.

The  $x$  (or  $t$  or whatever) is thus a working variable and as such it is not crucial. It just allows us to check on validity of our equation at different “locations”. Typically, a differential equation captures some natural law, and when we investigate some process, that law should be valid for all times or at all places where this process takes place.

To see an example I just dreamed up this differential equation:  $y(x) + x = \sin(x + y''(x))$ . Since  $x$  is a dummy variable, we could also write the same equation as  $y(t) + t = \sin(t + y''(t))$ . Mathematically it is the same, but the name of the variable may have its meaning in an application from which this equation (perhaps) came, so we change the variable name only if there is a reason, which pretty much never happens.

Actually, people rarely write differential equations in this way. Traditionally we skip the variable when writing the function  $y$ , so most people would write just  $y + x = \sin(x + y'')$ . In fact, this is still not the most typical example, we prefer our differential equations in the form where the highest derivative is isolated. It is quite obvious that the above equation cannot be rewritten into the form  $y'' = \dots$ , because the sine function is not invertible in general. Differential equations are very tough, probably the hardest equations to solve, and not having the highest derivative isolated makes it even harder.

So here's a better example:  $y'' = e^{x+y} - (y')^2$ . This time the highest derivative is nicely isolated, which can be very helpful. In particular, we can easily spot the degree of this highest derivative; it plays an important role and we call it the order of the equation. Both of our examples so far were ODEs of order 2.

Here is another example:  $y' = -xy^2$ . This is an ordinary differential equation of order 1 and we claim that the function  $y(x) = \frac{2}{x^2}$  is its solution. What does it mean? In general, a solution to some equation is an object that, when substituted into that equation, creates a true situation.

However, now the suspected solution is a function, and when we substitute it into our differential equation, then both sides become functions. We know that equality of functions is always dependent on our choice of sets on which we compare those functions. For instance, equality of functions  $x = |x|$  is true when viewed on  $(0, \infty)$ , but not true when viewed on  $(-1, 1)$ . Thus, when talking about a solution of a differential equation, we also have to say on which set this equation is considered.

This set is usually easily determined when we substitute the suspected function into our equation. Let's see:

$$\text{LHS: } y' = [2x^{-2}]' = -4x^{-3} = -\frac{4}{x^3},$$

$$\text{RHS: } -xy^2 = -x\left(\frac{2}{x^2}\right)^2 = -\frac{4}{x^3}.$$

We see that the left hand-side and the right hand-side agree for all  $x \neq 0$ , which brings us to another important point. Differential equations describe rules that govern some process. Imagining

for the moment that the variable  $x$  is time, the fact that the solution makes no sense at  $x = 0$  means that we lost all information at that point and the process was free to do whatever it wanted (at least as far as our equation is concerned). It therefore makes no sense trying to find a solution that spans this gap.

Consequently, when solving differential equations, we always consider only solutions on intervals. In our particular case we can choose any interval that does not include zero, and because we want to give as complete answer as possible, there are two natural candidates. Thus we would say that we actually found two solutions, one is  $y(x) = \frac{2}{x^2}$  for  $x \in (-\infty, 0)$  and the other is  $y(x) = \frac{2}{x^2}$  for  $x \in (0, \infty)$ .

What we just learned? Differential equations have functions as solutions, and a specification of a solution is not complete without saying on which interval it is valid.

What kind of information does such an ODE carry? In many sciences (in particular in physics) we study “situations”, we focus on some part of the world and try to understand what is going on. Typically we describe this part using some quantities (position, temperature, electric current etc.) and they typically change. One fruitful approach is to freeze time and inspect the situation, ask what factors are in play and what kind of change they are trying to achieve. Since change means derivative, we arrive at a differential equation.

In our equation we may imagine that the right hand side  $-xy^2$  represents some influence of the environment that is trying to change the quantity  $y$ . The fact that the influence features  $y$  shows that this quantity somehow interacts with itself, which is fairly typical.

This influence may affect  $y$  in various ways. Sometimes it wants to change its value directly, that is, it specifies the rate at which  $y$  is supposed to change. Since the rate of change is derivative, we arrived at our equation. But this influence may act differently, for instance it may attempt to change the rate at which  $y$  is already changing, so then it would actually be the acceleration and the corresponding differential equation would be  $y'' = -2xy^2$ , a second-order ODE.

Now figuring this all out is a job of physicists and other scientists and engineers, what we take from this tale is that the form where the highest derivative is isolated seems to appear naturally. When mathematicians develop theories for solving various types of equations, they sometimes prefer different arrangements of ODEs, as we will see later.

**Example 1.a:** Here we look closer at an extremely simple differential equation:  $y' = 80$ .

It is an ODE of order 1, and any solution  $y(x)$  must satisfy  $y' = 80$ . Obviously, the answer should be  $y(t) = 80t + C$  for any constant  $C$ . Let's ask some good questions.

First, note that in the equation the working variable was not specified. This happens, and I chose  $t$  for it as it will fit nicely with a story I am about to tell.

Second, the answer is not quite correct as we should always specify where the solution is valid. In our example there is no trouble, hence no reason to rule out any value of  $t$ . We therefore write  $y(t) = 80t + C$ ,  $t \in \mathbb{R}$ . Now this is a correct answer.

Third, what about the  $C$ ? It seems that our equation has infinitely many solutions. In fact this is to be expected. A typical ODE describes some rule, a natural law, and a natural law by itself never forces just one outcome, but allows for many outcomes, depending on other conditions. Which brings us to the story we promised.

One possible representation of our equation is this: Imagine trains travelling along a railway. This greatly simplifies keeping track of their positions, because we only need to know how far along the track they are, measured from some chosen point, that is, just one variable is needed. In fact, railway people need this information, so they already set up the situation for us, there are mileposts along railways. Now consider one particular car. Its position depends on time and therefore it can be captured in the form of a function, call it  $y(t)$ . Physicists tell us that derivative of position is speed, so if the car travels along the railway at 80, its position must satisfy the equation

$y' = 80$ . Here we assume that time is measured in hours, for instance starting at midnight, and position in kilometres (we are not bothered in the least by the fact that mileposts show kilometers in most countries).

Thus we may imagine that our differential equation describes a law that governs all trains that travel along a particular railway at speed 80. It is quite obvious that there can be many possible trains like that, because they can drive at different places on the railway. Thus the infinite number of solutions makes sense.

Now imagine that we start at eight in the morning, just at the milepost saying 213. Mathematically, our positional function satisfies  $y(8) = 213$ . If we also assume that our velocity is always 80, then we no longer have any freedom of position, it is uniquely determined. And indeed, among functions of the form  $y(t) = 80t + C$  there is only one that satisfies  $y(8) = 213$ , it is the function  $y_p(t) = 80(t - 8) + 213 = 80t - 427$ .

△

This example has shown us several important things. A typical differential equation has infinitely many solutions. Often we can capture them all in one formula with parameter(s), this is then called a **general solution** and it is very desirable to have it. Traditionally the parameters are denoted  $C, D, \dots$  and they can have any (real) value.

Common sense tells us that if we also add some specific requirements into the story, then with a bit of luck we can adjust those parameters to obtain a solution that satisfies those requirements. Very often (and naturally) we specify what happens at the beginning, this is called **initial conditions**. When we use conditions to choose just one specific solution of the equation, we call it the **particular solution**.

Experience tells us that when we have a certain number of parameters, then we need exactly the same number of requirements to determine them uniquely. Now this is not a general fact, but as we will see, it does work with the nicer differential equations that we will study here.

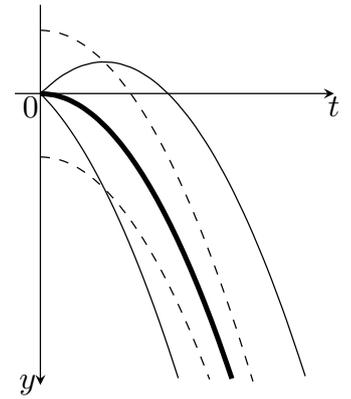
We learned about as much as we could from this example, let's move to another one.

**Example 1.b:** Imagine what happens to some object that we dropped off our table. Experience suggests that it starts falling down, gravity being the culprit. To simplify things we will assume that it moves straight down, so its position with respect to time can be described by its elevation  $y(t)$  with respect to some chosen reference level. In this example it will be convenient to use the table from which we dropped that thing as level zero and measure the distance downward, so when the thing falls down,  $y(t)$  increases and is positive (we like positive numbers). Since things fall pretty fast, we will measure time in seconds and displacement in meters.

To make our life easier even more, we will ignore all other influences (air resistance, variable gravitational field, Coriolis force etc.), and just worry about the gravity force described by the famous  $g$  that happens to be about 9.8 m/s<sup>2</sup> where I sit now. Physics tells us that gravity influences acceleration, the corresponding differential equation is then  $y'' = 9.8$ . Note that we did not have to adjust signs, as acceleration points in the same direction as our axis  $y$ , that is, downward. We obtained a nice ODE of order 2. In fact, this differential equation is the law of gravity as it applies to objects near the surface of the Earth.

To solve this equation we integrate twice, the first time we get  $y'(t) = 9.8t + C$ , the second time we get  $y(t) = \frac{1}{2}9.8t^2 + Ct + D$ ,  $t \in \mathbb{R}$ . This is a general solution of our ODE.

As expected, we obtained infinitely many solutions, this time with two parameters that we can use to narrow things down to one specific situation. Note an interesting coincidence. In the previous example we had an ODE of order 1 and one parameter, this time we have order 2 and two parameters. This is not an accident but a general rule (at least for nicer equations). The picture on the right shows some solutions, we can drop things from different elevations and we can give them different initial velocities up or down.



To obtain a specific particular solution we need two pieces of information. Let's start with initial conditions. We may start measuring time exactly when we drop the thing, and at the beginning it is at the table level, so  $y(0) = 0$ . One more condition needed, we will use it to describe how we actually dropped the things.

Did we just release it, or did we actually hurl it down in anger (or up in joy)? This information can be expressed as the starting velocity of our object. Let's say that we simply released it, so  $y'(0) = 0$ . We just chose the situation marked as the thick curve in the picture.

What we obtained is one of the most typical situations: We have a differential equation and conditions that determine how the process that we study started.

$$\begin{aligned}y'' &= 9.8 \\y(0) &= 0 \\y'(0) &= 0\end{aligned}$$

This setup is called the **Initial Value Problem** (IVP).

How do we handle this? We already found a general solution  $y(t) = \frac{1}{2}9.8t^2 + Ct + D$  and we know its derivative  $y'(t) = 9.8t + C$ . Thus we can substitute these formulas for  $y$  in the initial conditions:

$$\begin{aligned}y(0) = 0 &\implies \frac{1}{2}9.8 \cdot 0^2 + C \cdot 0 + D = 0 \implies D = 0 \\y'(0) = 0 &\implies 9.8 \cdot 0 + C = 0 \implies C = 0\end{aligned}$$

We see that our conditions are met by the particular solution  $y(t) = \frac{1}{2}9.8t^2$ ,  $t \in \mathbb{R}$ .

Note that this is a mathematical solution to a mathematical problem, then it really makes sense to take real numbers for time. However, in the specific situation that our initial value problem describes, that falling object was not governed by the gravity equation before we released it, so the equation cannot be applied to negative times. This is fairly typical; in applications we usually start measuring time when the process we are interested in starts, and thus it does not make much sense going back in time. People in applications therefore often prefer “practical” solutions to mathematical ones; in our practical problem we would prefer the solution  $y(t) = \frac{1}{2}9.8t^2$ ,  $t \geq 0$ .

Sometimes the person who gave us this equation to solve makes things clear by restricting time already in the question like this: “ $y'' = 9.8$  for  $t \geq 0$ ”.

Initial conditions are the most popular type of problem, but sometimes it is natural to also consider other types of conditions. For instance, we may observe that we dropped the thing at time 0, so  $y(0) = 0$ , and half-a-second later it landed on the floor, making a drop of 80 cm. Mathematically, we also require that  $y(0.5) = 0.8$ .

This time our conditions specify what happens at the beginning and at the end, they fall into a broader category called **boundary conditions**. Again, knowing a general solution makes our life easy:

$$\begin{aligned}y(0) = 0 &\implies \frac{1}{2}9.8 \cdot 0^2 + C \cdot 0 + D = 0 \implies D = 0 \\y(0.5) = 0.8 &\implies \frac{1}{2}9.8 \cdot (0.5)^2 + C \cdot 0.5 + D = 0.8 \implies 0.125 \cdot 9.8 + 0.5C + D = 0.8\end{aligned}$$

We obtain  $D = 0$ ,  $C = -0.85$ . The corresponding particular solution is  $y(t) = \frac{1}{2}9.8t^2 - 0.85t$ ,

$t \in \mathbb{R}$ . It seems that this time we flipped the thing up a bit when releasing it, since we have  $y'(0) = -0.85$ , that is, a negative initial velocity (recall that  $y$  points down).

Again, this is a purely mathematical solution. Practical answer should take into account that we do not really know what happened before and after our conditions, so it makes also sense to answer like this:  $y(t) = \frac{1}{2}9.8t^2 - 0.85t$ ,  $t \in [0, 0.5]$ .

△

Note that we did not really need any theory for solving this problem, just common sense and a working knowledge of calculus. We will learn some theories later, but they will always use a good deal of common sense and it is worth trying to see motivations and meanings of the procedures that we develop here.

Note also that our differential equation did not feature the working variable. This is natural. As the stone falls, it does not care whether our watch is set to the summer savings time or not. Consequently, one can expect that differential equations that describe natural phenomena should not feature the independent variable. Such equations are called autonomous and we will look closer at them in section 8a.

Given a differential equation, there are several things we would like to have. Foremost among them is a general solution given by a formula with parameters. This is called an **analytic solution** and we say that we solved the equation analytically. Then we can easily determine particular solutions according to suitable additional requirements. We can also do some analysis of the solutions, typical question might be:

- When do solutions increase and decrease?
- How do solutions behave if we let the independent variable go to infinity?
- How does the behaviour of solutions depend on our choice of initial conditions?

In particular, could it happen that we change an initial condition just a little bit and it results in a radical change in behaviour of solution? Such sensitivity is usually not welcome in applications.

As you can imagine, such information can be very useful for someone who is trying to understand some process. We will address these questions in chapters that follow. There is an interesting angle to all these questions. It often happens with real life equations that we actually cannot find a general solution, in other words, we cannot solve them analytically. Then it would be really useful if we could somehow get answers to our questions without knowing a general solution, that is, just from the equation itself. We will look into this in chapter 8.

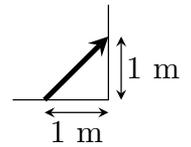
We have met some interesting concepts here. If we want to address them mathematically, we first have to dress them up in mathematical language, which we do in chapter 6.

## 2. Meeting numerical analysis

Computing is present all around us, our civilization depends on it for its survival. How good are we at it? At elementary school we learned how to perform four basic operations by hand, usually involving integers and/or fractions. Using these we can also handle powers as repeated multiplications – that is, when exponents are positive integers. We never really progressed past these accomplishments, and remarkably enough, this is also the extent of abilities one finds in computer processors.

Now you may argue that we can work with other objects, too, for instance square roots, exponentials and logarithms, in short we were taught to work in the world of real numbers. However, there is a catch: What are these real numbers?

Imagine a familiar situation: You need a beam to prop up a fence, it should go in an oblique way, one end on the ground 1 meter from the fence, the other end attached to the fence 1 m above the ground. Mathematically, we are talking the hypotenuse of a right-angle triangle with legs of length 1, so the answer is obvious: The beam should be  $\sqrt{2}$  meters long. From a mathematical point of view we have the problem licked. Now imagine coming up to a carpenter, asking for a beam whose length is  $\sqrt{2}$ . Well, how long is  $\sqrt{2}$ , asks he? There is no such number on his ruler.



This is, in a nutshell, the difference between theoretical mathematics and real-life mathematics. People out there need concrete answers if they are to build something. Numerical analysis is a branch of mathematics whose purpose is to provide practical answers. Sometimes it just translates theoretical answers into “real” ones, but more often it actually has to provide answers to problems that theory cannot solve. For instance, note that we obtained our beam length by solving the equation  $x^2 = 2$ . We were able to solve it analytically and expressed its solution using a formula,  $x = \sqrt{2}$ . Just a small change with turn our equation into another,  $x^5 + 2x^2 = 2$ , and suddenly we are unable to provide any solution analytically. But there is a solution and numerical mathematics can tell us about it.

Let’s go back to our carpenter. Of course, we can punch a few buttons on our calculator and tell the carpenter to look up the 141.4 cm mark on his ruler, but this brings us to a very good question: Where did the calculator get this number, given that we claimed above that it can only do the same basic algebraic operations as our mammal brains? And the answer is that it uses some approach developed by numerical analysis.

Let’s make an important observation. We know that 1.414 is not really the same as  $\sqrt{2}$ , but our carpenter was happy with our answer, because when cutting with a saw one is lucky to get the right length within half a millimeter. If we needed a strut for an airplane, we would definitely need to be more precise. And when a calculator wants to show us the value of  $\sqrt{2}$ , it has to work even more, because a typical calculator has at least 10 digits precision. The moral of this story is that numerical calculations are always done to a specified precision. The customer always has to specify what error is considered acceptable, and numerical analysis has to offer tools that allow us to guarantee that the answer is good enough.

**Example 2.a:** We will provide a decimal value for the number  $e^\pi$ .

In a typical calculus course students learn that  $e^c = \lim_{n \rightarrow \infty} \left(1 + \frac{c}{n}\right)^n$ . This can be thought of as an algorithm for getting  $e^c$ : We substitute larger and larger numbers for  $n$  and analyze what the outcomes are doing. In order to get  $e^c$  precisely we would have to go through an infinite computation, but that would take too long, so we simply put in a large number and see what happens.

We want  $e^\pi$ , so I asked my calculator to substitute  $n = 10^5$  in the formula  $\left(1 + \frac{\pi}{n}\right)^n$  and the answer I obtained is 23.1395587... Now what should we think of this number? We obviously cannot hope that it is  $e^\pi$ , but is it a reasonably good approximation? Given that  $e^\pi = 23.1406926...$ , we see that we have the first three digits correct. However, we should actually imagine that we do

not know the real  $e^\pi$  yet (otherwise we would not need to approximate it). How can we judge the quality of our result if we do not know the precise answer?

This is a question that permeates numerical analysis, and people tackle it by analyzing the chosen procedure and estimating what error might have happened. We typically investigate the worst case scenario, and then try to make it as small as needed by the customer, which makes sure that our answer is good enough.

We are not ready for such an analysis yet, but we have a nice opportunity to look at sources of errors.

One source is obvious: Instead of “infinity” we substituted some other number. This is a common situation, most numerical methods do not calculate exactly what we need, but something else instead. The error caused by this can be called the **error of method**, but it is commonly called the **truncation method**. Why? A typical numerical method is based on some theoretical approach that would yield a precise answer after infinitely many steps, but we cut the procedure short, that is, we truncate our calculations. And that is exactly what we did with our limit formula here.

Every time numerical analysis invents some procedure, the first task is to deduce some upper estimate for the error of the method. This error estimate probably features some information about numbers involved in the calculation (like the  $\pi$  here, perhaps the error would be different when approximating  $e^{13}$ ), but definitely we can expect some *indicator of quality*. In our example  $n$  tells us how close to infinity we got, so this should be the indicator. We expect that for larger  $n$  we get better approximations and this is actually true.

Let's try it. We substitute  $n = 10^7$  into  $(1 + \frac{\pi}{n})^n$  and obtain 23.1408559... Now five digits are correct, it seems to work. Can we get even closer? We substitute  $n = 10^9$  and get 23.1038667... Funny, this is worse than before. How about  $n = 10^{11}$ ? We obtain 20.0855369... Now this is weird, we are actually getting further away from the correct answer instead of getting closer to it. How could it be?

The culprit here is the other kind of error we have to worry about, the **numerical error**. We have to worry about it every time we do calculations not on paper, but in a computer or a calculator. These machines have only limited memory, and therefore they cannot store all real numbers precisely, instead they round them. Since engineering calculations that could be entirely done using reasonably small integers only are rare, in most real-life calculations we have to expect that our computer (or calculator) will keep forgetting parts of our numbers. This kind of error is often called **round-off error**.

We can see it in our calculation right away. Our formula uses  $\pi$ , which is an irrational number, in other words no calculator or computer can keep it precisely. My calculator happens to work in 13 digit precision, so right from the start we introduced some error. Which brings us to another interesting source of errors. In many real-life calculations we use data that come from some experiments. Since measurements are never precise, we again start such calculations already with imprecise data. One can talk of **error on input**, but this distinction is not really important. We simply have a situation that we do some calculations with numbers that have some errors, and it does not matter much why this error came or even when, those mistakes might well appeared in the middle of the calculation due to rounding.

The question we really worry about is this: How does our procedure react to imprecise data? Will it throw the result completely off? This is in particular worrying when it comes to iterative methods, where the same calculation is repeated many times over and the results of one step are used as data for the next step. Thus the errors have an opportunity to grow, they are joined by new errors, this can easily get out of hand. We therefore prefer numerical procedures that can keep errors under control. Such calculations are called **numerically stable**. We investigate this by analyzing what happens to errors as they move through our calculations. This topic is called **error propagation** and we will dedicate a special chapter to it.

To sum it up, numerical analysis involves three important steps. First we deduce some procedure for finding an approximate solution of a certain type of problem. The quality of this numerical method is then judged by two aspects, the error of this method and its numerical stability. Sometimes we also have to worry about a third aspect, because even the most reliable approximation is of no use if we get it several years too late. We will also sometimes look at **computational complexity**.

Let us return to the problem of determining  $e = e^\pi$ . Our procedure is obviously not suitable as it has problem with numerical errors. Is there an alternative? In calculus we learned about Taylor polynomial. In particular, we know that

$$e^x = 1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3 + \frac{1}{4!}x^4 + \dots$$

Of course we cannot really do infinite sums on computers, but they can be easily persuaded to make very long sums. We want to estimate  $e = e^\pi$ , so we choose some cutoff index  $N$  and ask nicely for the value of  $\sum_{k=0}^N \frac{\pi^k}{k!}$ , hoping that the answer will be close to  $e^\pi$ .

First I tried it for  $N = 6$  (seven terms) by hand on my calculator, obtaining

$$1 + \pi + \frac{\pi^2}{2} + \frac{\pi^3}{6} + \frac{\pi^4}{24} + \frac{\pi^5}{120} + \frac{\pi^6}{720} = 22.1882466\dots$$

Not very good. Not having patience for this, I fired up my computer, switched to precision of 12 digits (just like my calculator so that we have a comparison), programmed a little loop and asked it to go up to  $N = 13$ . The result was 23.1400945..., which is much better, but still worse than our previous approach. How about  $N = 20$ ? The result is 23.1406926..., which is much better than our previous approach ever got.

The series approach handles errors quite well and it is a much better approach than the limit one. On the other hand, to get very good precision we would have to add many terms, so there may be some concern that we would have to wait for our answer too long. Fortunately, one can show that the **computational complexity** of this calculation is not too bad and with modern fast computers we need not worry. However, there are problems where this would be a big concern, we will see it when we get to matrix calculations

△

Having read this chapter, you should have a fairly good idea what to expect in numerical parts of this book. We look at a certain type of problem and develop some methods for solving it approximately. In order to compare methods and judge them we will then study the error of each method, their numerical stability, and sometimes also their computational complexity.

Our analysis will usually draw on our knowledge of calculus, Taylor expansion being one of the most popular tools. If you do not feel comfortable with it, you can look at chapter 38 at the end.

### 3. Errors in calculations, numerical stability

In this chapter we will focus on what happens to errors once they enter our calculation, and then also look closer at how computers handle numbers. In order to do it properly we have to first specify what we actually mean by an error.

#### 3a. Absolute and relative error

The setup is very simple. We have some value  $x$  that we want/need to know, but instead of this *precise value* we've got an *approximate value*  $\hat{x}$ . When we talk of an error, it is natural to look at the difference  $|x - \hat{x}|$ . Some authors take this expression as the main object and study it, but a more common approach is to work just with the difference, without the absolute value, to make analysis easier. Thus for us, the error will be just  $x - \hat{x}$ .

**Definition 3a.1.**

Let  $x$  be a number and  $\hat{x}$  its estimate. Then we define the **absolute error** of this estimate as  $E_x = x - \hat{x}$ .

We then use  $|E_x|$  to see how large this error is. Note that it would be also possible to take  $\hat{x} - x$  in the definition and some authors do, so be careful when reading books. We will stick with our definition.

Having this definition, the precise value can be obtained as “estimate plus error”,  $x = \hat{x} + E_x$ . This looks nice, but there is a fatal flaw: To determine the absolute error we actually need to know the precise value  $x$ , and if we knew it, we would have no need for some approximate value and error.

We therefore change our aim, instead of precise information we will try to provide estimates for errors. Intuitively, by an **estimate** of an absolute error we mean some number  $e_x$  that is guaranteed to satisfy  $|E_x| \leq e_x$ . This approach is often used in engineering. Note that if we find an estimate like that, we can estimate as follows:

$$|x - \hat{x}| \leq e_x \implies -e_x \leq x - \hat{x} \leq e_x \implies \hat{x} - e_x \leq x \leq \hat{x} + e_x.$$

That is, using our approximation  $\hat{x}$  and the error estimate we obtain the interval  $[\hat{x} - e_x, \hat{x} + e_x]$  where the true value is guaranteed to be. In engineering we often write this as  $x = \hat{x} \pm e_x$ . It is required that the approximation  $\hat{x}$  and estimate of error  $e_x$  use the same units so that they are comparable at the first glance.

**Example 3a.a:** Imagine a familiar situation: We want to know how long a stick is, and we use a typical school ruler marked in centimeters and millimeters.



In this situation we would probably say that the end of the stick is closer to 25.3 than to 25.4, so the answer is 25.3 cm. We also see that when finding the closest millimeter mark, we cannot make a mistake larger than 0.5 mm, so this is a nice estimate of absolute error,  $|E_x| \leq 0.5$  mm. We can therefore write our answer like this:

$$l = 25.3 \pm 0.05 \text{ cm.}$$

As required, we converted our error estimate to the same unit as the length. We could have also written  $253 \pm 0.5$  mm.

We will return to this example shortly.

△

Absolute error can be very useful in many situations. Often we see it when we determine some quantity by measuring, as every measuring device has some inherent limitation and this limitation

is typically in the form of absolute error, as we just saw. Also in manufacturing, tolerances there are usually given as an absolute error given that they typically stem from limitations of tools. For instance, when cutting wood with a saw, it is not easy to make the cut precisely, but one can usually manage to get it right within about half a millimeter, regardless how large the piece is. This would be an absolute error.

When we have an estimate of our error, we need to judge how serious is it. However, the absolute error is not very helpful in this. Imagine that you are trying to optimize your daily schedule and need to know the distance between your bed and the bus station that takes you to school. In such a situation, precision up to 0.5 mm is needlessly high, we will happily accept even errors as large as several centimeters. Given that precision costs money, we'd better use a different tool. On the third hand, if we wanted to measure the thickness of a hair, then the answer "0.3 ± 0.5 mm" is pretty much useless, given that the error is larger than the value (it seems we can even have hairs of negative thickness).

This shows that judging seriousness of an absolute error requires comparing it to the value we expect to get. In mathematics we do such comparison by division.

**Definition 3a.2.**

Let  $x$  be a non-zero number and  $\hat{x}$  its estimate. Then we define the **relative error** of this estimate as  $\varepsilon_x = \frac{|E_x|}{|x|}$ .

For  $x = 0$  relative error remains undefined.

We have the usual problem, in practical problems we do not know data needed to determine this error. We will therefore be happy with some estimate for the relative error. In fact, estimates are all a practical engineer needs, because using them one can control the error from above, we do not mind if it actually happens to be smaller. However, in order to get those estimates we will need to investigate the precise errors theoretically, so they have their use, although we rarely know them in practice.

What kind of information does the relative error carry? Division is related to proportion, but the interpretation is a bit harder owing to the fact that we have the true value  $x$  in the denominator. Still, we can see something right away. If the relative error is 1 or more, then the absolute error is at least as large as the thing we are supposed to determine. For instance, when  $\varepsilon_x = 1$ , then our measurement could have been anything between 0 and  $2x$ . This is pretty much worthless. We therefore expect to see relative errors smaller than one, preferably much smaller than one.

**Example 3a.b:** In the example 3a.a we obtained the answer  $\hat{x} = 25.3$  with error  $E_x$  bounded by 0.5. Thus we obtain a bound for the relative error

$$\varepsilon_x = \frac{|E_x|}{|x|} \leq \frac{0.5}{25.3} = 0.0197\dots \approx 0.02.$$

We can also look at the result  $25.3 \pm 0.05$  in another way: the largest possible length is 25.35 and the smallest is 25.25. We see the uncertainty of our result starts at the third digit, which is about a hundredth of our result. By a remarkable coincidence, our relative error is also two in a hundred. This is worth investigating.

△

We will now explore the situation when we are trying to find the number  $x = 31642$ , instead producing some approximation  $\hat{x}$  with relative error bounded by  $0.01 = \frac{1}{100}$ . Note that we have  $|E_x| = |x| \cdot \varepsilon_x$ , then the largest possible absolute error is  $35742 \cdot \frac{1}{100} = 357.42$ . This means that our approximation falls into the range  $[31284.58, 31999.42]$ . When we compare this with the precise value, we see that our guess correctly found the first two digits. This makes sense, dividing by a

hundred means that we move the decimal point to the left by two places, so the digit that was originally first now appears at the place of the third digit and influences it.

Now imagine that we have a better bound,  $\varepsilon_x \leq \frac{1}{1000}$ . This time the error is at most 31.642 and our estimate is found in the range [31510.3358, 31673.642]. This time the first three digits of our estimate are correct. This suggests an interesting pattern: If the relative error is bounded by  $\frac{1}{10^p}$ , then our approximation has the first  $p$  digits correct.

Unfortunately, this is not quite true. Consider the relative error  $\varepsilon_x \leq \frac{1}{10000}$ . Then our estimate must come from the range [31638.8358, 31645.1642], so it is quite possible that it was, say, 31639 and the first four digits are not correct. You can see where the problem is, sometimes when adding we have to carry over a 1, or we have to borrow it when subtracting. We could try to play it safe and talk of  $p - 1$  correct digits.

However, if we try to formulate this mathematically, we run into another problem: We do not quite know what a correct digit is.

**Example 3a.c:** We all know  $\pi$  in some form. Probably everyone met the estimate 3.14, some people may remember longer forms (due to a silly bet in high-school, I happen to remember  $\pi$  to 17 decimals—a completely useless piece of data occupying a chunk of my precious memory). How correct are these various guesses?

When somebody says  $\pi = 3.14$  or perhaps  $\pi \approx 3.1432768$ , it seems clear that the first three digits of these two approximations are correct, because they agree with  $\pi = 3.14159265358979323\dots$ , in fact they agree with all “longer” versions of  $\pi$  that people remember, like 3.141, 3.14159265 etc. I think all people would agree here.

How about 3.1415? One would be tempted to say that the first five digits (that is, all digits) are correct, and support it by the same argument, namely that they agree with digits of the precise value.

However, note that the next digit in  $\pi$  is “9”, so in fact its fifth digit “5” feels more like a six. Here’s another way to look at it: The guess 3.1415 has absolute error about 0.0000926..., while the guess 3.1416 has absolute error about 0.0000073..., which is more than ten times smaller. Thus it can be argued that 3.1416 has five digits correct, while 3.1415 only four.

△

We could now try to come up with some definition of correct digits and develop some theory about them, but it is not worth our time, and I do not recall anyone doing it. It is not used in mathematical theory. Still, people do use this notion in informal talk and we all have some vague idea what it means, so it is not quite useless. In particular, our observation gave us some feeling for the meaning of the relative error, so let’s express it as an informal rough guide:

- **Rule of thumb:** If the relative error is  $\frac{1}{10^{p+1}}$ , then the first  $p$  digits of a number are correct (whatever this means).

Alternatively, if we express the relative error as a real number with a fixed decimal dot, then the number of zeros after the decimal dot tells us how many digits in our estimate are correct.

The second statement looks very convenient and you can try that it works in our example. Note that in computers we actually work in a binary code, then in the first statement we would have to use  $\frac{1}{2^{p+1}}$ . The second statement has the advantage that it works (roughly) in any number system, we just have to express our approximation and its error with respect to the same base.

Our rule of thumb went from relative error to precision. We will go the other way later, see Fact 3c.2.

We wanted to get some feeling for the meaning of relative error here. In our examples we observed the following fact:

- If  $x, \hat{x} > 0$ , then  $(1 - \varepsilon_x)x < \hat{x} < (1 + \varepsilon_x)x$ .

Unfortunately, this is not exactly useful, because it tells us where to find our estimate  $\hat{x}$  when we know  $x$ . We would appreciate it the other way around. Of course, the inequalities can be rearranged, assuming that  $1 - \varepsilon_x > 0$  we can write

$$\bullet \frac{1}{1+\varepsilon_x}\hat{x} < x < \frac{1}{1-\varepsilon_x}\hat{x}.$$

The assumption  $1 - \varepsilon_x > 0$  is not restrictive, since the case  $\varepsilon_x \geq 1$  means that the error is at least as large as our measurement, which we definitely do not expect in applications.

Using such estimates we can also obtain a result about relative error of our measurement  $\hat{x}$  based on the knowledge of the absolute error  $E_x$ , which makes sense as we often have a fairly good idea how large it is.

**Fact 3a.3.**

Let  $\hat{x} \neq 0$  be some approximation of the number  $x$ . Assume that we have some estimate  $e_k$  for its absolute error and that it satisfies  $\frac{e_k}{|\hat{x}|} < \frac{1}{2}$ . Then the relative error satisfies  $\varepsilon_x < 2\frac{e_k}{|\hat{x}|}$ .

**Proof:** For convenience, denote  $\eta = \frac{e_k}{|\hat{x}|}$ . Then  $|E_x| \leq e_k = \eta|\hat{x}|$ , that is,  $-\eta|\hat{x}| < x - \hat{x} < \eta|\hat{x}|$ .

Assume that  $\hat{x} > 0$ . Then we can rearrange this inequality to read

$$(1 - \eta)\hat{x} < x < (1 + \eta)\hat{x},$$

in particular  $|x| > (1 - \eta)|\hat{x}|$ . If  $\hat{x} < 0$ , we rearrange to

$$(1 + \eta)\hat{x} < x < (1 - \eta)\hat{x},$$

now all numbers are negative and we again confirm that  $|x| > (1 - \eta)|\hat{x}|$ . Thus we can in general estimate that

$$\varepsilon_x = \frac{|E_x|}{|x|} < \frac{e_k}{(1 - \eta)|\hat{x}|} = \frac{1}{1 - \eta} \frac{e_k}{|\hat{x}|} < \frac{1}{\frac{1}{2}} \eta = 2\eta.$$

The proof is complete. □

This will be useful later, in particular in chapter 19. If we want to find an approximation with relative error at most  $\varepsilon$ , it is enough to find  $\hat{x}$  so that  $x$  is guaranteed to be between  $(1 - \frac{\varepsilon}{2})\hat{x}$  and  $(1 + \frac{\varepsilon}{2})\hat{x}$ .

### 3b. Error propagation

In this section we address the key question of this chapter: We need to do some calculations and numbers that enter into it are not precise. How do their errors influence the outcome? In other words, how do errors in input influence the output?

Since most calculations are based on the basic four algebraic operations, it makes sense to start with them.

**Thm 3b.1.**

Consider real numbers  $x, y$  and their estimates  $\hat{x}, \hat{y}$ . Then the following are true:

- |   |   |
|---|---|
| (i) $ E_{cx}  =  c  \cdot  E_x ,$   | $\varepsilon_{cx} = \varepsilon_x$ for $c \in \mathbb{R};$  |
| (ii) $ E_{x+y}  \leq  E_x  +  E_y ,$  | $\varepsilon_{x+y} \leq \max(\varepsilon_x, \varepsilon_y)$ for $x, y > 0;$   |
| (iii) $ E_{x-y}  \leq  E_x  +  E_y ,$   | $\varepsilon_{x-y} \leq \max(\varepsilon_x, \varepsilon_y) \frac{ x + y }{ x-y };$                                      |
| (iv) $ E_{x \cdot y}  \leq  y  \cdot  E_x  +  \hat{x}  \cdot  E_y ,$            | $\varepsilon_{x \cdot y} \leq \varepsilon_x + (1 + \varepsilon_x)\varepsilon_y;$  |
| (v) $ E_{x/y}  \leq \frac{1}{ y } ( E_x  +  E_y  \frac{ \hat{x} }{ \hat{y} }),$ | $\varepsilon_{x/y} \leq \varepsilon_x + \frac{1+\varepsilon_x}{1-\varepsilon_y} \varepsilon_y$ for $\varepsilon_y < 1.$ |

All parts are proved similarly. We expect a precise outcome of some calculation, for instance  $x + y$ , but instead we obtain, due to approximate inputs, the answer  $\hat{x} + \hat{y}$ . To these two numbers we then apply the definitions of absolute and relative errors. Some operations can be done easily, some require tricks.

**Proof:** (i):  $E_{cx} = cx - c\hat{x} = c(x - \hat{x}) = cE_x$ , so  $|E_{cx}| = |c| \cdot |E_x|$ . Then

$$\varepsilon_{cx} = \frac{|E_{cx}|}{|cx|} = \frac{|c| \cdot |E_x|}{|c| \cdot |x|} = \varepsilon_x.$$

(ii):  $E_{x+y} = (x + y) - (\hat{x} + \hat{y}) = (x - \hat{x}) + (y - \hat{y}) = E_x + E_y$ . By the triangle inequality,  $|E_{x+y}| = |E_x + E_y| \leq |E_x| + |E_y|$ .

$$\varepsilon_{x+y} = \frac{|E_x + E_y|}{|x+y|} \leq \frac{|E_x| + |E_y|}{|x+y|} = \frac{|E_x|}{|x|} \frac{|x|}{|x+y|} + \frac{|E_y|}{|y|} \frac{|y|}{|x+y|} = \varepsilon_x \frac{|x|}{|x+y|} + \varepsilon_y \frac{|y|}{|x+y|}.$$

Now we assume  $x, y > 0$  and replace  $\varepsilon_x, \varepsilon_y$  with the larger of the two.

$$\varepsilon_{x+y} \leq \max(\varepsilon_x, \varepsilon_y) \frac{x+y}{x+y} = \max(\varepsilon_x, \varepsilon_y) \frac{x+y}{x+y} = \max(\varepsilon_x, \varepsilon_y).$$

(iii): The estimate for  $E_{x-y}$  is just like in (ii).

$$\varepsilon_{x-y} = \frac{|E_x + E_y|}{|x-y|} \leq \frac{|E_x| + |E_y|}{|x-y|} = \varepsilon_x \frac{|x|}{|x-y|} + \varepsilon_y \frac{|y|}{|x-y|} \leq \max(\varepsilon_x, \varepsilon_y) \frac{|x| + |y|}{|x-y|}.$$

(iv): Here we will not be able to properly rearrange terms in the absolute error right from the start, so we improve situation by a subtracting a suitable term, which we have to add immediately in order to preserve equality.

$$E_{x \cdot y} = xy - \hat{x}\hat{y} = xy - \hat{x}y + \hat{x}y - \hat{x}\hat{y} = y(x - \hat{x}) + \hat{x}(y - \hat{y}) = yE_x + \hat{x}E_y.$$

Then we apply absolute value and the triangle inequality.

Note that it is possible to add-subtract the term  $x\hat{y}$  and obtain the estimate  $E_{x \cdot y} = \hat{y}E_x + xE_y$ .

From our estimate we get

$$\varepsilon_{x \cdot y} = \frac{|E_{xy}|}{|xy|} \leq \frac{|y| \cdot |E_x| + |E_y| \cdot |\hat{x}|}{|x| \cdot |y|} = \frac{|E_x|}{|x|} + \frac{|E_y| |\hat{x}|}{|x| \cdot |y|} = \varepsilon_x + \varepsilon_y \frac{|\hat{x}|}{|x|} \leq \varepsilon_x + \varepsilon_y (1 + \varepsilon_x).$$

We used the estimate  $|\hat{x}| \leq |x| + |E_x| = (1 + \varepsilon_x)|x|$ .

(vi):

$$\begin{aligned} E_{x/y} &= \frac{x}{y} - \frac{\hat{x}}{\hat{y}} = \frac{x\hat{y} - \hat{x}y}{y\hat{y}} = \frac{x\hat{y} - \hat{x}\hat{y} + \hat{x}\hat{y} - \hat{x}y}{y\hat{y}} = \frac{\hat{y}E_x - \hat{x}E_y}{y\hat{y}} = \frac{E_x}{y} - \frac{\hat{x}E_y}{y\hat{y}} \\ &= \frac{1}{y} \left( E_x - \frac{\hat{x}}{\hat{y}} E_y \right). \end{aligned}$$

Passing to absolute value and using the triangle inequality we get the statement. Note that an alternative statement can be deduced if, in the third step, we introduce  $xy$  instead of  $\hat{x}\hat{y}$ .

Relative error:

$$\varepsilon_{x/y} = \frac{|E_{x/y}|}{|x/y|} \leq \frac{|y|}{|x|} \frac{1}{|y|} \left( |E_x| + \frac{|\hat{x}|}{|\hat{y}|} |E_y| \right) = \frac{|E_x|}{|x|} + \frac{|\hat{x}|}{|\hat{y}|} \frac{|y|}{|x|} \frac{|E_y|}{|y|} = \varepsilon_x + \frac{|\hat{x}|}{|\hat{y}|} \frac{|y|}{|x|} \varepsilon_y.$$

The final result is obtained using  $|\hat{x}| \leq |x| + |E_x| = (1 + \varepsilon_x)|x|$  and  $|\hat{y}| \geq |y| - |E_y| = (1 - \varepsilon_y)|y|$ , here we had to assume that  $|E_y| < |y|$ .

Note that using the alternative estimate we would get  $\varepsilon_{x/y} = \frac{|y|}{|\hat{y}|} (\varepsilon_x + \varepsilon_y) = \frac{1}{1 - \varepsilon_y} (\varepsilon_x + \varepsilon_y)$ .  $\square$

Note that for positive numbers  $\varepsilon < \frac{1}{2}$  we have  $\frac{1}{1 - \varepsilon} \leq 1 + 2\varepsilon$ , so if  $\varepsilon_y$  is small enough, we can rewrite the result for division as

$$\varepsilon_{x/y} \leq \varepsilon_x + (1 + \varepsilon_x)(1 + 2\varepsilon_y)\varepsilon_y.$$

The estimates of absolute error are not used often, rather, they are tools for getting the more interesting results on relative error. Indeed, it is the relative error that decides how reliable our

answer is. However, while the results above are exact, we rarely use them in this precise form, their purpose is different: To warn us about dangers. And for this purpose it is enough to use simplified versions that are perhaps not so precise but easy to analyze.

We will now assume that the approximations are really good. Thus we may imagine that  $\frac{\hat{x}}{x}$  is essentially one, or that expressions like  $1 + \varepsilon_x$  or  $1 - \varepsilon_y$  are equal to one. IN this way we arrive at more friendly formulas.

We will make one more simplifying step. In many situations we do not have precise knowledge of relative errors of individual inputs, but we have a common upper bound  $\varepsilon$  for them. Using this we get even friendlier estimates. As is usual in linear algebra, we join addition and multiplication by a constant into one linear formula.

**Fact 3b.2.**

Assume that numbers  $x, y > 0$  have relative errors bounded by  $\varepsilon > 0$ . Then the following estimates are (almost) true:

$$\begin{aligned} \varepsilon_{cx+y} &\leq \varepsilon, & \varepsilon_{x-y} &\leq \frac{x+y}{|x-y|}\varepsilon, \\ \varepsilon_{x \cdot y} &\leq 2\varepsilon, & \varepsilon_{x/y} &\leq 2\varepsilon. \end{aligned}$$

Note that the first two inequalities are actually precise, but for multiplication and division, the relative error may be slightly larger than  $2\varepsilon$ .

Now it is time to talk about operations and errors. We see that the best operations are addition and multiplication by a known constant, because the answer is as reliable as the input. Multiplication and division are also reasonably good (and sometimes better), but we have to be really very careful if our calculations include many such operations. Let us think about it.

Imagine that we do ten multiplications in a row with numbers that originally had relative error  $\varepsilon$ . By our estimate, the result can have error as large as  $2^{10}\varepsilon = 1024\varepsilon \approx 1000\varepsilon$ . By the “rule of thumb”, this translates to three digits that we lost from the reliable part of the result. We could say that on average, for every three and a third multiplications we lose one correct digit in our result. The same is true about division.

The most dangerous of the bunch is subtraction. When we subtract two numbers that are close, the fraction in the estimate grows large, suggesting that the answer can be completely off. And indeed, sometimes it is. In fact, we do not need any theory to see this.

Imagine that we have two numbers that are very similar and share a common bound on relative error. This relative error translates to a number of digits that are reliable (common to both numbers) and the remaining digits to the right are suspect. Let’s imagine two such numbers.

$$\begin{array}{c} \xleftarrow{\text{OK}} \\ abc\dots e f uv\dots w \\ - abc\dots e f ux\dots y \end{array}$$

The letters  $a, b, c, \dots$  represent digits. Because these numbers are close, the first digits are common to both. The digits  $a, b, c, \dots, f$  are guaranteed to be correct, after that there is the questionable part. What happens when we subtract?

The reliable part gets cancelled and the outcome of the operation is determined exactly by the parts that we trust least.

Thus when we design calculations, we have to be careful about subtracting very similar numbers.

It should be noted that the formulas are upper bounds, so often the situation is better, sometimes markedly so. For instance, when we add two approximate numbers, then there is a chance that their errors go in opposite directions (one is a bit smaller than it should be, the other larger), so when added the errors cancel each other (perhaps not entirely, but at least partially), and as a

result the output is more reliable than the input. In fact, statistically, this happens in half of the additions. Statistical study of errors is a useful field of mathematics and people who routinely work with errors (for instance those in experimental fields) are well versed in it.

With all operations we can get lucky (and we often do), but not always; since we have to guarantee that our result is good enough, we have to expect the worst and find out how to handle it.

**Example 3b.a:** Assume that instead with numbers  $x = 13$  and  $y = 14$  we work with their approximations  $\hat{x} = (1 + 0.1)x = 14.3$  and  $\hat{y} = (1 + 0.1)y = 15.4$ . Our previous investigations suggest that the relative error of both approximations will be 0.1. Indeed,

$$\varepsilon_x = \frac{|x - \hat{x}|}{|x|} = \frac{0.1x}{x} = 0.1,$$

similarly for  $y$ . With relative error 0.1 we suspect that already the second significant digit of our approximations may be off, and indeed it is in case of  $y$ , so things work as expected. Now we look at operations.

$$\begin{aligned} x + y &= 27, & \hat{x} + \hat{y} &= 29.7, \\ \varepsilon_{x+y} &= \frac{|27 - 29.7|}{|27|} = 0.1; \\ x - y &= -1, & \hat{x} - \hat{y} &= -1.1, \\ \varepsilon_{x-y} &= \frac{|(-1) - (-1.1)|}{|-1|} = 0.1; \\ x \cdot y &= 182, & \hat{x} \cdot \hat{y} &= 215.6, \\ \varepsilon_{x \cdot y} &= \frac{|182 - 215.6|}{|182|} \approx 0.19; \\ \frac{x}{y} &= \frac{13}{14}, & \frac{\hat{x}}{\hat{y}} &= \frac{143}{154}, \\ \varepsilon_{x/y} &= 0. \end{aligned}$$

We see that addition and multiplication behave as expected with worst case scenario, where the addition preserves the relative error from input and multiplication roughly doubles it. With division we got lucky. With subtraction there are no limits on how good or bad it can get, in our example we again got lucky and relative error from input was preserved.

Now we will try these approximations:  $\hat{x} = (1 + 0.1)x = 14.3$  and  $\hat{y} = (1 - 0.1)y = 12.6$ . Again, the relative errors are 0.1. We look at operations.

$$\begin{aligned} x + y &= 27, & \hat{x} + \hat{y} &= 26.9, \\ \varepsilon_{x+y} &= \frac{|0.1|}{|27|} \approx 0.004; \\ x - y &= -1, & \hat{x} - \hat{y} &= 1.7, \\ \varepsilon_{x-y} &= \frac{|-2.7|}{|-1|} = 2.7; \\ x \cdot y &= 182, & \hat{x} \cdot \hat{y} &= 180.18, \\ \varepsilon_{x \cdot y} &= \frac{|1.82|}{|182|} \approx 0.001; \\ \frac{x}{y} &= \frac{13}{14}, & \frac{\hat{x}}{\hat{y}} &= \frac{143}{126}, \\ \varepsilon_{x/y} &\approx 0.22. \end{aligned}$$

This time we got lucky with addition and multiplication, which is no surprise given that the two

errors pulled in opposite ways. By the way, how does our investigation about the meaning of relative error work here? The relative error of multiplication is 0.001 and indeed, the first two digits of the approximate result fit with the correct answer.

In the relative error for the addition we see two zeros after the decimal dot, are the first two digits of our estimate correct? Not quite, but then again, note that the rule of thumb assumed that the first digit after the zeros is 1. Here, 0.004 is almost half way between 0.001 and 0.01, so we see that the second significant digit is a bit unsure as well. Actually, when we want to talk of two significant digits, it makes sense to round in this way, and our sum rounds to 27, correct to two places. So the rule of thumb sort of works, but you can see why we did not want to make any definite statement about it. It is just a rough guide.

Division works according to our rules, the relative error from input doubled (sort of). Finally, subtractions has shown its hand here, the relative error increased by a very large factor (27 times). As we discussed, relative error of 1 or more indicates a worthless approximation, and this is exactly seen here, our approximate subtraction did not even get the sign of the outcome right.

△

**Remark:** Using the above results on elementary operations we easily deduce for instance the following (we show a precise estimate for absolute value and then a simplified estimate for relative value as in the last fact):

- $|E_{1/x}| \leq \frac{|x|}{|\hat{x}|} |E_x|$  and  $\varepsilon_{1/x} \leq \varepsilon$ ,
- $|E_{x^n}| \leq \max(|x|, |\hat{x}|)^n n |E_x|$  and  $\varepsilon_{x^n} \leq n\varepsilon$ .

An inquisitive reader may want to prove them directly as an exercise.

△

We know how basic operations behave and using these we can analyze more complicated formulas that are created using them. However, we sometimes also work with functions. What happens then? Often we can use our knowledge of algebra to deduce what is going on.

**Example 3b.b:** What is the relative error of  $e^x$  when the input  $\hat{x}$  has relative error  $\varepsilon$ ?

We follow the usual approach using definitions and start with absolute error.

$$E_{e^x} = e^x - e^{\hat{x}} = e^{x-\hat{x}+\hat{x}} - e^{\hat{x}} = e^{E_x+\hat{x}} - e^{\hat{x}} = e^{\hat{x}}(e^{E_x} - 1).$$

This result, however, does not fit well with the next step, so we try differently.

$$E_{e^x} = e^x - e^{\hat{x}} = e^x - e^{\hat{x}-x+x} = e^x - e^{x-E_x} = e^x(1 - e^{-E_x}),$$

from here we can do

$$\varepsilon_{e^x} = \frac{e^x |1 - e^{-E_x}|}{e^x} = |1 - e^{-x\varepsilon}|.$$

We see that the result is not exactly easy to analyze. This error goes to zero as  $\varepsilon \rightarrow 0$ , that is nice, but unfortunately we do not have  $\varepsilon \rightarrow 0$  in real life, the relative error of input is usually given and stays the same. We also see that when  $x$  is large, then  $e^{-x\varepsilon} \approx 0$  and thus the relative error tends to 1, which is very unpleasant.

If one does wish to obtain a not so precise but more digestible form, it is possible to rewrite  $1 - e^{-E_x}$  using Taylor expansion and then simplify. However, we prefer to arrive at the same answer in a different way.

△

The other popular approach (and more general) is to use Taylor expansion.

**Example 3b.c:** In general we have this expansion:

$$f(a+h) = f(a) + f'(a)h + \frac{1}{2}f''(a)h^2 + \frac{1}{3!}f'''(a)h^3 + \dots$$

If we use it with  $a = x$  and  $h = -E_x$ , then  $a + h = \hat{x}$  and we obtain

$$f(\hat{x}) = f(x) - f'(x)E_x + \frac{1}{2}f''(x)E_x^2 - \frac{1}{3!}f'''(x)E_x^3 + \dots,$$

that is,

$$E_{f(x)} = f(x) - f(\hat{x}) = f'(x)E_x - \frac{1}{2}f''(x)E_x^2 + \frac{1}{3!}f'''(x)E_x^3 - \dots$$

If  $E_x$  is very small, then higher powers become negligible (see  $O(h^a)$  in chapter 38) and we can estimate  $E_{f(x)} \approx f'(x)E_x$ . Thus

$$\varepsilon_{f(x)} \approx \frac{f'(x) \cdot x}{f(x)} \varepsilon_x.$$

Let's try it with  $f(x) = e^x$ . We obtain

$$\varepsilon_{e^x} \approx \frac{e^x x}{e^x} \varepsilon = x\varepsilon.$$

We see that the relative error of the outcome grows linearly with  $x$ , not a very pleasant situation.

△

All the work in this section was done on the assumption that operations are performed properly. We usually use computers and calculators to compute for us and they, unfortunately, do not do operations precisely. Thus we have to look at operations from a practical point of view. Moreover, doing calculations on calculator/computer actually introduces errors right from the start, so this is another reason to look at them closer, they are a major (and sometimes the only) source of errors on input.

### 3c. Numbers in computers

Computers can work with many types of numbers. If we are lucky, we may be able to solve our problems using integers only, and computers can work with integers precisely (unless they get too large), offering precise answers. However, usually we are not that lucky. Engineering, physics and other sciences use numbers that are not integers, moreover, they use functions (sine, logarithm, roots) where the outcome is usually not even a rational number. Unfortunately, the set of real numbers is too rich to be handled by computers precisely.

Given the limited nature of a computer's memory, we are naturally forced to make do with a finite set of numbers. The most popular way to mimic the world of real numbers under such conditions is to use the floating point representation.

First we note that every real number can be written in the form

$$x = d_1.d_2d_3d_4d_5\dots \times 10^e,$$

where  $d_i$  are digits (that is, integers from the range  $0, \dots, 9$ ), while  $e$  is any integer. For instance,

$$135 = 1.35 \times 10^2, \quad \frac{1}{400} = 0.0025 = 2.5 \times 10^{-3}, \quad \frac{1710}{99} = 1.72727272\dots \times 10^1.$$

The part  $d_1.d_2d_3d_4d_5\dots$  is called **mantissa** and  $e$  is called the **exponent**. The mantissa can be infinite. You may recall that if the mantissa is finite or starts being periodic, then the number is in fact a rational number (a fraction), but we will not use this piece of knowledge here. It may be also surprising for some that the expression as above is not unique, for instance for the number 1 we have two representations:  $1.00000\dots = 0.99999\dots$ . Again, we will not need to worry about it here and so we will leave this topic to curious readers. If you want to follow this trail, sooner or later you encounter the notion of a series of real numbers.

So far we lived in the world of theory, now we need to pass to computers (and calculators). The obvious problems lies with those chains  $d_1d_2d_3\dots$  that can be arbitrarily long. It is also quite obvious that the further we go to the right, the less important the digits are. Thus it seems natural to decide on a certain number  $p$  of digits we care about and ignore the rest.

It also pays to unify the notation a bit, so we decide that we will write all numbers (with the exception of zero) in such a way that  $d_1 \neq 0$ , we did so in the examples above. When we also restrict the values for the exponent  $e$ , we arrive at a finite set of numbers that a computer can store and work with. We say that we expressed numbers in a **floating point format**. The advantage is that these numbers can be stored as an integer  $e$  and another integer  $d_1d_2\dots d_p$ .

It should be noted that some people see floating point numbers differently. While we demanded that the digit before the decimal point is not zero, it is also possible to do the opposite and ask for zero there, then we would want the first digit after the decimal dot to be non-zero. With our numbers above it would go

$$135 = 0.135 \times 10^3, \quad \frac{1}{400} = 0.0025 = 0.25 \times 10^{-2}, \quad \frac{1710}{99} = 0.172727272\dots \times 10^2.$$

Generally, people who prefer this work with numbers  $x = 0.d_1d_2d_3d_4d_5\dots \times 10^e$ , again they need to store integers  $d_1d_2d_3d_4\dots$  and  $e$ . It is fairly obvious that these two approaches differ only formally, one can convert between them easily. One and the other may have some minor practical advantages here and there, but essentially it is just a matter of taste. Again, it is advisable to choose one form and stick with it, we made our choice above.

**Definition 3c.1.**

By a floating point representation of a number  $x$  with respect to base  $\beta$ , with precision of  $p$  significant digits we mean the best approximation  $fl(x)$  of  $x$  that can be written as

$$fl(x) = d_1.d_2d_3\dots d_p \times \beta^e,$$

where  $d_1 \in \{1, \dots, \beta - 1\}$  and  $d_2, \dots, d_p \in \{0, 1, \dots, \beta - 1\}$ .

The number  $e$  is called the exponent, the part  $d_1.d_2\dots d_p$  is called the significand or the mantissa.

The word “best” can be interpreted in various ways. We may ask for the nearest in the sense of distance between numbers (rounding to the nearest number), or we can look only at numbers not larger than  $x$  when picking  $fl(x)$  (rounding down, truncating). We note that the latter is obviously less precise, the error can be twice as large compared to rounding to the nearest, but it has the advantage of being done efficiently: We simply find the (possibly infinite) expansion of  $x$  in the number system with base  $\beta$  and ignore all but the first  $p$  digits. This is the reason why truncating is rather popular in computer science. On the other hand, engineers prefer rounding to the nearest, because they really care about precision. Now I am more of a computer guy than an engineer, but we are trying to be useful here so I will go with them. In this book we will always round to the nearest allowed number.

When we keep the first  $p$  digits of a number, then these digits should be reliable (with possible exception of the last one, given that we do not really know what a correct digit is and allow for different ways of rounding). We know that there is a relationship between correct digits and relative error, before we explored it in one direction (from relative error to correct digits), now we go the other way.

**Fact 3c.2.**

Assume that a number  $x \neq 0$  was represented as  $fl(x) \neq 0$  in floating point representation with base  $\beta$  and precision  $p$ . Then the relative error is bounded as follows:

$$\varepsilon_x \leq \frac{1}{2}\beta \cdot \beta^{-p}.$$

As noted above, if the representation was done using truncation, the estimate for relative error would double.

**Proof:** For simplicity, assume that  $x > 0$ .

We can write the number  $x$  with respect to base  $\beta$  so that the first digit is not zero, that is,

$$x = d_1.d_2d_3\dots d_p d_{p+1}\dots \times \beta^e.$$

In order to obtain its floating point representation, we round this to  $p$  digits, we use rounding to the nearest number. There are two possible outcomes.

1) If we round down, then

$$fl(x) = d_1.d_2d_3\dots d_p \times \beta^e.$$

The error is

$$E_x = x - fl(x) = 0.00\dots 0d_{p+1}\dots \times \beta^e \geq 0,$$

there are  $p - 1$  zeros before the digit  $d_{p+1}$ . Thus we can write it as  $d_{p+1}.d_{p+2}d_{p+3}\dots\beta^{-p}\beta^e$ . Note that in order for this rounding down to happen, the digit  $d_{p+1}$  cannot be larger than  $\frac{1}{2}\beta$ . Moreover, if it is  $\frac{1}{2}\beta$ , then the digits after it must be zero. Thus we can estimate

$$E_x = d_{p+1}.d_{p+2}d_{p+3}\dots\beta^{-p}\beta^e \leq \frac{1}{2}\beta \cdot \beta^{-p}\beta^e.$$

On the other hand,

$$x \geq 1.00\dots 0 \times \beta^e.$$

Thus we obtain

$$\varepsilon_x = \frac{|E_x|}{x} \leq \frac{\frac{1}{2}\beta \cdot \beta^{-p}\beta^e}{1 \times \beta^e} = \frac{1}{2}\beta \cdot \beta^{-p}.$$

2) If we round up, then

$$fl(x) = d_1.d_2d_3\dots D_p \times \beta^e,$$

where  $D_p = d_p + 1$ . The error is some negative number that cannot be larger than

$$0.00\dots 0D_{p+1} \times \beta^e,$$

where  $D_{p+1} = (\beta - 1) - d_{p+1}$  and there are  $p - 1$  zeros before it. Rounding up means that  $d_{p+1} \geq \frac{1}{2}\beta$ , therefore  $D_{p+1} \leq \frac{1}{2}\beta$  and we can estimate

$$|E_x| = -E_x \leq 0.00\dots 0D_{p+1} \times \beta^e = D_{p+1} \times \beta^{-p}\beta^e \leq \frac{1}{2}\beta\beta^{-p}\beta^e.$$

As before we conclude that  $\varepsilon_x \leq \frac{1}{2}\beta^{-p}$ .

□

There is one more thing we have to consider. Given limited memory, there must be also a bound on exponents  $e$ . For instance, my calculator allows only exponents in the range  $-99\dots 99$  and I understand this is fairly common. This means that there are only finitely many numbers available for calculations.

One unpleasant consequence is that there is the largest possible number and anything over it gets discarded as an error. For instance, I am unable to compute  $70!$  on my calculator. When computation leads to value that is too large, we call it the “overflow error”.

This has to be taken into account when choosing formulas for calculation. For instance, if you plan to evaluate  $\binom{n}{3}$  as  $n \cdot (n - 1) \cdot (n - 2)/6$ , you run a much higher risk of overflow error than if you go for  $n/3 \cdot (n - 1)/2 \cdot (n - 2)$ .

On the other hand, there is also the smallest possible (positive) number  $1.00\dots \times 10^{e_{\min}}$ . Some people call this the “machine epsilon”. Anything smaller than that is treated as zero by the computer (the “underflow error”). If numbers in our calculation drop below this threshold, they are lost. One may think that this loss is negligible, but it is quite possible that these small

numbers may have been enlarged again in steps to follow. Again, one has to be careful when designing evaluations of formulas.

The existence of smallest positive number has some interesting consequences.

- The computer cannot recognize or confirm zero. When some number inputted in a function yields zero, it does not mean that you found a root. You may be very far from it.
- The computer cannot recognize equality. The fact that two expressions, when subtracted, yield zero, does not mean that they are the same.

This is actually related to another problems, computer has troubles telling apart close numbers. Some people define “machine epsilon” as the distance between 1 and the smallest number greater than one that the computer can store. This is the finest distinction computer is capable of.

Now let’s talk about operations as performed in computers. With multiplication there is one possible danger, since it can increase numbers a lot, so we may run into an overflow error. Similarly, with division we may run into an underflow error. Other than that they behave quite well.

On the other hand, addition and subtraction are much more dangerous. Why? When two numbers are added (or subtracted), they must first be converted to a form where they have the same exponent. This is done by changing the number with smaller exponent, we shift its digits to the right in exchange for increased exponent. The problem is that when we shift, we are not allowed to remember the longer mantissa, so we lose information.

**Example 3c.a:** Imagine that we work with 3-digit precision, so the relative error is at most  $5 \cdot 10^{-3}$ . How do we add  $135000 + 543$ ? Computer sees it as follows:  $1.35 \times 10^5 + 5.43 \times 10^2$ . Now we need to fix the second number. On paper we would write  $0.00543 \times 10^5$ , but the computer can keep only the first three digits, namely the zeros. The computation therefore proceeds as follows.

$$1.35 \times 10^5 + 5.43 \times 10^2 = 1.35 \times 10^5 + 0.00 \times 10^5 = 1.35 \times 10^5.$$

It is as if we did not add anything, the summand got lost completely. One may argue that it influenced the outcome only by a small amount, the correct answer is more like  $1.36 \times 10^5$ , so the relative error that was created in this way is still small. That is almost true, the answer has relative error

$$\varepsilon_x = \frac{|1.35 \times 10^5 - 1.36 \times 10^5|}{1.36 \times 10^5} = \frac{1}{136} \approx 0.07.$$

Not bad, but note that it is larger than the error on input. This is the real danger here, if such addition is repeated, the error may grow beyond control.

Situation would not improve with a better precision. How about 4-digit precision?

$$1.356 \times 10^5 + 5.432 \times 10^2 = 1.356 \times 10^5 + 0.005 \times 10^5 = 1.361 \times 10^5.$$

This time the addition did not get lost totally, but some information is missing. As a matter of fact, by shifting and dropping digits we increase the relative error of that number thousand times. Again, one can argue that this number is so small compared to the first that it does not really influence the outcome (which is true), and again the danger lies in repeated additions. This is a big problem, and there are known cases of crucial applications where millions of additions with small errors accumulated large error, leading to a catastrophic failure.

△

This has been known for a while and there are ways to mitigate this problem. One approach that works well is to first order the numbers by their magnitude and then add them starting from the smallest. However, the sorting stage can be too time consuming, and moreover, sometimes we add numbers as they come, with no chance of storing them for delayed processing.

Another approach is to keep track of what was lost. One popular algorithm is the Kahan summation formula.

```
c:=0; s:=x1;
```

```

for j=2 to N do
  x:=xj-c; y:=s+x; c:=(y-s)-x;
  s:=y;

```

We use  $c$  to store the error made when adding and modify the next summand with it. It was shown that using this formula, the relative error when adding  $n$  numbers cannot be larger than about  $2\varepsilon + O(n\varepsilon^2)$ .

The troubles with addition/subtraction have another interesting consequence: On computers, the associative and distributive laws are no longer true. We will illustrate the former with a simple example, details of why some parts are lost can be worked out as in the above example.

On a machine with precision  $p = 3$ , floating point calculations lead to the following:

$$(x + y) + z = (135000 + 543) + 782 = (135000 + 0) + 782 = 135000 + 782 = 135000 + 0 = 135000,$$

$$x + (y + z) = 135000 + (543 + 782) = 135000 + 1320 = 135000 + 1000 = 136000.$$

The moral of the story is that when we do some calculations on a computer or a calculator, then these new problems are added to those from error propagation and there is no safe operation left. Let's make a comprehensive list.

	theory (error propagation)	computers (floating point)
addition	safe	bad when numbers different
subtraction	bad when numbers similar	bad when numbers different
multiplication	safe when not many	danger of overflow
division	safe when not many	danger of underflow

All this has to be taken into account when designing computations for real world engineering and science.

### 3d. Bonus: Scientific notation for numbers

Floating point representation is related to scientific notation for numbers. It is a common format for writing numbers in engineering and science in the form  $m \times 10^e$ . If we demand that  $1 \leq m < 10$ , then the number is called normalized. For instance,  $350 = 35 \times 10^1 = 3.5 \times 10^2$ , the last number is normalized.

A typical source of scientific numbers are calculations and measurements, then it is customary to include all digits that are known to be correct, in this context they are called "significant digits". While mathematically speaking  $1.4 = 1.40$ , for an engineer the second number carries more information. Indeed, in real world calculations we rarely end up with numbers as nice as 1.4, so an engineer assumes that the "4" is most likely followed by some other digits that are now known. Leading zeros do not count as significant digits, so the numbers  $0.308$ ,  $3.08 \times 10^{-1}$  and  $0.00308 \times 10^2$  all have three significant digits.

When a number comes from a measurement, it usually comes also with uncertainty, which can be thought of as a bound for absolute error. There are rules how to write such result of measurement:

- The uncertainty should be rounded to one significant digit with exception of the case when it would be 1, then we round to two significant places.
- The number and its uncertainty are written with the same exponent (so we write it only once) and we put so many digits in the number that the last one coincides with the last significant digit of the uncertainty. Note that this means that we write also digit (or two) that are not certain, breaking a previous rule.

Examples:  $276 \pm 3 \times 10^3$ ,  $2.76 \pm 0.03 \times 10^5$ ,  $9.876 \pm 0.013 \times 10^4$ .

- Often people prefer to choose the exponent in such a way that the significant digit(s) of the uncertainty appear just after the decimal point:  $27.6 \pm 0.3 \times 10^4$ . This rule usually prevents us from using the normal form.

Also the rounding is precisely specified for scientific calculations. In particular, it defines how we should proceed when the first digit we will not keep is 5. The rule says that we should round in such a way that the resulting number is even. So for instance when rounding to two significant digits we do  $125 \mapsto 120$  and  $135 \mapsto 140$ .

How come? At elementary school I was taught to always round 5 up. The problem with this rule is that 0.5 is exactly midway between two numbers. If we always round up, we create a bias which may through off calculations unnecessarily. Rounding “to the even” has the advantage of being done (on average) half of the time up and half of the time down, so there is no bias.

By the way, when I go shopping, the cash register works with prices with two decimal digits, but the sum is rounded to an integer. I wonder how they do it, in particular how they round amounts like 13.50.

## 4. Approximating derivative

We start our journey through numerical analysis with one of the simpler tasks: We will try to approximate the derivative  $f'(a)$  of a given function  $f$  at a given point  $a$ . The problem itself is quite simple, but the results and insights will come handy later; moreover, it will be a starting point for some very useful explorations.

### 4a. Estimating derivative from definition

First a natural question: Why don't we simply take a derivative of  $f(x)$  and then substitute  $a$  into it? Because the function  $f$  need not be given by a formula. Engineers often need to work with functions that they know only by their values at specific points. For instance, if  $f(x)$  represents local temperature at time  $x$ , then we can obtain the value of  $f$  at any given time (assuming that we have some measuring device), but we cannot expect to obtain a formula.

That is the right mindset for our question. We have a function  $f$ , not a formula but rather a procedure that allows us to evaluate  $f(x)$  at any  $x$ . Can we somehow deduce the value of  $f'(a)$ ?

There is a natural approach: The definition says that  $f'(a) = \lim_{x \rightarrow a} \left( \frac{f(x) - f(a)}{x - a} \right)$ . If we take a particular  $x$  close to  $a$  and substitute into the formula, we can hope that the outcome will be close to the actual value of  $f'(a)$ . In numerical analysis it works better if we use the other popular version of definition:  $f'(a) = \lim_{h \rightarrow 0} \left( \frac{f(a+h) - f(a)}{h} \right)$ . The number  $h$  serves as a natural measure of how close to  $a$  we got with our  $x = a + h$ . If our intuition is right (and if we are lucky), then we should obtain better approximations when we use smaller  $h$ . Thus  $h$  serves as an indication of "quality" of our attempt. We often call it the **step size**, for instance in this particular case it really tells us how large a step we made when going from  $a$  to  $x$ .

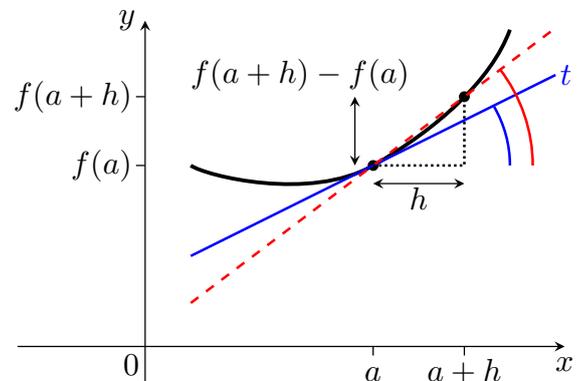
In numerical analysis we like to have this "quality indicator" positive, so we restrict our choice of  $h$  and arrive at the following natural formula:

$$f'(a) \approx \frac{f(a+h) - f(a)}{h} \text{ for } h > 0.$$

Geometric interpretation is quite obvious, we approximate the slope of the tangent line using the slope of a secant line that is close to the tangent.

The picture shows it all. The tangent line is in blue and the approximating secant line in red. The slope is determined using an appropriate triangle.

Note that in our picture, the slope of the secant line is greater than the one for the tangent line. If we use a different  $h > 0$ , it will still be true. This is obviously related to the fact that our function is concave up.



The reader may draw other pictures to see that for functions that are concave down, the opposite happens and our ratio would underestimate the actual value of derivative.

A consistent error in one direction is very unwelcome in numerical calculations, as it can pull the result off to one side. Such a calculation is then biased. Can we try something else?

The definition of derivative also allows for “looking left”. We can do it now by choosing a positive step size  $h > 0$  and step left to  $a - h$ . The formula for the resulting slope can be deduced from the picture on the right.

Another possibility is to imagine that  $h$  is negative in the definition, then the step size is  $k = -h > 0$ . Substituting this into the limit formula we get

$$f'(a) \approx \frac{f(a - k) - f(a)}{-k} = \frac{f(a) - f(a - k)}{k} \text{ for } k > 0.$$

This yields the same formula as geometric approach.

Note that this alternative approximation is also consistently biased, for instance it will always underestimate when a function is concave up.

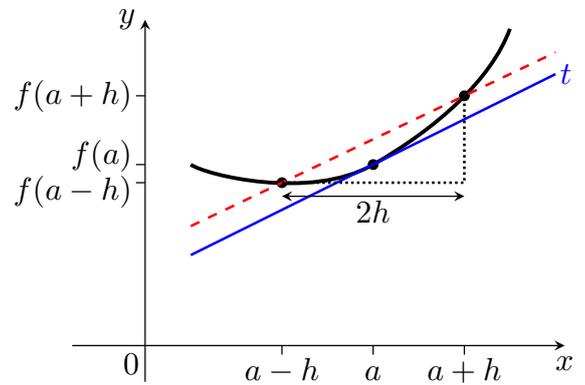
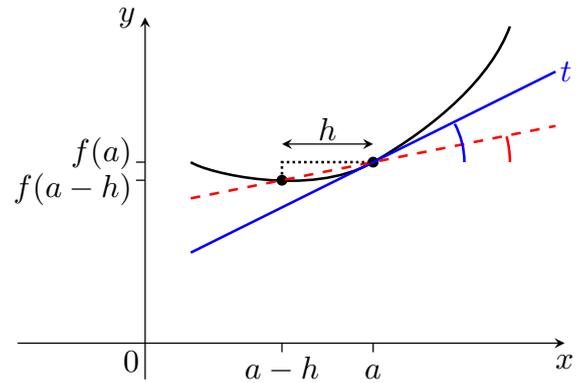
In general these two methods should be equally good (or bad), since for functions there is no left-right bias in their shapes. Because we suspect that the errors of these two methods should go in opposite ways, we may hedge our bets by taking the average:

$$\begin{aligned} f'(a) &\approx \frac{1}{2} \left( \frac{f(a + h) - f(a)}{h} + \frac{f(a) - f(a - h)}{h} \right) \\ &= \frac{f(a + h) - f(a - h)}{2h}. \end{aligned}$$

The resulting formula also has a natural geometric interpretation as the slope of a secant line that we obtain by looking left and right from  $a$ .

The picture suggests that this could work better than the previous two attempts. We will shortly show that this is indeed the case.

These are the most popular methods for approximating derivative and they have names.



**Definition 4a.1.**

Consider a function  $f$  that is differentiable at  $a$ . We introduce the following approximations for  $f'(a)$ :

$$f'(a) \approx \frac{f(a + h) - f(a)}{h}, \quad (\text{forward difference})$$

$$f'(a) \approx \frac{f(a) - f(a - h)}{h}, \quad (\text{backward difference})$$

$$f'(a) \approx \frac{f(a + h) - f(a - h)}{2h}, \quad (\text{central difference})$$

where  $h > 0$ .

So far so good, now it is time for the key question: If we use one of these formulas, what is the error that we make? Right now we mean the error of the method, that is, we assume that the calculations are done precisely.

Using basic knowledge of analysis we obtain a satisfactory answer.

**Theorem 4a.2.**

Let  $f$  be a function twice differentiable on some neighborhood  $I = (a - \delta, a + \delta)$  of a point  $a$ . Assume that its second derivative is bounded on  $I$ , that is, there exists some  $M$  so that  $|f''(x)| \leq M$  for  $x \in I$ . Then we have the following estimates:

$$\left| f'(a) - \frac{f(a+h) - f(a)}{h} \right| \leq \frac{M}{2}h,$$

$$\left| f'(a) - \frac{f(a) - f(a-h)}{h} \right| \leq \frac{M}{2}h$$

for all  $h \in (0, \delta)$ .

Before we start the proof, the reader has to face a key question: Am I comfortable with Taylor expansions and asymptotic growth? This topic is usually covered in calculus courses, for readers who feel that some review is in order we wrote an introduction, see chapter 38.

**Proof:** Consider some  $h \in (0, \delta)$ . Using Taylor polynomial with the Lagrange form of remainder we can write

$$f(a+h) = f(a) + f'(a)h + \frac{1}{2}f''(\xi)h^2$$

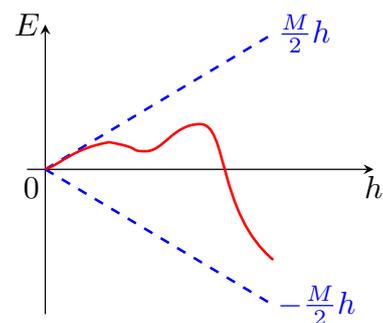
for some  $\xi \in (a, a+h) \subseteq I$ . Then we can estimate

$$f(a+h) - f(a) = f'(a)h + \frac{1}{2}f''(\xi)h^2 \implies \left| f'(a) - \frac{f(a+h) - f(a)}{h} \right| = \left| \frac{1}{2}f''(\xi)h \right| \leq \frac{M}{2}h.$$

The estimate for the left difference is done similarly by expanding  $f(a-h)$ . □

Note that this estimate is only an upper bound, that is, it gives the worst case scenario. In particular it does not imply that if we compute an approximation with a certain  $h > 0$  and another with a smaller  $h$ , then the latter will have a smaller error. In fact, it can easily happen that the opposite takes place, and taking smaller step size sometimes (but rarely) increases the error.

The right interpretation of our estimate is that it provides a sort of “envelope” within which the error must be, as illustrated on the right (error in red). So while the error may sometimes increase when we make  $h$  smaller, it cannot escape our envelope and by taking still smaller and smaller  $h$  we eventually get very small errors. This applies to all estimates that come in this book.



Note that the result above can be stated as follows: The truncation error of these two methods is of order  $O(h)$ . We will now show another proof, in fact it is not just a proof but rather a way of obtaining approximating formulas for derivatives and their error estimates.

### 4b. Deriving differences using Taylor expansions

At the beginning we gather data. Let's say that we want to use the values  $f(a)$  and  $f(a+h)$  to approximate derivative  $f'(a)$ . First we expand them about the center  $a$ . We have to decide how long the expansion should be, with experience one can usually guess well enough. Here we will stop at  $h^2$  and express the remaining part of expansion using the  $O$  notation. By the way, this requires that  $f$  has continuous derivatives up to the order three.

$$f(a+h) = f(a) + f'(a)h + \frac{1}{2}f''(a)h^2 + O(h^3),$$

$$f(a) = f(a).$$

Now we combine the data. How? This is the moment when too much freedom ties one's hands because it's not clear what to do, there are many ways in which one can put together two numbers. So we restrict our freedom by going easy and just consider linear combinations  $Af(a+h) + Bf(a)$ . What do we obtain?

$$Af(a+h) + Bf(a) = [A+B]f(a) + Af'(a)h + \frac{1}{2}Af''(a)h^2 + O(h^3),$$

here we used the fact that  $AO(h^3) = O(h^3)$ . This is what we have, what do we want? The derivative. We see it on the right, we just have to adjust the equality for it to come out. The first step is obvious, we divide by  $h$ .

$$\frac{Af(a+h) + Bf(a)}{h} = [A+B]f(a)\frac{1}{h} + Af'(a) + \frac{1}{2}Af''(a)h + O(h^2).$$

Now it is time to determine values of  $A$  and  $B$  so that our combination provides exactly what we need. One requirement is clear, we want the derivative and thus we need  $A = 1$ . With two unknowns we have one more opportunity to ask for something, and it is a good thing, too. We will want to use our formula for small  $h$  and the first term on the right gets very large then, overshadowing the derivative that we want. We need that term out of our way, so the second requirement is that  $A+B = 0$ . Two equations with two unknowns, that sounds right. We easily solve, obtaining  $A = 1$ ,  $B = -1$ . We arrived at the formula

$$\frac{f(a+h) - f(a)}{h} = f'(a) + \frac{1}{2}f''(a)h + O(h^2).$$

This is the forward difference, and we see that the error of this method is given by whatever is left on the right other than the desired derivative. If we want to talk about the absolute error as defined above, we have to be careful about the sign. The error is “true value minus its approximation”, so we rearrange the equality accordingly:

$$f'(a) - \frac{f(a+h) - f(a)}{h} = -\frac{1}{2}f''(a)h + O(h^2).$$

Thus the error is

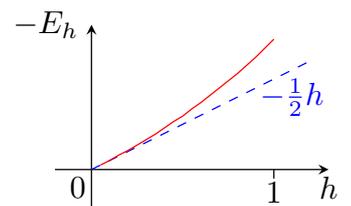
$$E_h = -\frac{1}{2}f''(a)h + O(h^2) = O(h).$$

This confirms our previous result, but it actually offers more. Since  $h^2$  is negligible compared to  $h$  around zero, for very small values of  $h$  the error is essentially equal to  $-\frac{1}{2}f''(a)h$ .

This is a useful information, because we want our estimates to be “sharp”. By this we mean that they cannot be improved in general, that is, there are cases when things are really as bad as we say. For instance, there was a chance that when we derived our error estimate with  $O(h)$  in the previous section, we used a tool that was not good enough, and perhaps with a better tool a better estimate, say,  $O(h^2)$ , might have been achieved. Then the error would decrease at much faster rate. Note that if this were the case, then our estimate  $O(h)$  would still be correct, but it would be needlessly too pessimistic, as a better result would be actually true. That is, our estimate would not be sharp then.

However, our more precise form of estimate shows that this cannot be the case, because for functions with  $f''(a) \neq 0$  the linear term appears in the error and thus—as an incomparably larger term—it cannot be kept under the control by  $h^2$ .

We made a practical experiment and approximated  $[e^x]'(0) = 1$  using the forward difference, in the graph we see how the error  $E_h$  (in the red) depends on  $h$ . Because the exponential is concave up, we expect this approximation to overshoot and the error  $E_h$  should be negative, which it is, we plotted  $-E_h$  to get a nicer picture.



Since  $[e^x]''(0) = 1$ , our last result says that the error curve should be really close to  $-\frac{1}{2}h$ , that is, linear, for very small  $h$ . We printed this line to our graph in blue. As expected, for larger values

of  $h$  the fit is not quite perfect, because the parts that we hid under the  $O(h^2)$  umbrella influence our error. However,  $h^2$  disappears fast for small  $h$  and soon its influence is gone, around  $h \approx 0.3$  the fit is already very good.

Now we address the question of how long the Taylor expansion should be. What happens if we take a longer one in the above calculations? We would get a more precise description of the error, for instance like this:

$$E_h = -\frac{1}{2}f''(a)h - \frac{1}{6}f'''(a)h^2 - \frac{1}{24}f''''(a)h^3 + O(h^4)$$

Then we would say that this is  $-\frac{1}{2}f''(a)h + O(h^2)$  or  $O(h)$ , so we would arrive at the same answer, just with a bit more work as we would have to carry around those useless terms. Nothing wrong here.

If we used a shorter expansion by just one term, we would get

$$f'(a) - \frac{f(a+h) - f(a)}{h} = O(h).$$

This means that we would confirm the error estimate as usually stated, but we would not know whether it is sharp, that is, we would not know whether it is possible to actually obtain a better estimate with more work (or another approach). Often this is not such a big deal and we save a bit of time, so working in a way that yields  $O(h^a)$  directly can be useful.

If we took a still shorter expansion, then we would not be able to finish our calculations as key terms would be missing. This would send us a clear message to start again, but with a longer expansion.

So far so good, but note that we usually do not have the information that we used in the above calculations, like  $f''(a)$  or  $M = \max |f''|$ . So from the practical point of view these estimates are not exactly terribly useful. However, they are important, and for two reasons. First, they allow us to compare different methods, and second, these estimates become building blocks later when developing methods for solving some other problems.

Now we will use more data, which usually allows for better information. What can we do with values  $f(a+h)$ ,  $f(a)$ ,  $f(a-h)$ ? We will follow exactly the same path. Since we are not quite sure how it will go, we will extend expansions by one term to be on the safe side, which means that  $f$  should be four times continuously differentiable.

$$\begin{aligned} f(a+h) &= f(a) + f'(a)h + \frac{1}{2}f''(a)h^2 + \frac{1}{6}f'''(a)h^3 + O(h^4), \\ f(a) &= f(a), \\ f(a-h) &= f(a) - f'(a)h + \frac{1}{2}f''(a)h^2 - \frac{1}{6}f'''(a)h^3 + O(h^4). \end{aligned}$$

Now we form a general linear combination of these rows.

$$\begin{aligned} Af(a+h) + Bf(a) + Cf(a-h) & \tag{*} \\ &= [A+B+C]f(a) + [A-C]f'(a)h + [A+C]\frac{1}{2}f''(a)h^2 + [A-C]\frac{1}{6}f'''(a)h^3 + O(h^4). \end{aligned}$$

We want to see  $f'(a)$  on the right, so we divide (\*) by  $h$ :

$$\begin{aligned} \frac{Af(a+h) + Bf(a) + Cf(a-h)}{h} & \\ &= [A+B+C]\frac{1}{h}f(a) + [A-C]f'(a) + \frac{1}{2}[A+C]f''(a)h + \frac{1}{6}[A-C]f'''(a)h^2 + O(h^3). \end{aligned}$$

Now we need to make sure that  $A-C=1$  so that the desired derivative appears on the right. Again, the term in front of it threatens to blot out our result due to  $\frac{1}{h} \rightarrow \infty$ , we prevent it by asking for  $A+B+C=0$ . We obtained two equations, which means that we are free to state one more requirement. We use it to get rid of the largest error term, that is, we require that  $\frac{1}{2}[A+C]=0$ . This system of equations leads to  $A=\frac{1}{2}$ ,  $B=0$ ,  $C=-\frac{1}{2}$ . After we substitute, we see that we obtained the formula for the central difference.

We rearrange a bit:

$$f'(a) - \frac{f(a+h) - f(a-h)}{2h} = -\frac{1}{6}f'''(a)h^2 + O(h^3).$$

This shows that the error is of order  $O(h^2)$  and that this estimate is sharp.

Let's go back to the formula (\*). We see other derivatives on the right. How about if we try to get  $f''(a)$  out of it? Then it would make sense to divide by  $h^2$ :

$$\begin{aligned} & \frac{Af(a+h) + Bf(a) + Cf(a-h)}{h^2} \\ &= [A+B+C]f(a)\frac{1}{h^2} + [A-C]f'(a)\frac{1}{h} + \frac{1}{2}[A+C]f''(a) + \frac{1}{6}[A-C]f'''(a)h + O(h^2). \end{aligned}$$

In order to get  $f''(a)$  on the right, we have to make sure that  $\frac{1}{2}[A+C] = 1$ . We also need the two terms before it to disappear (both grow large for small  $h$ ), which brings two more requirements:  $A+B+C = 0$  and  $A-C = 0$ . We ran out of conditions to make, so we cannot get rid of any error term. Solving the three equations we obtain  $A=C=1$ ,  $B=2$ , that is, the formula

$$\frac{f(a+h) - 2f(a) + f(a-h)}{h^2} = f''(a) + O(h^2).$$

We obtained an approximating formula for the second derivative (which is also sometimes needed) and an error estimate. Note that the order is one better than we expect, since we did not remove any terms. This is a bonus for symmetry of the situation. Algebraically we see it in the fact that when we made the coefficient  $A-C$  in the  $f'(a)$  term zero, it also removed the  $f'''(a)$  term.

This however means that we do not know whether the  $O(h^2)$  estimate is sharp. Perhaps some other terms in the error part disappear, too. In order to see this we would have to use longer expansions and do the work again. The next term in the linear combination is  $\frac{1}{24}[A+C]f''''(a)h^2$  and our choice makes  $A+C=1$ , so it does not disappear, our estimate is the best possible.

Let's put down what we just figured out.

**Fact 4b.1.**

Consider a function  $f$  that is three times continuously differentiable on some neighborhood of a point  $a$ . Then the following approximating formulas are true as  $h \rightarrow 0^+$ :

$$f'(a) = \frac{f(a+h) - f(a)}{h} + O(h), \quad (\text{forward difference})$$

$$f'(a) = \frac{f(a) - f(a-h)}{h} + O(h), \quad (\text{backward difference})$$

$$f'(a) = \frac{f(a+h) - f(a-h)}{2h} + O(h^2), \quad (\text{central difference})$$

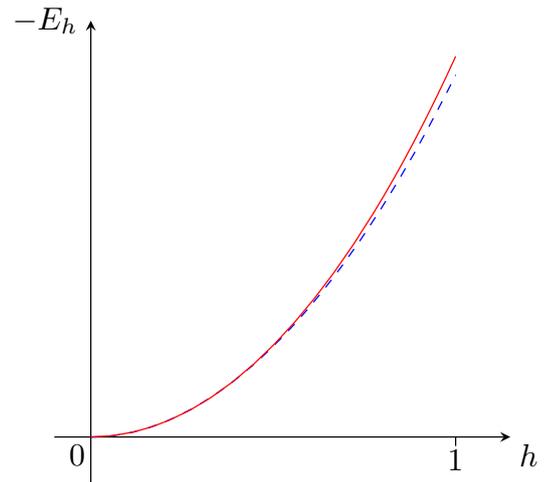
If  $f$  is four times differentiable on some neighborhood of  $a$ , we obtain

$$f''(a) = \frac{f(a+h) - 2f(a) + f(a-h)}{h^2} + O(h^2) \text{ as } h \rightarrow 0^+.$$

The result shows that in general, the central difference is preferable to forward and backward difference. In most cases it indeed shows significantly better approximation.

In this experiment we approximated  $[6e^x]'(0)$  using the central difference. Now it is not so easy to guess just from a picture whether this estimate overshoots or undershoots. However, our analysis above shows the error to be  $-\frac{1}{6}f'''(a)h^2 + O(h^3)$ , that is,  $-h^2 + O(h^3)$ , so now we expect it to be negative and draw  $-E_h$  to see the magnitude of error (or we could have simply taken the absolute value).

To see how well theory stacks up against reality, we also plotted the graph of  $h^2$  as blue and dashed curve there. At first there is a difference, that's the  $O(h^3)$  part that disappears fast, soon the two curves match.



If we want to derive a formula for a higher derivative or with a better error estimate, we need more values for  $f$ , traditionally we reuse the step  $h$ , so we combine values  $f(a)$ ,  $f(a \pm h)$ ,  $f(a \pm 2h)$ , etc. How many? If we want the  $k$ th derivative, we need to make the coefficient in front of  $f^{(k)}(a)$  equal to 1 and we also have to make the terms before it disappear, that makes  $k$  requirements. This would lead to an error of order  $h$ . We may improve this by removing also terms after  $f^{(k)}(a)$ , leading to more requirements. Thus it follows that we need to know the value of  $f$  at  $k + m$  points if we want to deduce an approximation for  $f^{(k)}(a)$  with error of order  $h^{1+m}$ , and  $f$  needs to be  $k + m + 1$ -times differentiable.

One interesting thing about the procedure we just saw in action is that it allows us to create custom formulas for specific needs.

**Example 4b.a:** We derive a formula for  $f'(a)$  with error of order  $h^3$ , with the restriction that we cannot use values to the left of  $a$ . We will need to know  $f$  at four points, we prepare the necessary Taylor expansions. Since we expect error of order  $h^3$ , we guess that such terms will be needed in the expansion. Since we also expect from experience that the resulting formula will get divided by  $h$  in the process, this means that we actually have to keep track of  $h^4$  in our expansions. One could go all the way up to  $h^5$  to be on the safe side, but we will take the chance and stop at  $h^4$ .

$$\begin{aligned} f(a) &= f(a), \\ f(a+h) &= f(a) + f'(a)h + \frac{1}{2}f''(a)h^2 + \frac{1}{6}f'''(a)h^3 + \frac{1}{24}f''''(a)h^4 + O(h^5), \\ f(a+2h) &= f(a) + f'(a) \cdot 2h + \frac{1}{2}f''(a) \cdot 4h^2 + \frac{1}{6}f'''(a) \cdot 8h^3 + \frac{1}{24}f''''(a) \cdot 16h^4 + O(h^5), \\ f(a+3h) &= f(a) + f'(a) \cdot 3h + \frac{1}{2}f''(a) \cdot 9h^2 + \frac{1}{6}f'''(a) \cdot 27h^3 + \frac{1}{24}f''''(a) \cdot 81h^4 + O(h^5). \end{aligned}$$

A general linear combination is

$$\begin{aligned} Af(a) + Bf(a+h) + Cf(a+2h) + Df(a+3h) \\ = [A+B+C+D]f(a) + [B+2C+3D]f'(a)h + \frac{1}{2}[B+4C+9D]f''(a)h^2 \\ + \frac{1}{6}[B+8C+27D]f'''(a)h^3 + \frac{1}{24}[B+16C+81D]f''''(a)h^4 + O(h^5). \end{aligned}$$

Because we want  $f'(a)$ , we divide by  $h$ .

$$\begin{aligned} \frac{Af(a) + Bf(a+h) + Cf(a+2h) + Df(a+3h)}{h} \\ = [A+B+C+D]\frac{1}{h}f(a) + [B+2C+3D]f'(a) + \frac{1}{2}[B+4C+9D]f''(a)h \\ + \frac{1}{6}[B+8C+27D]f'''(a)h^2 + \frac{1}{24}[B+16C+81D]f''''(a)h^3 + O(h^4). \end{aligned}$$

Our requirements are: The coefficient at  $f'(a)$  must be 1, the term before it must disappear, and because we want to improve the usual error estimate  $h^1$  by two powers, we have to get rid of two

terms after  $f'(a)$ . The resulting system is

$$\begin{aligned} A + B + C + D &= 0 \\ B + 2C + 3D &= 1 \\ B + 4C + 9D &= 0 \\ B + 8C + 27D &= 0 \end{aligned} \implies A = \frac{-11}{6}, B = \frac{18}{6}, C = \frac{-9}{6}, D = \frac{2}{6}.$$

We obtain the formula

$$f'(a) = \frac{-11f(a) + 18f(a+h) - 9f(a+2h) + 2f(a+3h)}{6h} - \frac{1}{4}f''''(a)h^3 + O(h^4).$$

We note that after substituting the right values for  $A, B, C, D$ , the term with  $h^3$  did not disappear, so  $O(h^3)$  is indeed the best general estimate for the error that we can have. If we did not care for knowing that our estimate is the best possible, we could have used Taylor expansions with one less term, that is, ending with with  $+O(h^4)$ .

△

For typical situations one can find appropriate formulas in books. In particular, the differences introduced above can be generalized to yield an arbitrary derivative.

**Fact 4b.2.**

For  $k \in \mathbb{N}$  and functions with derivatives of sufficient order we have

$$f^{(k)}(a) = \frac{1}{h^k} \sum_{i=1}^k (-1)^i \binom{k}{i} f(a + (k-i)h) + O(h) \quad (\text{forward difference})$$

$$f^{(k)}(a) = \frac{1}{h^k} \sum_{i=1}^k (-1)^i \binom{k}{i} f(a - ih) + O(h) \quad (\text{backward difference})$$

$$f^{(k)}(a) = \frac{1}{(2h)^k} \sum_{i=1}^k (-1)^i \binom{k}{i} f(a + (k-2i)h) + O(h^2) \quad (\text{central difference})$$

### 4c. Numerical stability

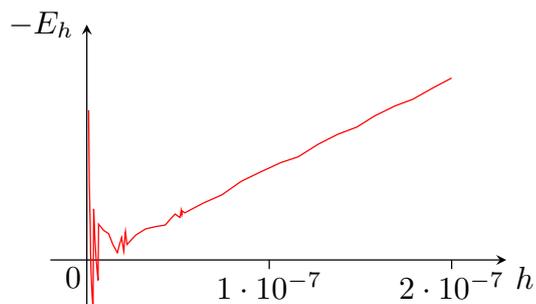
We investigated truncation error, the error caused by the method itself. As we discussed in chapter 3, we also have to worry about numerical errors that come up when we do calculations using computers. How do our formulas react to those errors?

First we look at the formula for forward difference

$$f'(a) \approx \frac{f(a+h) - f(a)}{h}$$

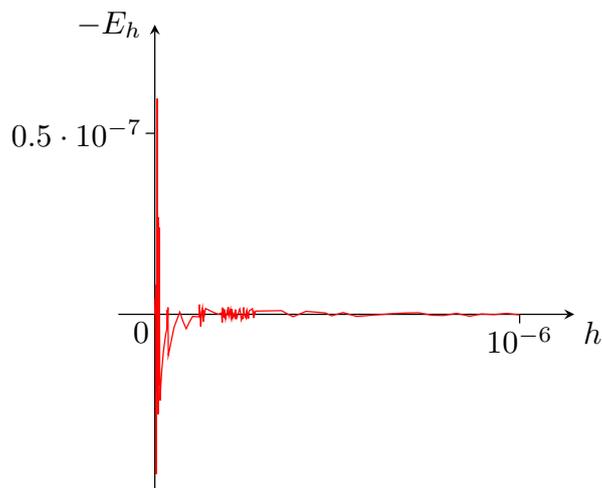
and recall from chapter that division behaves rather well, it does not significantly increase relative error of the outcome compared to the precision of input. However, the difference is a different story, we know that it can be bad. The most dangerous situation is when we subtract numbers that are very close, but that is exactly what we do here. This gives us a reason to worry. Is our fear justified?

We return to our attempt to approximate  $[e^x]'(0)$  using forward difference. We took the graph that we have above and we focused on the area closer to zero. We see that initially the error follows more or less along a straight line, this corresponds to the linear error of the method as analyzed above. But around  $h \approx 0.5 \times 10^{-7}$  the error suddenly starts behaving erratically and can become relatively large. This is exactly what happens when the true result of our calculation gets overshadowed by numerical errors. As we get closer to the origin with  $h$ , the numerical error starts growing so much that it becomes dominant, dwarfing the error of method. Bringing  $h$  even closer to 0 makes results useless.



It should be noted that these calculations are made with relative precision  $10^{-9}$ , so we obviously cannot expect much when  $h$  approaches this value, but here trouble started much earlier. It should be also noted that we do not see how bad the situation really is in this picture. There are some pretty bad errors there, but in fact when plotting, we substituted just some values of  $h$  around the origin. When we zoom in closer and look at the error at more places, we discover significantly larger errors than those shown in the picture. It gets *very* bad, total chaos.

Now we return to the second experiment, when the central difference is applied to  $[6e^x]'(0)$ . Since the error is close to the parabola  $h^2$ , it should be about zero in the depicted region, and it does look so in the right half. But then things go wrong. There is one interesting aspect here. The central difference is supposed to be better than the forward difference, but note that the numerical artefacts start appearing already for  $h \approx 5 \cdot 10^{-7}$ , which is earlier than for the forward difference. Please note that the two axes are not to the same scale, we stretched the vertical axis to show better what is happening.



However (and this is where it gets interesting), since the central difference provides (for the same  $h$ ) significantly better approximations than the forward and backward difference, we can expect to reach the required precision much sooner with the central difference compared to the other two methods, that is, we can stop before the numerical errors kick in. In general, methods of higher order provide good enough answers for significantly larger steps  $h$ , which not only saves our time, but it also makes it less likely that we will need to worry about numerical errors.

It would be nice if we could determine the value of  $h$  that yields the most reliable result. Unfortunately, such analysis is involved even in cases when  $f$  is given by a nice formula and rarely yields more than a rough estimate. We will not go into it in this introductory text.

It should be noted that people do not often need to estimate derivative for its own sake, as the main problem. Rather, the differences are used as ingredients in methods for solving other problems. There the theoretical error estimates deduced here will be very useful.

#### 4d. Richardson extrapolation

We will motivate this section with an example.

**Example 4d.a:** Consider the function  $e^x$ . We use the central difference to approximate  $[e^x]'(0)$  with step  $h = 0.2$  and obtain  $D_{0.2} \approx 1.0066800$ . The error of this approximation is about 0.0067,

but we will now pretend that we do not know that, since we use this approximation of derivative in situations when we cannot get the precise answer.

So the right frame of mind is that we know the function  $f$ , we obtained the number  $D_{0.2}$  and we have no idea how good this approximation is. We know that the error behaves like  $ch^2 + O(h)$ , but we do not know the constants, so this is of no help in judging how good our result is.

Now we use the same method, but we take half of the original step,  $D_{0.1} \approx 1.0016675$ . We may hope that this approximation is better, but even that may be wrong.

However, in most cases the error behaves reasonably, so we will now make some assumptions. The first assumption is that  $D_{0.1}$  is closer to the actual (unknown) answer  $D$  than  $D_{0.2}$ , and the second assumption is that the errors exhibit a certain bias, so both approximations lie to one side of the correct answer. We thus have the following picture:



Now I know that our approximations are actually larger than the precise answer, but we pretend not to know it and there is a 50-50 chance of those approximations being on the left or on the right of  $D$ . I chose the left side just to show that it will not matter.

We are interested in the error  $E$ , and it is time for the third assumption. We will assume that  $E_h \approx ch^2$ . Denoting by  $E$  the error of  $D_{0.1}$ , we can argue as follows:

$$E_{0.2} \approx c(0.2)^2 = c(2 \cdot 0.1)^2 = 4c(0.1)^2 \approx 4E_{0.1} = 4E.$$

When we add this into the picture, we see that the distance between  $D_{0.2}$  and  $D_{0.1}$  is about  $3E$ . In other words,

$$E = \frac{1}{3}(D_{0.1} - D_{0.2}) \approx -0.0016708.$$

If we remember for a second that we know the actual answer, we may notice that the real error is  $E_h = 1 - D_{0.1} \approx -0.0016675$ . It seems that our three assumptions lead to a very good approximation of the error.

When we replace 0.1 in our calculations above with  $h$ , we obtain a formula for the error estimate of  $D_h$  using  $D_h$  and  $D_{2h}$ .

Now there was a bit of hand-waving taking place while deducing this result, but the fact is that we have little else to go by. Thus we are happy that for reasonable functions the following estimate for the error of  $D_h$  works rather well:

$$E = \frac{1}{3}(D_h - D_{2h}).$$

You can draw a second picture, where approximations  $D_h$  are to the right of  $D$ , and see for yourself that the resulting formula is the same. As we will discuss below, this is a very useful result.

Since we have an estimate for the error, we may use it to get a better approximation for the derivative:

$$D \approx D_h + E = D_h + \frac{1}{3}(D_h - D_{2h}) = \frac{4D_h - D_{2h}}{3}.$$

How does it work in our example? We get

$$\frac{4D_{0.1} - D_{0.2}}{3} \approx 0.9999967.$$

Now this is a remarkably good approximation, its error is about 0.0000334, much better than our previous attempts.

△

As we observed before, one problem with numerical methods is that they offer approximate answers, but we usually do not know how good they are. Any result that helps with this is therefore very welcome, even though it is based on some assumptions and guesswork.

We will now explore the ideas from our example, but in a more mathematical way and in general.

Imagine that we have some quantity  $D$  that we want to estimate and some procedure  $D(h)$  that produces approximations of  $D$  based on a parameter  $h > 0$ . Moreover, imagine that for our procedure we have an error estimate

$$D - D(h) = ch^p + O(h^q)$$

for some positive integers  $p < q$ . For instance, the forward difference as a method for estimating derivative satisfies this with  $p = 1$  and  $q = 2$ .

Now imagine that we use this method twice, with step  $2h$  and step  $h$ . We obtain the following.

$$\begin{aligned} D - D(h) &= ch^p + O(h^q), \\ D - D(2h) &= c(2h)^p + O((2h)^q) = c2^p h^p + O(h^q). \end{aligned}$$

We used the fact that  $O(\alpha \cdot h^q) = O(h^q)$ . There are two unknown quantities in these equations,  $c$  and  $D$ . The latter is of interest, so we use elimination to get rid of  $c$ .

$$\begin{aligned} c2^p h^p &= 2^p D - 2^p D(h) + O(h^q) \\ c2^p h^p &= D - D(2h) + O(h^q) \end{aligned} \implies D = \frac{2^p D(h) - D(2h)}{2^p - 1} + O(h^q).$$

Again, we used rules for manipulating  $O(h^q)$  to simplify the formulas

We obtained an estimate for the quantity  $D$  whose error is of order  $h^q$ . Note that when we use a method of order 2 (which is the case with the central difference), this formula fits with the one we deduced in our example above.

Having an approximation for  $D$ , we can substitute into the formula for error, obtaining

$$D - D(h) = \frac{2^p D(h) - D(2h)}{2^p - 1} - D(h) + O(h^q) = \frac{D(h) - D(2h)}{2^p - 1} + O(h^q).$$

Again, for  $p = 2$  this agrees with the formula from our example. Our more precise calculation shows how much we can trust this error estimate, it is good up to  $O(h^q)$ . Since the error itself is of order  $h^p$  and  $p < q$ , the uncertainty of the error estimate should be significantly smaller than the estimate itself.

To sum up, using two approximations  $D(2h)$  and  $D(h)$  we were able to estimate the error of  $D(h)$  and also obtain a (hopefully) better approximation. However, for this better approximation we have no error estimate, so in practice it is often better to just keep  $D(h)$ , since then we can make a judgement on its reliability.

We will capture our conclusions in a theorem. We will allow for other multiples of the step than  $2h$ , although this one is traditional.

**Theorem 4d.1.** (Richardson extrapolation)

Let  $D$  be some quantity and  $D(h)$  some method for producing approximations of  $D$  depending on step  $h > 0$ . Assume that for some positive integers  $p < q$  the following estimate is true:

$$D - D(h) = ch^p + O(h^q).$$

Choose some positive integer  $k > 1$ .

Then the formula  $R(h) = \frac{k^p D(h) - D(kh)}{k^p - 1}$  is an approximation of  $D$  with error of order  $q$ .

Moreover, the expression  $E = \frac{D(h) - D(kh)}{k^p - 1}$  is an estimate of the error  $D - D(h)$  with error of order  $q$ .

Note that  $R(h)$  itself is a method with error of order  $h^q$ , so we can apply Richardson extrapolation to it and obtain an estimate with even better order of error.

**Example 4d.b:** We know that the central difference as an approximation of derivative has error of order  $h^2$ . However, if we take the proof (see section 4b) and repeat it, this time with longer expansions, we discover that all odd powers in the error part disappear due to symmetry. Thus, denoting the central difference by  $D(h)$ , we can write

$$D - D(h) = ch^2 + Ch^4 + O(h^6).$$

Assume that we used this central difference with steps  $4h$ ,  $2h$  and  $h$ . We apply Richardson extrapolation to the first two and also to the second two, recalling that  $p = 2$  we obtain

$$R(2h) = \frac{4D(2h) - D(4h)}{3}, \quad R(h) = \frac{4D(h) - D(2h)}{3}.$$

We know that both these estimates have error of order  $q = 4$ . It can be shown that since the odd powers in the error estimate for  $D$  disappeared, they will also disappear in the error estimate for  $R$ , and so the error of  $R(h)$  is  $dh^4 + O(h^6)$ . Thus we can apply extrapolation to new numbers and obtain

$$R = \frac{2^4 R(h) - R(2h)}{2^4 - 1}.$$

This estimate has error of order  $h^6$ .

We will recycle results from our previous example and add another approximation,  $D_{0.4} \approx 1.0268808$ . Applying Richardson extrapolation to the pair  $D_{0.4}$ ,  $D_{0.2}$  we obtain an error estimate for  $D_{0.2}$ :  $E_{0.2} \approx 0.00673360$  and Richardson approximation  $R(0.2) \approx 0.9999464$ .

From the example above we have  $R(0.1) \approx 0.9999967$ . The second generation Richardson extrapolation then yields

$$\frac{16R(0.1) - R(0.2)}{15} \approx 1.0000000.$$

That is a pretty good estimate.

We could play this game longer: Starting with four estimates with order of error  $h^2$ , create three first generation Richardson extrapolations with error  $O(h^4)$ , then two second generation extrapolations, and after the third stage we would come up with one estimate with error  $O(h^8)$ .

△

We already praised the usefulness of error estimates, but one may ask what is the point of the improved approximation, given that we can get a small error simply by substituting smaller  $h$  to the approximating formula (for instance the central difference). There are two possible reasons for using Richardson extrapolation.

The first reason can be found in the previous section. If we use really small  $h$ , we run into numerical errors. Richardson extrapolation allows us to obtain good estimates without going to really small  $h$ .

The second reason comes up when the evaluation of function values is not simple. It may be a long calculation, it may even force us to do some experiments. With Richardson extrapolation, instead of putting a lot of time or effort into evaluating  $f(a+h)$ , we may simply combine previous results using a few basic operations and obtain a better approximation.

## 4d.2 Working with the error estimate

Error management is an important part of every numerical calculation. Imagine that we are asked for an approximation of some quantity  $D$  with error at most  $\varepsilon$ . We have an approximating procedure  $D(h)$  that, for a certain  $h$ , supplies a number, but typically we do not know how good it is as an approximation. We could try smaller and smaller  $h$  and we expect that eventually our result will be good enough, but when do we stop? How do we find the “good enough” moment?

With a bit of luck our procedure  $D(h)$  satisfies an error estimate as above. Then we can proceed as follows. We calculate  $D(h)$  and a control approximation  $D(2h)$  that allows us to find an error

estimate

$$E_h = \frac{D(h) - D(2h)}{2^p - 1}.$$

If  $|E_h| < \varepsilon$ , then it is very likely that our answer is good enough. What do we do if  $|E_h|$  is too large? We could try using smaller and smaller  $h$  until we are happy, but there is a better way than this trial-and-error procedure.

Since the error is large, we expect that our next step should be smaller, so what can we say about  $D(sh)$ , where  $s \in (0, 1)$ ? Since the error is of order  $p$ , we may estimate as follows:

$$D - D(sh) \approx c(sh)^p = s^p ch^p = s^p E_h.$$

What do we want from this error?

$$|E_{sh}| < \varepsilon \implies s^p |E_h| \leq \varepsilon \implies s < \sqrt[p]{\frac{\varepsilon}{|E_h|}}.$$

So we choose some  $s$  that satisfies this inequality and then we evaluate  $D(sh)$ . There is a very good chance that our approximation will be good enough, we can estimate its error using the familiar estimate with  $D(sh)$  and  $D(2sh)$ .

If we are not happy, we use  $E_{sh}$  to derive a new recommended value for  $s$  and repeat the process.

**Example 4d.c:** We return to our favourite  $[e^x]'(0) = 1 = D$ . We want to get an approximation with error at most  $e = 0.00001$  using the central difference.

We already tried  $D_{0.2}$  and  $D_{0.1}$ , we obtained the error estimate  $|E_{0.1}| \approx 0.0017$ . This is definitely not good enough. A better step  $sh$  is suggested by the inequality

$$s < \sqrt{\frac{0.00001}{E_{0.1}}} \approx 0.077.$$

To be on the safe side we take  $s = 0.07$ , so our next step size will be  $h = 0.07 \cdot 0.1 = 0.007$ .

We find  $D(h) \approx 1.00000815$  and  $D(2h) \approx 1.0000327$ . The error estimate is

$$E_h = \frac{D(h) - D(2h)}{3} = -0.00000817.$$

This is smaller than 0.00001, so we are reasonably confident that our answer  $D(0.007) \approx 1.000008$  is precise enough.

As it happens, we actually know that it is true, so this seems to work.

△

Our exposition of Richardson extrapolation and work with error estimates were very general, because we will want to return to these ideas later, when exploring other topics of numerical mathematics.

As we already mentioned, estimating derivative just for its own sake is not a very frequent task of numerical mathematics. We will use results from this chapter later, but equally importantly, this fairly simple topic allowed us to explore concerns, tools and paths that are taken when developing numerical methods. The insight gained here will be very valuable throughout the rest of this book.

## 5. Approximating integral

In this chapter we address the following question: Given a function  $f$  on an interval  $[a, b]$ , we want to approximate the value of  $I = \int_a^b f(x) dx$ . This is one of the oldest parts of numerical analysis and also goes by the old-fashioned name “numerical quadrature”.

Obviously we will have to know something about  $f$  to do it, namely we will use the knowledge of values of  $f$  at certain points (we say that we sample the function). The natural approach is to space those points evenly, that is, use a so-called equidistant partition of the interval.

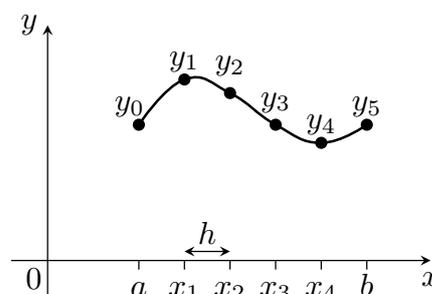
To this end we choose  $n \in \mathbb{N}$  and divide the given interval into  $n$  equal parts, thus creating points  $x_0, x_2, \dots, x_n$ , where  $x_0 = a$  and  $x_n = b$ . It is obvious that the distance between any two neighboring points is  $h = \frac{b-a}{n}$ , we call this the step size and we can expect that it will again in some way characterize the quality of our approximation. We also have a simple formula for these points,

$$x_i = a + hi, \quad i = 0, \dots, n,$$

and we will use the corresponding values  $f(x_i)$  to approximate the value of  $I$ .

We know that this integral is actually the area of the region under the graph of  $f$  on  $I$ . When we draw vertical lines through points  $x_i$ , this region is split into vertical strips. The starting point is to approximate the area of one such strip, that is, the value of  $\int_{x_{i-1}}^{x_i} f(x) dx$ , using the known values of  $f$  related to this subinterval, which means that we use the knowledge of  $f$  at the endpoints.

The desired approximation is obtained by summing up approximations for all strips. Similarly, we will determine the error of this approximation by summing up errors that were made in approximating the individual strips. We will first look in detail at the simplest method and develop key notions and tools. In the next section we will apply those ideas to improve our approximations.



### 5a. Rectangle rules

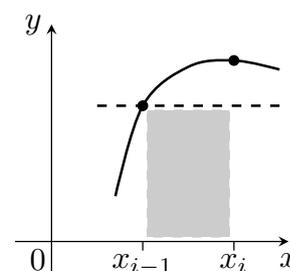
There are several definitions of a definite integral, the most popular is due to Riemann and uses approximation by rectangles based on equidistant division of the integration interval, which nicely ties in with our setup. It thus seems natural to approximate each strip using a rectangle whose width is obviously given by the distance between dividing points, we call it  $h$  here.

The height of this rectangle must somehow relate to values of  $f$  on that strip, and we have two available, at the left and right ends. There is no reason to prefer one or the other, so we simply pick one, and it will be convenient to make this choice common for all strips.

If we decide to use values on the left of each strip, we obtain the basic approximation

$$\int_{x_{i-1}}^{x_i} f(x) dx \approx f(x_{i-1}) \cdot (x_i - x_{i-1}) = f(x_{i-1})h.$$

Summing up we get an approximation for the whole given integral. Similarly we deduce a formula that uses information at the right end of each strip.



**Definition 5a.1.**

Let  $f$  be a function integrable on an interval  $[a, b]$ . For  $n \in \mathbb{N}$  denote  $h = \frac{b-a}{n}$  and consider points  $x_i = a + ih$ .

By the **method of left rectangles** or **rectangle rule with left endpoints**

for approximating the integral  $\int_a^b f(x) dx$  we mean the formula

$$R_l(n) = h \sum_{i=0}^{n-1} f(x_i) = h[f(x_0) + \cdots + f(x_{n-1})].$$

By the **method of right rectangles** or **rectangle rule with right endpoints**

for approximating the integral  $\int_a^b f(x) dx$  we mean the formula

$$R_r(n) = h \sum_{i=1}^n f(x_i) = h[f(x_1) + \cdots + f(x_n)].$$

The basic part that determines an integration method is traditionally called a **panel**. For the rectangle method, panels correspond to strips. Note that some authors use the name “rectangle formula” for the approximation over one panel. When they want to refer to the formula for the whole integration interval, they talk of “compound rectangle formula”. Similar terminology applies to other methods described below.

As we will see, the rectangle rule can be easily significantly improved with just a minor modification of the formula. Why is it the traditional starting point for numerical integration then? Because due to its simplicity we can analyze it without technical difficulties, which allows us to focus on notions and ideas instead of calculations. It is also a natural approach due to its direct relation to the definition of Riemann definite integral. We will now investigate this method and introduce key concepts related to numerical integration.

We start with the familiar question: How precise is our approximation? As usual we start with the **truncation error** (error of the method), where we assume precise calculations and only ask about the error caused by evaluating areas of rectangles instead of actual regions.

The traditional approach is to first focus on one panel, that is, determine local error. In order to simplify notation we forget about the general setup for now and focus on an integral of the form  $\int_c^{c+h} f(x) dx$ , which we want to approximate using one rectangle. If we fix its height based on the value  $f(c)$ , we are using exactly the left rectangle formula.

**Fact 5a.2.**

Consider a function  $f$  on an interval  $[c, c+h]$ , where  $h > 0$ . If  $f$  is differentiable on this interval, then there is  $\xi \in [c, c+h]$  such that

$$\int_c^{c+h} f(x) dx - f(c)h = \frac{1}{2}f'(\xi)h^2.$$

**Proof:** Using the Mean Value Theorem, for every  $x \in [c, c+h]$  there must be some  $\xi_x \in (c, c+h)$

so that  $f(x) - f(c) = f'(\xi_x) \cdot (x - c)$ . Then we can write

$$\begin{aligned} \int_c^{c+h} f(x) dx - f(c)h &= \int_c^{c+h} f(x) dx - \int_c^{c+h} f(c) dx = \int_c^{c+h} f(x) - f(c) dx \\ &= \int_c^{c+h} f'(\xi_x) \cdot (x - c) dx. \end{aligned}$$

The formula  $x \mapsto f'(\xi_x)$  defines a function that can be shown to be continuous, so we can use the Mean Value Theorem for integrals and deduce that there must be some  $x_0 \in (c, c + h)$  such that

$$\int_c^{c+h} f'(\xi_x) \cdot (x - c) dx = f'(\xi_{x_0}) \int_c^{c+h} (x - c) dx = f'(\xi_{x_0}) \frac{1}{2} h^2.$$

Denoting  $\xi_{x_0} = \xi$  we obtain the desired formula. □

In a similar way one can prove an analogous estimate for the method of right rectangles. We will prove that estimate later (5a), when we show an alternative approach to local error estimates.

Having the error over one generic panel, we return to our setup with partition  $\{x_i\}$  and apply the error formula to each panel. Summing up the individual errors we obtain the total error for the whole integral.

**Theorem 5a.3.**

Consider a function  $f$  on an interval  $[a, b]$ , denote  $M_1 = \max_{x \in [a, b]} |f'(x)|$ . If we

approximate the integral  $I = \int_a^b f(x) dx$  using a rectangle rule, then for all  $n \in \mathbb{N}$  we have the following error estimates:

$$\begin{aligned} \text{(i)} \quad |I - R_l(n)| &\leq \frac{1}{2}(b - a)^2 M_1 \frac{1}{n}, \\ \text{(ii)} \quad |I - R_r(n)| &\leq \frac{1}{2}(b - a)^2 M_1 \frac{1}{n}. \end{aligned}$$

**Proof:** (i): Given  $n \in \mathbb{N}$ , we set up  $h = \frac{b-a}{n}$  and the partition  $\{x_i\}$ . Applying Fact 5a.2 to intervals  $[x_{i-1}, x_i]$  we obtain

$$\int_{x_i}^{x_{i+1}} f(x) dx - f(x_i)h = \frac{1}{2} f'(\xi_i) h^2, \quad i = 0, \dots, n - 1.$$

Here  $\xi_i$  are some numbers from appropriate intervals. Now we add all these equalities and note that  $\sum f(x_i)h = R_l(n)$ , so

$$\int_a^b f(x) dx - R_l(n) = \frac{1}{2} h^2 \sum_{i=0}^{n-1} f'(\xi_i).$$

Using our assumption on  $f'$ , we can estimate

$$\begin{aligned} \left| \int_a^b f(x) dx - R_l(n) \right| &\leq \frac{1}{2} h^2 \sum_{i=0}^{n-1} |f'(\xi_i)| \leq \frac{1}{2} h^2 \sum_{i=0}^{n-1} M_1 \\ &= \frac{1}{2} h^2 n M_1 = \frac{1}{2} \frac{(b - a)^2}{n^2} n M_1 = \frac{1}{2} (b - a)^2 M_1 \frac{1}{n}. \end{aligned}$$

(ii) is proved similarly. □

**5a.4 Remark:** One can provide a precise answer for the total error as well. Note that  $\frac{1}{n} \sum_{i=0}^{n-1} f'(\xi_i)$  is the average of certain values of  $f'$ , which means that its outcome is definitely between the minimum and maximum of  $f'$  on  $[a, b]$ . Since  $f'$  as derivative has the Intermediate value property, there must be  $\xi \in [a, b]$  such that  $\frac{1}{n} \sum_{i=0}^{n-1} f'(\xi_i) = f'(\xi)$ . Consequently, we can write

$$\int_a^b f(x) dx - R_l(n) = \frac{1}{2} h^2 \sum_{k=0}^{n-1} f'(\xi_k) = \frac{1}{2} h^2 n f'(\xi) = \frac{1}{2} (b-a)^2 f'(\xi) \frac{1}{n}.$$

△

Using  $n = \frac{b-a}{h}$  we can rewrite the upper error estimate as  $|E| \leq \frac{1}{2} (b-a) M_1 h$ , which is a form that offers a natural interpretation. We did not put it in the theorem, because unlike the Fact 5a.2 where  $h$  can be anything (even negative with a bit of interpretation), here  $h$  comes from partitions, therefore we first need to have  $n$ . Then it makes sense to write the answer in terms of  $n$  as well. The form with  $n$  also has certain practical advantages, as we will see below.

We will now look closer at the error estimate  $\frac{1}{2} (b-a) M_1 h$ . How does it fit with our expectations? We will now ask our common sense about factors that influence precision.

1. There is no reason why functions should favour one part of real axis over other parts. Thus when we go through functions and approximate integrals, we can expect that on average, the error is created about the same at all places. This means that when we approximate two integrals and one has its integration interval twice as long as the other, then we also expect the error to be twice as large. Another point of view may be that on average, all panels contribute (on average) equally to error, and interval that is twice as long contains twice as many panels.

The same should happen with intervals three times longer, or half as long. In other words, we expect that the error should directly depend on the length of the integration interval. Conclusion: We definitely expect to see the term  $|b-a|$  in our error estimate, and we do see it there.

2. Looking back at one panel, imagine that we fix the height of the rectangle by the left endpoint. How would our function make our guess very wrong? By deviating from this level, which it achieves by going up (or down) quickly. We know what tells us how fast a function changes, its first derivative. Indeed, we do see it in our error estimate. It is not obvious in which particular way the error and growth of the function should be related, the theorem says that here again we have a direct linear dependence.

3. One can expect that the more information we have, the better the approximation. The amount of information is determined by  $n$ , but it is not as simple, because we then take into account also the length of the interval. When an interval is very long, then one would need more points to sample it as well as some shorter interval. Thus a much better indicator is found in  $h$ , which incorporates both the interval length and  $n$  into one handy parameter and tells us how fine our sampling is. We can see the impact of  $h$  also in another way: The wider a strip, the more time the function has to stray away of the right height.

We expect that the smaller the  $h$ , the smaller the error. We see exactly that in our estimate.

We will look at more methods for approximating integrals, and some of these will be better than others. How do we see it in their error estimates? The quality of the method will be seen from the way its error depends on  $h$ . The rectangle rule has an error bound that depends linearly on  $h$ . Thinking back to the previous chapter, we know that if the error depended on  $h^2$ , we could expect a better behaviour. Indeed, estimates that will follow all feature various powers of  $h$ .

**Definition 5a.5.**

We say that a method  $I_n$  for approximating integrals is of **order**  $p$  for  $p \in \mathbb{N}$ , provided that for every (sufficiently smooth) function  $f$  on an interval  $[a, b]$  there is  $C > 0$  such that

$$\left| \int_a^b f(x) dx - I_n \right| \leq C \frac{1}{n^p}$$

for all  $n \in \mathbb{N}$ .

What do we mean by “sufficiently smooth”? Whenever we derive an error estimate, we will have some assumption on smoothness of  $f$ , for instance the estimates for left and right rectangles above require the existence of  $\max |f'|$ , which means that we require  $f$  to be differentiable. For other methods we will require more.

We again obtain a more intuitive form by passing to a version with  $h = \frac{b-a}{n}$ . We see that method of order  $p$  means that  $|I - I_n|$  is  $O(h^p)$  as  $h \rightarrow 0$ . However, there is a technical catch,  $h$  can be taken only from the set  $\{\frac{b-a}{n} : n \in \mathbb{N}\}$ . That’s why it is better to use the form with  $n$  in formal statements.

So far we only analyzed error for one method, but the procedure will be the same also for other methods to come. We first derive a local estimate (for one panel) of order  $O(h^{p+1})$ , the global (total) error then has the order  $nh^{p+1}$ , that is,  $h^p$ .

This also makes sense the other way around. Since there is no preferable treatment for any specific part or function, we expect that a desired global order  $O(h^p)$  spreads evenly among the panels, so one panel should have an error of order  $\frac{1}{n}h^p$ , that is,  $O(h^{p+1})$ .

The version with  $h$  shows why we prefer methods of higher order, the error then disappears faster. We can appreciate higher orders also when we look at the error estimate with  $n$ . For the rectangle rules we have an estimate of the form  $|E_n| \leq \frac{C}{n}$  (we can say that they are of linear order). This suggests that if we double the number of points, the error should be halved.

On the other hand, a second order method with error estimate  $|E_n| \leq \frac{C}{n^2}$  works more efficiently, by doubling the number of points we decrease the error four times. Higher orders are more efficient still.

Of course, it does not work precisely like that. For one, those formulas are just upper estimates, so it may happen that by increasing  $n$  we actually increase the actual error. However, such cases are rare. There is also influence of the constant  $C$ , which is different for various methods and a method of higher order may have much larger  $C$ . However, we already know that when we compare expressions of different orders, then the influence of constants becomes negligible once we pass to really large numbers  $n$  or small  $h$ . Thus the picture that we painted above is perhaps not quite correct, but it sends the right message.

## 5a.6 Developing integration methods by matching Taylor expansions

Recall the main method for deriving derivative approximations from chapter 4. We first decide on what data we want to use. Then we expand them with respect to a common center. We form a general linear combination and then adjust coefficients so that the resulting linear combination yields the thing that we want. Usually it yields also some other terms, these would form the error.

We will now show that this method also works with integration over one panel. However, we have to first make one important step. When we expand our data into Taylor series, we obtain some combinations of derivatives; then it was straightforward to tweak it so that some derivative pops out of it. However, now our target is an integral, which is something that we definitely do not find in a Taylor expansion.

To fix this problem we will also write that integral as an expansion and try to match terms. How do we get such an expansion?

There are two possibilities. One approach is to expand  $f$  first. We will make an exception and use the form that is typical in calculus courses.

$$f(x) = f(c) + f'(c)(x - c) + \frac{1}{2}f''(c)(x - c)^2 + \frac{1}{3!}f'''(c)(x - c)^3 + \frac{1}{4!}f''''(c)(x - c)^4 + \dots$$

Now we integrate.

$$\begin{aligned} \int_c^{c+h} f(x) dx &= \int_c^{c+h} f(c) + f'(c)(x - c) + \frac{1}{2}f''(c)(x - c)^2 + \frac{1}{3!}f'''(c)(x - c)^3 + \frac{1}{4!}f''''(c)(x - c)^4 + \dots dx \\ &= \left[ f(c)(x - c) + \frac{1}{2}f'(c)(x - c)^2 + \frac{1}{3!}f''(c)(x - c)^3 + \frac{1}{4!}f'''(c)(x - c)^4 + \frac{1}{5!}f''''(c)(x - c)^5 + \dots \right]_c^{c+h} \\ &= f(c)h + \frac{1}{2}f'(c)h^2 + \frac{1}{3!}f''(c)h^3 + \frac{1}{4!}f'''(c)h^4 + \frac{1}{5!}f''''(c)h^5 + \dots \end{aligned}$$

This is the expansion that we need.

We will show another approach that is less elementary (and thus perhaps more challenging), but it has the advantage that it can be adjusted to help in other situations as well. We start with an auxiliary function.

$$F(t) = \int_c^{c+t} f(x) dx.$$

By the Fundamental Theorem of Calculus,  $F$  has one more derivative than  $f$  and we have

$$F'(t) = f(c+t), F''(t) = f'(c+t), F'''(t) = f''(c+t), \dots$$

Now we expand  $F$  to Taylor series with center  $a = 0$  and then use our knowledge of derivatives of  $F$ , we also know that  $F(0) = \int_c^c f(x) dx = 0$ . Thus we obtain

$$\begin{aligned} F(t) &= F(0) + F'(0)t + \frac{1}{2}F''(0)t^2 + \frac{1}{3!}F'''(0)t^3 + \dots \\ &= 0 + f(c)t + \frac{1}{2}f'(c)t^2 + \frac{1}{3!}f''(c)t^3 + \dots \end{aligned}$$

When we use this with  $t = h$ , we arrive at the key formula.

$$\int_c^{c+h} f(x) dx = f(c)h + \frac{1}{2}f'(c)h^2 + \frac{1}{3!}f''(c)h^3 + \frac{1}{4!}f'''(c)h^4 + \frac{1}{5!}f^{(4)}(c)h^5 + \dots \quad (\text{I})$$

Note that this formula can be used also with negative  $h$ . We can also truncate it at some  $h^n$  and supply a precise error term in the form of the Lagrange remainder  $\frac{1}{(n+1)!}f^{(n+1)}(c)h^{n+1}$ .

Now we are ready. As an example we will deduce an error formula for the method of right triangles. Namely, we want to approximate  $\int_c^{c+h} f(x) dx$  using the data  $f(c+h)$ . We should look at linear combinations of data, in this case we consider multiples  $Af(c+h)$ . Let's compare what we have and what we want. We already know what order of error to expect, so we will expand just up to  $h^2$ .

$$\text{have: } Af(c+h) = Af(c) + Af'(c)h + A\frac{1}{2}f''(c)h^2 + O(h^3),$$

$$\text{want: } \int_c^{c+h} f(x) dx = f(c)h + \frac{1}{2}f'(c)h^2 + O(h^3).$$

If we want to match the expansions, we have to start from the most important term, which is the linear one. We have no choice, we have to take  $A = 1$  and multiply the first equality by  $h$ . We

compare again.

$$f(c+h)h = f(c)h + f'(c)h^2 + \frac{1}{2}f''(c)h^3 + O(h^4),$$

$$\int_c^{c+h} f(x) dx = f(c)h + \frac{1}{2}f'(c)h^2 + O(h^3).$$

Looking good, it also seems that we could have taken a shorter expansion of  $f(c+h)$ . Anyway, we now subtract the first equality from the second:

$$\int_c^{c+h} f(x) dx - f(c+h)h = -\frac{1}{2}f'(c)h^2 + O(h^3).$$

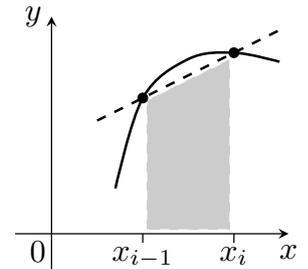
We just confirmed that the local error of the right rectangle formula is  $O(h^2)$ .

## 5b. Standard methods for approximating integral

The rectangle rules are a good start to explore numerical integration, but they are not the best start to actually evaluate integrals. Let's focus on one panel  $\int_c^{c+h} f(x) dx$  again. We have information about  $f$  at both endpoints, but rectangle rules only use one of these two, which means that there definitely is some room for improvement.

Having information about  $f$  at both ends, what is the best approximation for the graph of  $f$ ? The natural approach is to connect the two endpoints by a straight line, thus creating a trapezoid. The well-known formula yields

$$\int_{x_{i-1}}^{x_i} f(x) dx \approx \frac{f(x_{i-1}) + f(x_i)}{2} \cdot (x_i - x_{i-1}) = \frac{1}{2}(f(x_{i-1}) + f(x_i))h.$$



Note that it can take another interpretation: In our previous attempt we did not know whether to take the left or the right endpoint for the height of our rectangles, so we take the average of both.

Summing up all panels we get the resulting formula. Note that in one panel, the values at endpoints are taken with “weight”  $\frac{1}{2}$ . However, most panels are adjacent with other panels, so those endpoints are also used again, each value of  $f$  is used twice with weight  $\frac{1}{2}$ . The obvious exceptions to this rule are the values at the very ends of the integration interval. Thus the values of  $f$  are used with weights  $\frac{1}{2} 1 1 \cdots 1 \frac{1}{2}$ . It is traditional to factor out a suitable fraction so that the weights in the formula become integers, in this case  $1 2 2 \cdots 2 1$ .

### Definition 5b.1.

Let  $f$  be a function integrable on an interval  $[a, b]$ . For  $n \in \mathbb{N}$  denote  $h = \frac{b-a}{n}$  and consider points  $x_i = a + ih$ .

By the **trapezoid rule** for approximating the integral  $\int_a^b f(x) dx$  we mean the formula

$$T(n) = \frac{1}{2}h \left[ \sum_{i=0}^{n-1} f(x_i) + \sum_{i=1}^n f(x_i) \right] = \frac{1}{2}h \left[ f(x_0) + \sum_{i=1}^{n-1} 2f(x_i) + f(x_n) \right].$$

Since we use more information, we expect better results.

**Fact 5b.2.**

Consider a function  $f$  on an interval  $[c, c + h]$ , where  $h > 0$ . If  $f$  is twice differentiable on this interval, then there is  $\xi \in [c, c + h]$  such that

$$\int_c^{c+h} f(x) dx - \frac{1}{2}(f(c) + f(c+h))h = \frac{1}{12}f''(\xi)h^3.$$

We will not prove the precise result with  $f''(\xi)$  shown in the statement, because it is not of much practical use anyway. The important part is the order. We will use the approach from section 5a, it will also show that the trapezoid formula is the right one if we have two pieces of data.

**Proof:** We want to approximate

$$\int_c^{c+h} f(x) dx = f(c)h + \frac{1}{2}f'(c)h^2 + \frac{1}{3!}f''(c)h^3 + O(h^4). \quad (*)$$

We chose to expand up to  $O(h^4)$ , we will see whether it is enough (or perhaps too much, but that would be no problem, just a bit of extra work). Our aim is to get the best possible match for the expression on the right using two expansions,

$$f(c) = f(c)$$

$$f(c+h) = f(c) + f'(c)h + \frac{1}{2}f''(c)h^2 + \frac{1}{3!}f'''(c)h^3 + O(h^4).$$

To get  $f(c)h$  we need to multiply both equations by  $h$ , then we form a general linear combination.

$$\begin{aligned} Af(c)h + Bf(c+h)h &= [A+B]f(c)h + Bf'(c)h^2 + \frac{1}{2}Bf''(c)h^3 + \frac{1}{3!}Bf'''(c)h^4 + O(h^5) \\ &= [A+B]f(c)h + Bf'(c)h^2 + \frac{1}{2}Bf''(c)h^3 + O(h^4). \end{aligned}$$

So we could have started with a shorter expansion for  $f(c+h)$ , but better safe than sorry. Now we want the expression on the right to be the best match for the expansion (\*) of the integral, and the two parameters allow us to state two requirements.

$$\begin{aligned} A + B &= 1 \\ B &= \frac{1}{2} \implies A = B = \frac{1}{2}. \end{aligned}$$

Thus

$$\frac{1}{2}f(c)h + \frac{1}{2}f(c+h)h = f(c)h + \frac{1}{2}f'(c)h^2 + \frac{1}{4}f''(c)h^3 + O(h^4).$$

Subtracting this from the expansion of the integral we obtain

$$\begin{aligned} \int_c^{c+h} f(x) dx - \frac{1}{2}(f(c) + f(c+h))h &= \frac{1}{3!}f''(c)h^3 + O(h^4) - \frac{1}{4}f''(c)h^3 - O(h^4) \\ &= -\frac{1}{12}f''(c)h^3 + O(h^4). \end{aligned}$$

We confirmed that the local error is of order  $O(h^3)$  and we also see that we cannot really expect anything better, unless the function  $f$  satisfies  $f''(c) = 0$ . We also obtained the right constant  $\frac{1}{12}$  for the more precise form of the estimate. □

The rest is now easy, just like before we add local error estimates and lose one power of  $h$ .

**Theorem 5b.3.**

Consider a function  $f$  on an interval  $[a, b]$ , denote  $M_2 = \max_{x \in [a, b]} |f''(x)|$ . If we approximate the integral  $I = \int_a^b f(x) dx$  using the trapezoid rule, then for all  $n \in \mathbb{N}$  we have the following error estimate:

$$|I - T(n)| \leq \frac{1}{12}(b-a)^3 M_2 \frac{1}{n^2}.$$

In other words, error is bounded by  $\frac{1}{12}(b-a)M_2h^2 = O(h^2)$ . As expected, trapezoids approximate functions better and lead to a better method.

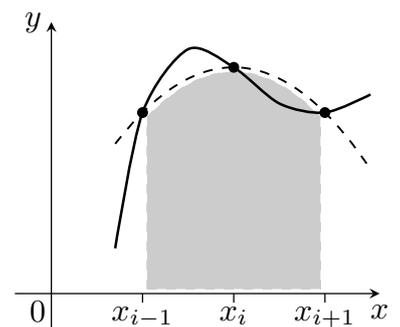
Note that it is natural to see the second derivative in the estimate. The growth or decrease of  $f$  is not a problem now, because we approximate using slanted lines and these can adjust. What those lines cannot do well is to adjust to curved graphs. The more a function curves, the harder it is for the trapezoids to match it, and the second derivative is exactly the one that talks about “twistiness” of a graph.

If we want an even better method, we have to supply more data to match more terms in the series (I). If things work as they seem, we would need three pieces of data to arrive at a method of order three. Where do we get them?

When we decided at the beginning on a certain setup (equidistant partition and a corresponding sampling of  $f$ ), we closed the door on sampling the function also someplace else. So if we want to have more pieces of data in one panel, we have to use more strips.

In order to have three values of  $f$  we need to joint two neighboring strips as a basic panel. For that to work we need  $n$  to be even, but that is no problem as  $n$  is ours to choose.

So we take two adjacent strips, in other words, we ask the question what kind of approximation we can construct when given three values  $f(x_{i-1})$ ,  $f(x_i)$ , and  $f(x_{i+1})$ . We could try a broken line (two segments), but that would lead to the trapezoid rule again. A more interesting idea is to put a piece of parabola through these points, since we know that a parabola is uniquely determined by three points, what a coincidence.



Determining coefficients of the right parabola is routine, interpolation theory (or common sense) lead to

$$\begin{aligned} p(x) &= \frac{f(x_{i-1})}{(x_{i-1} - x_i)(x_{i-1} - x_{i+1})}(x - x_i)(x - x_{i+1}) + \frac{f(x_i)}{(x_i - x_{i-1})(x_i - x_{i+1})}(x - x_{i-1})(x - x_{i+1}) \\ &\quad + \frac{f(x_{i+1})}{(x_{i+1} - x_{i-1})(x_{i+1} - x_i)}(x - x_{i-1})(x - x_i) \\ &= \frac{f(x_{i-1})}{2h^2}(x - x_i)(x - x_{i+1}) + \frac{f(x_i)}{h^2}(x - x_{i-1})(x - x_{i+1}) + \frac{f(x_{i+1})}{2h^2}(x - x_{i-1})(x - x_i). \end{aligned}$$

Then we integrate it over the interval  $[x_{i-1}, x_{i+1}]$ , obtaining

$$\begin{aligned} \int_{x_{i-1}}^{x_{i+1}} f(x) dx &\approx \int_{x_{i-1}}^{x_{i+1}} \frac{f(x_{i-1})}{2h^2}(x-x_i)(x-x_{i+1}) + \frac{f(x_i)}{h^2}(x-x_{i-1})(x-x_{i+1}) \\ &\quad + \frac{f(x_{i+1})}{2h^2}(x-x_{i-1})(x-x_i) dx \\ &= \left[ \frac{f(x_{i-1})}{2h^2} \left( \frac{1}{2}(x-x_i)^2(x-x_{i+1}) - \frac{1}{6}(x-x_i)^3 \right) \right. \\ &\quad + \frac{f(x_i)}{h^2} \left( \frac{1}{2}(x-x_{i-1})(x-x_{i+1})^2 - \frac{1}{6}(x-x_{i+1})^3 \right) \\ &\quad \left. + \frac{f(x_{i+1})}{2h^2} \left( \frac{1}{2}(x-x_{i-1})^2(x-x_i) - \frac{1}{6}(x-x_{i-1})^3 \right) \right]_{x_{i-1}}^{x_{i+1}} \\ &= \frac{h}{3} [f(x_{i-1}) + 4f(x_i) + f(x_{i+1})]. \end{aligned}$$

We see that the weights of function values are (after factoring out the fraction) 1 4 1. Now we sum up the panels. As usual, the values where panels meet are used twice, this concerns the values whose weights with respect to one panel are 1. Thus in the global formula their weights will be 2, with the obvious exceptions of the values at the very end of integration interval. In this way we arrive at the pattern of weights

$$1 \ 4 \ 2 \ 4 \ 2 \ 4 \ \cdots \ 2 \ 4 \ 2 \ 4 \ 1.$$

**Definition 5b.4.**

Let  $f$  be a function integrable on an interval  $[a, b]$ . For  $n \in \mathbb{N}$ ,  $n$  even, denote  $h = \frac{b-a}{n}$  and consider points  $x_i = a + ih$ .

By the **Simpson rule** for approximating the integral  $\int_a^b f(x) dx$  we mean the formula

$$\begin{aligned} S(n) &= \frac{1}{3}h \left[ f(x_0) + 4 \sum_{i=1}^{n/2} f(x_{2i-1}) + 2 \sum_{i=1}^{n/2-1} f(x_{2i}) + f(x_n) \right] \\ &= \frac{1}{3}h \left[ f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + \cdots + 4f(x_{n-1}) + f(x_n) \right]. \end{aligned}$$

The local error is surprisingly good.

**Fact 5b.5.**

Consider a function  $f$  on an interval  $[c-h, c+h]$ , where  $h > 0$ . If  $f$  is four times differentiable on this interval, then there is  $\xi \in [c-h, c+h]$  such that

$$\int_{c-h}^{c+h} f(x) dx - \frac{1}{3}(f(c-h) + 4f(c) + f(c+h))h = -\frac{1}{90}f''''(\xi)h^5.$$

We have shown the traditional precise form of the error. As usual, we prefer to prove just the correct order estimate, but as a reward we will see that the Simpson rule is the right way to use three pieces of data.

**Proof:** First we need to derive a series for the integral  $\int_{c-h}^{c+h} f(x) dx$ . We want to be on the safe side, so we will expand up to the power  $h^5$ . Rather than doing it directly, we prefer to put it

together using the results that we already have.

$$\begin{aligned} \int_{c-h}^{c+h} f(x) dx &= \int_{c-h}^c f(x) dx + \int_c^{c+h} f(x) dx = - \int_c^{c-h} f(x) dx + F(h) = -F(-h) + F(h) \\ &= F(h) - F(-h). \end{aligned}$$

We thus obtain the desired expression by subtracting the expansions

$$\begin{aligned} F(h) &= f(c)h + \frac{1}{2}f'(c)h^2 + \frac{1}{3!}f''(c)h^3 + \frac{1}{4!}f'''(c)h^4 + \frac{1}{5!}f^{(4)}(c)h^5 + O(h^6), \\ F(-h) &= -f(c)h + \frac{1}{2}f'(c)h^2 - \frac{1}{3!}f''(c)h^3 + \frac{1}{4!}f'''(c)h^4 - \frac{1}{5!}f^{(4)}(c)h^5 + O(h^6). \end{aligned}$$

We obtain

$$\int_{c-h}^{c+h} f(x) dx = 2f(c)h + \frac{2}{3!}f''(c)h^3 + \frac{2}{5!}f^{(4)}(c)h^5 + O(h^6) \quad (*)$$

The expression on the right is our target. Note that some terms already disappeared, which is a nice bonus caused by the symmetry of our situation. What data are available?

$$\begin{aligned} f(c+h) &= f(c) + f'(c)h + \frac{1}{2}f''(c)h^2 + \frac{1}{3!}f'''(c)h^3 + \frac{1}{4!}f^{(4)}(c)h^4 + O(h^5) \\ f(c) &= f(c) \end{aligned}$$

$$f(c-h) = f(c) - f'(c)h + \frac{1}{2}f''(c)h^2 - \frac{1}{3!}f'''(c)h^3 + \frac{1}{4!}f^{(4)}(c)h^4 + O(h^5)$$

We hope that just like in the proof for the trapezoid rule, it will be enough to have degree smaller by one. We will need  $f(c)h$ , so we multiply these three expansions by  $h$  and form a linear combination.

$$\begin{aligned} Af(c+h) + Bf(c)h + Cf(c-h)h &= [A+B+C]f(c)h + [A-C]f'(c)h^2 + \frac{1}{2}[A+C]f''(c)h^3 \\ &\quad + \frac{1}{3!}[A-C]f'''(c)h^4 + \frac{1}{4!}[A+C]f^{(4)}(c)h^5 + O(h^6). \end{aligned}$$

We can make three demands, so we will try to match the the first three terms of this expansion with the expansion in (\*).

$$\begin{aligned} A+B+C &= 2 \\ A-C &= 0 \implies A = \frac{1}{3}, B = \frac{4}{3}, C = \frac{1}{3}. \\ \frac{1}{2}(A+C) &= \frac{2}{3!} \end{aligned}$$

Thus

$$\frac{1}{3}f(c+h) + \frac{4}{3}f(c)h + \frac{1}{3}f(c-h)h = 2f(c)h + \frac{2}{3!}f''(c)h^3 + \frac{1}{4!}\frac{2}{3}f^{(4)}(c)h^5 + O(h^6).$$

We subtract this equality from (\*),

$$\begin{aligned} \int_{c-h}^{c+h} f(x) dx - \frac{1}{3}h[f(c-h) + 4f(c) + f(c+h)] &= \frac{2}{5!}f^{(4)}(c)h^5 + O(h^6) - \frac{1}{4!}\frac{2}{3}f^{(4)}(c)h^5 - O(h^6) \\ &= -\frac{1}{90}f^{(4)}(c)h^5 + O(h^6). \end{aligned}$$

This concludes the proof

□

We confirmed the surprising fact that the local error is of order  $h^5$ , but with three pieces of information we expected  $h^4$ . We saw the reason above: Due to symmetry of the situation, the term with  $h^4$  disappeared from our expansion of the integral, and in the expansion of data it shared

coefficient with a term that we needed to make zero.

This does not invalidate our feeling that  $p$  pieces of data in one panel lead to a method of order  $p$ , it's just that sometimes things are better.

Now we sum up the errors for all panels. We have to be careful because this time there are only  $\frac{1}{2}n$  of them. The usual tricks lead to the desired estimate.

**Theorem 5b.6.**

Consider a function  $f$  on an interval  $[a, b]$ , denote  $M_4 = \max_{x \in [a, b]} |f''''(x)|$ . If we

approximate the integral  $I = \int_a^b f(x) dx$  using the Simpson rule, then for all  $n \in \mathbb{N}$  we have the following error estimate:

$$|I - S(n)| \leq \frac{1}{180} (b - a)^5 M_4 \frac{1}{n^4}.$$

That is, the global error is bounded by  $\frac{1}{180}(b - a)M_4h^4$ .

The Simpson method is probably the most popular entry method for approximating integrals, it combines simplicity with better-than-expected performance. Often it is enough. When better precision is required, people prefer to use other approaches (see below) rather than going for higher order of accuracy with our approach. Other methods may be also more suitable to specific types of integrals.

We do have a method of fourth order, but it was by a happy coincidence, so to see how it goes we will try to develop such a method in a standard way. We expect to need four pieces of data for such a method, which can be done by taking three adjacent strips as a basic panel.

Geometrically, three adjacent strips provide four points on a graph, these then determine a unique polynomial of order three going through them.

The calculations are again routine (and longer) and assuming that we use slices determined by  $x_{i-1}$ ,  $x_i$ ,  $x_{i+1}$ , and  $x_{i+2}$ , we obtain the following approximation for one panel.

$$\int_{x_{i-1}}^{x_{i+2}} f(x) dx \approx \frac{3}{8} (f(x_{i-1}) + 3f(x_i) + 3f(x_{i+1}) + f(x_{i+2})) \cdot h.$$

Summing up we get the desired rule.

**Definition 5b.7.**

Let  $f$  be a function integrable on an interval  $[a, b]$ . For  $n \in \mathbb{N}$ ,  $n$  divisible by three, denote  $h = \frac{b-a}{n}$  and consider points  $x_i = a + ih$ .

By the **Simpson 3/8 rule** for approximating the integral  $\int_a^b f(x) dx$  we mean the formula

$$\begin{aligned} S(n) &= \frac{3}{8} h \left[ f(x_0) + 3 \sum_{i=1}^{n/3} f(x_{3i-2}) + 3 \sum_{i=1}^{n/3} f(x_{3i-1}) + 2 \sum_{i=1}^{n/3-1} f(x_{3i}) + f(x_n) \right] \\ &= \frac{3}{8} h \left[ f(x_0) + 3f(x_1) + 3f(x_2) + 2f(x_3) \right. \\ &\quad \left. + 3f(x_4) + 3f(x_5) + 2f(x_6) + \cdots + 3f(x_{n-1}) + f(x_n) \right]. \end{aligned}$$

The order of the error is as expected. Local error on one panel is given by the formula

$$E = -\frac{3}{80} f''''(\xi) h^5 + O(h^6).$$

For the global error we have

**Theorem 5b.8.**

Consider a function  $f$  on an interval  $[a, b]$ , denote  $M_4 = \max_{x \in [a, b]} |f''''(x)|$ . If we

approximate the integral  $I = \int_a^b f(x) dx$  using the Simpson 3/8 rule, then we have the following error estimate:

$$|I - S_{3/8}(n)| \leq \frac{1}{80} (b - a)^5 M_4 \frac{1}{n^4}.$$

That is, we have the bound  $\frac{1}{80}(b - a)M_4h^4 = O(h^4)$ .

The error is of the same order as for the Simpson method, but the different constant means that it is a bit less than half of that. However, we pay for this slight improvement by a more complicated formula, which may or may not be a factor. In most applications people seem to prefer the simpler option, but not always. For instance, the Simpson 3/8 rule is the standard way to estimate volume of ship hulls used in by British naval authorities.

It is possible to play this game further: Connect four panels, use five values to determine a fourth order polynomial etc., but people do not find it worthwhile. In particular, the theoretical improvement in error order requires that our demands on differentiability increase. However, this is not always satisfied, and experience suggests that for less smooth functions those more advanced schemes actually perform worse.

### 5b.9 Numerical stability

So far we compared methods using their theoretical error, that is, the error of the method. How do they fare in real life, when used on computers that work with limited precision?

From the point of view of error propagation these formulas are quite safe. Indeed, multiplication by a constant preserves relative error. Addition can be a problem if we add numbers of almost equal magnitude but with opposite signs (that is, if we in fact subtract), but here we add values of a function taken at near points, so extreme changes are not expected.

Is there any concern arising from the fact that we usually evaluate these formulas in computers where the floating point format causes round-off errors? We know that there is a possible problem when adding numbers with very different magnitudes, and that indeed is a very real possibility here. Not that we would expect the function to change significantly on short notice, the individual entries in the sum are usually comparable, but when  $n$  is large, we are adding really lots of these numbers and there is a danger that the partial sum becomes so large than the next numbers we add become lost. We addressed this problem in chapter 3 and noted that there are some approaches that can add numbers in a safer manner, for instance the Kahan summation. It would be prudent to incorporate such measures when coding integrating formulas.

### 5b.10 Overview

All the methods we saw so far had the same framework. We sampled the given function  $f$  at regular intervals, obtaining values  $f(x_0), \dots, f(x_n)$ . We were looking for an estimate of an integral that satisfies the general formula

$$\int_{x_0}^{x_n} f(x) dx = h \sum_{i=0}^n w_i f(x_i)$$

for some weights  $w_i$ . It is curious that just by choosing more and more complicated patterns for  $w_i$  we obtained more and more precise approximations.

This general approach is called **Newton-Cotes** formulas. There are two general types, closed formulas use information about  $f$  at the ends of the integration interval. Our methods are like that. There are also open formulas where the endpoints are not used, we will see one such method below.

Note that it is useful if all weights are not negative, because this minimizes the chance that we will have a large numerical error due to subtracting similar numbers. Our methods were like that. Note that if we tried for higher and higher orders by joining more strips, around order 11 we would start encountering negative weights, one more reason not to go this way.

The weights should also satisfy the condition  $\sum_{i=0}^n w_i = n$ . This is actually fairly easy to show. Given that a constant function  $f(x) = 1$  is about the nicest function one can find, we expect that reasonable integration formulas should provide exact answer. So on the one hand we have

$$\int_a^b 1 dx = b - a$$

and on the other hand we have

$$h \sum_{i=0}^n w_i f(x_i) = h \sum_{i=0}^n w_i.$$

Thus

$$b - a = h \sum_{i=0}^n w_i \implies \sum_{i=0}^n w_i = \frac{b - a}{h} = n.$$

## 5c. Romberg integration

We deduced error estimates for our methods, but they had one crucial weakness, they used data that are not available in a typical case. When we face a function  $f$  about which we have little information (apart from the ability to evaluate it), we can choose  $n$  and calculate some approximation for an integral using our favourite method, but we do not really know how good our result is.

There is an approach that can help, and we saw it in chapter 4, namely Richardson extrapolation (section 4d). However, we have to adjust the setting, because now we cannot choose the step  $h$  as we want, we have to start with number of partitions  $n$ . Other than that, the framework fits. We have a number  $I$  that we want to approximate (the integral) and we have a certain method  $I_n$  that produces approximations.

We also need to get an approximation with half the step, which we obtain by doubling the  $n$ . Richardson extrapolation works with the error estimate  $I - I_n = Ch^p + O(h^q)$ , which we easily rewrite as  $I - I_n = \frac{C}{n^p} + O(1/n^q)$ .

Now we can use the results from section 4d, or we can deduce the results in analogous ways, this time working with  $n$  as parameter. We start with two runs:

$$\begin{aligned} I - I_n &= \frac{C}{n^p} + O(1/n^q), \\ I - I_{2n} &= \frac{C}{2^p n^p} + \frac{1}{2^q} O(1/n^q). \end{aligned}$$

Now we eliminate  $C$ ,

$$\begin{aligned} \frac{C}{n^p} &= I - I_n + O(1/n^q) \\ \frac{C}{n^p} &= 2^p I - 2^p I_{2n} + O(1/n^q) \end{aligned} \implies I = \frac{2^p I_{2n} - I_n}{2^p - 1} + O(1/n^q)$$

and obtain an estimate for  $I$  whose order is  $h^q$ . We also obtain an error estimate for  $I_{2n}$

$$E_{2n} = \frac{I_{2n} - I_n}{2^p - 1}.$$

We already mentioned that this error estimate, while not precise, is good enough to be very useful.

We have a little problem here. When estimating the derivative, we indeed had  $D - D(h) = Ch^p + O(h^q)$ . What made the procedure work is the fact that this  $C$  is common to all  $h$ , so we were able to eliminate it. When it comes to integration integration, we did obtain error estimates of this form, but  $C$  featured the value of a certain derivative at a point  $\xi$  which depended on  $n$ . This means that  $C$  is no longer a fixed constant, which means that our justification for Richardson extrapolation no longer works.

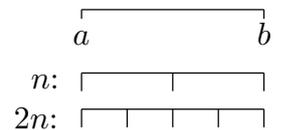
On the other hand, for reasonably well-behaved functions this constant  $C$  does not change too much, so while it is not as bulletproof as when we were estimating derivatives, Richardson extrapolation does give useful results for integration as well.

This is a moment when one would really like to see some proper mathematical statement that would quantify the situation, accompanied by a reasonably complete proof. The fact is, I have yet to find a textbook on numerical analysis that would offer this. A proper mathematical treatment of Richardson extrapolation for integrals is so involved that authors prefer to give just the basic idea, which is what we did here as well.

We appreciate that Richardson extrapolation provides a better approximation and also an error estimate, but its beauty when it comes to integration actually goes further than this. It turns out that we get all those goodies without really needing to do extra work.

How come? We need two approximations,  $I_n$  and  $I_{2n}$ . For the first one we need to know values of  $f$  at points  $y_0, \dots, y_n$ , while for the second one we need to know values of  $f$  at points  $x_0, \dots, x_{2n}$ . It should be noted that it is exactly this, the need to evaluate  $f$  at certain points, that takes up most of the time. So we definitely do not like the idea that we would need to do some extra evaluations.

Which brings us to the key observation. Note that both  $\{y_i\}$  and  $\{x_j\}$  are partitions of the same interval  $[a, b]$ . For  $I_n$  we use the step  $h = \frac{b-a}{n}$ , while for  $I_{2n}$  we use the step  $k = \frac{b-a}{2n}$ , so  $h = 2k$ . Compare the formulas:



$$y_i = a + ih = a + 2ik,$$

$$x_j = a + jk.$$

This shows that  $x_0 = y_0 = a$ ,  $x_2 = y_1$ ,  $x_4 = y_2$ ,  $\dots$ ,  $x_{2i} = y_i$ .

In other words, the partition  $\{y_i\}$  is a subset of partition  $\{x_j\}$ . This means that we do not need to evaluate  $f$  again for a new estimate. We can build the two sums for  $I_n$  and  $I_{2n}$  simultaneously by adding to two registers. We simply go through values  $f(x_j)$ . When  $j$  is odd, we add  $f(x_j)$  with a suitable weight to the register for  $I_{2n}$ . When  $j$  is even, we add it to both registers with appropriate weights. Thus in one run we get both approximations  $I_n$  and  $I_{2n}$  and we are ready for Richardson extrapolation.

What is the outcome? Assume that we use the trapezoid rule, then the error has estimate of the form  $Ch^2 + O(h^3)$ . If we apply the Richardson formula with  $p = 2$ , we expect to obtain an estimate with order  $h^3$ .

Now imagine that we do three approximations,  $I_n$ ,  $I_{2n}$  and  $I_{4n}$ . Applying Richardson extrapolation to the first two we obtain  $R_n$  with error of order  $O(h^3)$ , applying it to the last two we obtain an estimate  $R_{2n}$  of the same order. So now we can apply Richardson extrapolation with  $p = 3$  and obtain an approximation for the given integral with error  $O(h^4)$  and so on.

We could build pyramids like this with more and more steps, obtaining approximations of increasing orders. This approach is called the **Romberg integration**.

**Example 5c.a:** Consider the integral  $I = \int_0^2 e^x dx$ . We know the answer, it is  $I = e^2 - 1$ .

We start with the partition corresponding to  $N = 8$ , that is, we look at points

$$x_0 = 0.0, x_1 = 0.25, x_2 = 0.5, x_3 = 0.75, x_4 = 1.0, x_5 = 1.25, x_6 = 1.5, x_7 = 1.75, x_8 = 2.0.$$

The trapezoid rule with  $n = 8$  uses the step  $h = \frac{1}{4}$  and yields the estimate

$$I_8 = \frac{1}{2} \cdot \frac{1}{4} [e^0 + 2e^{0.25} + 2e^{0.5} + 2e^{0.75} + 2e^{1.0} + 2e^{1.25} + 2e^{1.5} + 2e^{1.75} + e^{2.0}].$$

The trapezoid rule with  $n = 4$  uses the step  $h = \frac{1}{2}$  and yields the estimate

$$I_4 = \frac{1}{2} \cdot \frac{1}{2} [e^0 + 2e^{0.5} + 2e^{1.0} + 2e^{1.5} + e^{2.0}].$$

Finally, the trapezoid rule with  $n = 2$  uses the step  $h = 1$  and yields the estimate

$$I_2 = \frac{1}{2} \cdot 1 [e^0 + 2e^{1.0} + e^{2.0}].$$

Note that we indeed did not need any new values of the function for  $I_4$  and  $I_2$ .

We can estimate the error of  $I_4$  and  $I_8$  using the Richardson formula, obtaining estimates

$$I - I_4 \approx \frac{I_4 - I_2}{2^2 - 1} = E_4,$$

$$I - I_8 \approx \frac{I_8 - I_4}{2^2 - 1} = E_8.$$

Let's compare the values.

$I - I_4$	$E_4$	$I - I_8$	$E_8$
-0.1326...	-0.1304...	-0.0332...	-0.0331...

The estimates work really well. Now we use pairs of estimates  $I_{2n}, I_n$  to obtain the first generation of Richardson extrapolations:

$$R_8 = \frac{2^2 I_8 - I_4}{2^2 - 1}, \quad R_4 = \frac{2^2 I_4 - I_2}{2^2 - 1}.$$

These should be estimates of order  $h^3$ , we can therefore use them to estimate the error of  $R_8$ ,

$$I - R_8 \approx \frac{R_8 - R_4}{2^3 - 1} = E.$$

Finally, we can apply Richardson extrapolation to  $R_4, R_8$  with  $p = 3$  to obtain approximation of  $I$  of order  $h^4$ . This would be the result of the Romberg integration applied to  $I_8, I_4$ , and  $I_2$ :

$$R = \frac{2^3 R_8 - R_4}{2^3 - 1}.$$

Let's compare actual errors and the estimate for  $I - R_8$ .

$I - R_4$	$I - R_8$	$E$	$R$
-0.00215...	-0.00014...	-0.00029...	-0.00015...

This does not quite look all that great. The error estimate is of the right order, but it is twice as large as it should be. What is worse, the approximation  $R$  has larger error than  $R_8$ , which is really unexpected.

Things like this can happen if we do not guess the right order of error. What do we really know of the error of  $R_n$ ? We have written  $O(h^3)$ , but that was just an upper estimate. Could it be that the actual order is better?

We try again, this time assuming that the first-generation Richardson extrapolation yields a method of order  $h^4$ . Then

$$I - R_8 \approx E = \frac{R_8 - R_4}{2^4 - 1}, \quad R = \frac{2^4 R_8 - R_4}{2^4 - 1}.$$

Now the results go as follows.

$I - R_4$	$I - R_8$	$E$	$R$
-0.002154...	-0.000138...	-0.000134...	-0.000003...

This is very good. The error estimate almost matches the actual error and the second generation Richardson extrapolation shows great accuracy. Indeed, a slightly different approach than the one we used above shows that the error of the trapezoid rule depends only on even powers of  $h$ . So once Richardson extrapolation cancels the term  $h^2$ , the next largest is  $h^4$ . And when we use the second generation of Richardson extrapolation to get rid of  $h^4$ , the resulting method is of order  $h^6$  and so on.

Just out of curiosity, if  $R_8$  is really a method of order  $h^4$ , then it should have error comparable to that of the Simpson method with the same data. Lets see:

$$S(8) = \frac{1}{3} \cdot \frac{1}{4} [e^0 + 4e^{0.25} + 2e^{0.5} + 4e^{0.75} + 2e^{1.0} + 4e^{1.25} + 2e^{1.5} + 4e^{1.75} + e^{2.0}].$$

The error is  $-0.000138\dots$ , very similar to the error of  $R_8$ , in fact their errors differ by  $1 \cdot 10^{-8}$ , which looks suspiciously like a numerical error rather than an actual difference.

What is really happening here? We will now look at a general case and focus on local situation. Since panels of  $I_n$  are twice as large as those of  $I_{2n}$ , we will start our investigation with two adjacent

panels, that is, we will try to approximate  $\int_{c-h}^{c+h} f(x) dx$ .

When evaluating  $I_n$  with step size  $h = 2k$ , this is seen as one panel, so we use trapezoid with sides  $f(c-h)$  and  $f(c+h)$ , obtaining the approximation

$$P_n = \frac{1}{2}(2k)[f(c-h) + f(c+h)].$$

When evaluating  $I_{2n}$  with step size  $k = \frac{1}{2}h$ , we treat each panel separately and obtain

$$P_{2n} = \frac{1}{2}k[f(c-h) + f(c)] + \frac{1}{2}k[f(c) + f(c+h)] = \frac{1}{2}k[f(c-h) + 2f(c) + f(c+h)].$$

Richardson extrapolation combines these two approximations as follows:

$$\begin{aligned} R_{2n} &= \frac{2^2 P_{2n} - P_n}{2^2 - 1} = \frac{1}{3}(2k[f(c-h) + 2f(c) + f(c+h)] - k[f(c-h) + f(c+h)]) \\ &= \frac{1}{3}k(f(c-h) + 4f(c) + f(c+h)). \end{aligned}$$

This is the panel approximation of the Simpson method. We just proved that the first generation of Richardson extrapolation, when applied to the trapezoid method, yields the Simpson method. What an amazing coincidence! And of course, this confirms that its order is really  $O(h^4)$  and that  $R_8$  should have the same error estimate as  $S(8)$ , they are equal.

△

This example suggests that the Romberg integration is a very promising way of obtaining very good approximations of integrals with moderate number of evaluations of  $f$ . It is traditional to start with the trapezoid rule and work its way up, each generation improving the order of error by two. The first generation yields—as we already saw—the Simpson method. The second generation yields another Newton-Cotes method, this time of order  $h^6$ , but from order  $h^8$  on Richardson extrapolations yield different methods than those of Newton-Cotes (approximation by polynomials), and those methods are more numerically stable. Thus if high accuracy is needed, Romberg integration is preferable to using high order Newton-Cotes methods.

### 5c.1 Bonus:

Here we will show that the trapezoid rule only uses even powers in the error estimate. We use a trick that introduces symmetry into the picture.

We start with a panel  $\int_c^{c+h} f(x) dx$ , but now we introduce an artificial center  $a = c + \frac{1}{2}h$  for all the expansions. Then

$$\begin{aligned} \int_c^{c+h} f(x) dx &= \int_{a-h/2}^a f(x) dx + \int_a^{a+h/2} f(x) dx = F(h/2) - F(-h/2) \\ &= 2f(a)\frac{h}{2} + \frac{2}{3!}f''(a)\left(\frac{h}{2}\right)^3 + \frac{2}{5!}f''''(a)\left(\frac{h}{2}\right)^5 + \dots \end{aligned}$$

We see that even powers disappeared. We want to approximate this using the values  $f(c) = f(a - h/2)$  and  $f(c + h) = f(a + h/2)$ . We expand and combine these expansions. Since we know that we will be using the trapezoid rule, we form the right linear combination right away.

$$\frac{1}{2}h[f(c) + f(c + h)] = 2f(a)\frac{h}{2} + \frac{2}{2!}f''(a)\left(\frac{h}{2}\right)^3 + \frac{2}{4!}f''''(a)\left(\frac{h}{2}\right)^5 + \dots$$

The first term of this expansion matches the first term in the expansion of the integral, so when we subtract, we obtain the local error of order  $h^3$ :

$$E_{\text{loc}} = \left(\frac{2}{3!} - \frac{2}{2!}\right)f''(a)\left(\frac{h}{2}\right)^3 + \left(\frac{2}{5!} - \frac{2}{4!}\right)f''''(a)\left(\frac{h}{2}\right)^5 + \dots$$

Summing up over the panels, we replace the averages of derivatives as above and obtain global error estimate

$$E_{\text{loc}} = \frac{(b-a)}{2^2} \left(\frac{1}{3!} - \frac{1}{2!}\right) f''(\xi_2) h^2 + \frac{(b-a)}{2^5} \left(\frac{1}{5!} - \frac{1}{4!}\right) f''''(\xi_4) h^4 + \dots$$

This confirms that the error estimate for the trapezoid rule really depends only on even powers of  $h$ .

## 6. Formal introduction to ordinary differential equations

In chapter 1 we introduced basic concepts related to ordinary differential equations. In this chapter we give them precise meaning so that we can use them properly. We will also present our first results about existence of solutions.

First we make a formal definition of an ODE. The accepted word definition says that it is “an equation with a function as its unknown that features some derivatives of the said unknown function”. Now we will say it mathematically.

When we say an equation, we envision a formula that features some knowns and some unknowns. In our particular case the unknown should be a function. To make our life easier, we can always rearrange the given equation so that all terms are on one side. For instance, given  $y'(x) + 13x^2y(x) = 23 \sin(y(x)) + e^x$ , we can write it as  $y'(x) + 13x^2y(x) - 23 \sin(y(x)) - e^x = 0$ .

In this equation,  $x$  is not really an unknown, because we are not trying to solve for it. Rather, it is a working variable for the unknown function  $y$ , and it is a dummy variable. It is traditional to skip it when writing the function  $y(x)$ , so in fact the above equation would normally be written like this:

$$y' + 13x^2y - 23 \sin(x) - e^x = 0.$$

We mostly work in this form, but sometimes it is good to remind oneself that in fact  $y = y(x)$ ,  $y' = y'(x)$  etc.

Anyway, on the left there is an expression that combines three objects: The dummy variable  $x$ , a function  $y$  and its derivative  $y'(x)$ . We can call these them by simple letters and introduce a formula that would capture how these three objects are treated:

$$f(r, s, t) = t + 13r^2s - 23 \sin(s) - e^r.$$

Now we can recover the (rearranged) given equation by writing

$$f(x, y, y') = 0.$$

See for yourself that when  $x$  is substituted for  $r$ ,  $y$  for  $s$  and  $y'$  for  $t$  in the function  $f$ , then we indeed get that equation.

In this way we can capture all differential equations in one pattern. Actually, this abstract view is not exactly useful for someone who is just trying to learn how to work with ordinary differential equations, but it does allow us to build a general theory by giving us a convenient handle on ODEs, so it worthwhile to see how it works.

For instance, it allows us to say what a solution is. How do we recognize a solution of, say, an algebraic equation with an unknown  $x$ ? It is some number, and if we substitute it into the given equation, its two sides become the same number. Now a differential equation has a function as its unknown. When we substitute a function into a given ODE, we obtain a formula on the left and a formula on the right, both featuring  $x$ . For instance, by substituting  $y(x) = \ln(x)$  into the original equation above, we obtain

$$\frac{1}{x} + 13x^2 \ln(x) = 23 \sin(\ln(x)) + e^x.$$

In order for  $y(x) = \ln(x)$  to be a solution, we would want the two expressions on the left and on the right to be the same. However, when comparing functions, we always have to say where we do it, that is, for what values of the dummy variable  $x$ . It may happen that two expressions agree for some  $x$  and disagree for others, for instance  $x = |x|$  is only true for  $x \geq 0$ . Thus when talking of solutions to a given ODE, we always have to specify formula and also a set on which we obtain equality in the equation. As we will explain below, it makes sense to work with intervals, usually open ones. We are ready.

**Definition 6.1.**

By an **ordinary differential equation** (ODE) we mean any equation of the form

$$f(x, y, y', \dots, y^{(n-1)}, y^{(n)}) = 0,$$

where  $f$  is a function of  $n + 1$  variables.

By its **solution on an (open) interval**  $I$  we mean any function  $y = y(x)$  on interval  $I$  that has all derivatives up to order  $n$  on  $I$  and for all  $x \in I$  satisfies

$$f(x, y(x), y'(x), \dots, y^{(n-1)}(x), y^{(n)}(x)) = 0.$$

**Example 6.a:** The equation

$$y'' - y \ln(y) - \frac{1}{x}y' = \frac{(y')^2}{y} + 3x e^x y$$

is an ordinary differential equation, we see that it features an unknown function  $y = y(x)$ . To make it fit formally, we observe that it has the form  $f(x, y, y', y'') = 0$  for the choice

$$f(r, s, t, u) = u - s \ln(s) - \frac{1}{r}t - \frac{t^2}{s}3r e^r s.$$

We will show that  $y(x) = e^{x^2 e^x}$  is a solution of our equation on any open interval not containing zero. First little preparation:

$$y'(x) = e^{x^2 e^x} [x^2 e^x]' = e^{x^2 e^x} (2x e^x + x^2 e^x).$$

We are ready. We take any  $x \in \mathbb{R} - \{0\}$  and observe that then  $e^{x^2 e^x} > 0$ , so we can substitute it without any trouble into both sides of our equation, we can also divide by  $x$ . We start with the left hand side.

$$\begin{aligned} LHS &= y'' - y \ln(y) - \frac{1}{x}y' \\ &= [e^{x^2 e^x}]'(2x e^x + x^2 e^x) + e^{x^2 e^x} [2x e^x + x^2 e^x]' - e^{x^2 e^x} x^2 e^x - \frac{1}{x}e^{x^2 e^x} (2x e^x + x^2 e^x) \\ &= e^{x^2 e^x} (2x e^x + x^2 e^x)^2 + e^{x^2 e^x} (2e^x + 4x e^x + x^2 e^x) - e^{x^2 e^x} x^2 e^x - e^x e^{x^2 e^x} (2e^x + x e^x) \\ &= e^{x^2 e^x} (2x e^x + x^2 e^x)^2 + 3x e^x e^{x^2 e^x}. \end{aligned}$$

Now the right hand side.

$$\begin{aligned} RHS &= \frac{(y')^2}{y} + 3x e^x y = \frac{(e^{x^2 e^x})^2 (2x e^x + x^2 e^x)^2}{e^{x^2 e^x}} + 3x e^x e^{x^2 e^x} \\ &= e^{x^2 e^x} (2x e^x + x^2 e^x)^2 + 3x e^x e^{x^2 e^x}. \end{aligned}$$

Both sides agree.

△

The general form  $f = 0$  of an ODE is convenient for theory, but not easy to handle. Fortunately for us, applications typically lead to equations where the highest derivative can be isolated. For instance, the equation in example 6.a can become explicit if we write it as

$$y'' = \frac{(y')^2}{y} + \frac{1}{x}y' + y \ln(y) + 3x e^x y$$

We can capture it in a general form as  $y'' = f(r, s, t)$  using  $f(r, s, t) = \frac{t^2}{s} + \frac{1}{r}t - s \ln(s) + 3r e^r s$ .

**Definition 6.2.**

By an **explicit ordinary differential equation of order  $n$**  we mean any equation of the form

$$y^{(n)} = f(x, y, y', \dots, y^{(n-1)}),$$

where  $f$  is a function of  $n$  variables.

Naturally, a function  $y(x)$  is a solution on interval  $I$  if

$$y^{(n)}(x) = f(x, y(x), y'(x), \dots, y^{(n-1)}(x)) \text{ on } I.$$

We see that our example is an explicit ODE of order 2. Generally, explicit ODE's are easier to solve than non-explicit ones. Thus (almost) all authors immediately pass to the explicit type and never look back. This book is no different, so from now on when we say "an ODE", we definitely mean the explicit form of the type  $y^{(n)} = f$ .

However, even explicit equations are still very tough and there are no general approaches to solving such ODEs. For instance, I have no idea how to solve the equation in example 6.a and I am willing to bet that nobody really does, although it is just a second order equation that features common and popular functions. So the second thing all authors do is to focus on specific types of ODEs. In this book we will look at four traditional types.

By the way, if the equation above cannot be solved, how did I get the solution then? I cheated, I started with  $y(x) = e^{x^2 e^x}$  and then I created an equation for it.

We will do theory here assuming that the names of the dummy variable and the function are  $x$  and  $y$ , but it should be noted that other names are possible, especially when equations come from applications. One popular choice is to use  $t$  for time, and when the unknown function describes position, people may like to call it  $x$ . In some fields it is also traditional to use dots to mark derivatives with respect to time, so the equation from example 6.a might have looked like this:

$$\ddot{x} = \frac{(\dot{x})^2}{x} + \frac{1}{t}\dot{x} - x \ln(x) + 3t e^t x.$$

It should be easy to adjust, especially if you really understand what you are doing.

Before we move on, note that there are good reasons to consider solutions on open intervals. First, we know that in order to differentiate a function at some point, the function has to exist on some neighborhood of it. In other words, if we want a function on some set  $M$  so that it has a derivative everywhere there, then this set must be open.

Why an interval? In a typical application, the unknown function represents the state of some system, that is, it is some quantity that changes according to some natural laws or perhaps some other rules. When we start an experiment and then interrupt it, let the system do what it wants and after a while start watching it again, it does not make much sense to see it as one event, because during the break we lost control of the situation. Properly it should be handled as two experiments. We get meaningful results only when the time of the experiment is uninterrupted, that is, when the independent variable is allowed to change without interruption. Thus we arrive at an interval.

For instance, in example 6.a we could say that we have a solution  $y(x) = e^{x^2 e^x}$  on  $(1, 13)$ . We could also say that we have a solution  $y(x) = e^{x^2 e^x}$  on  $(23, 31)$ . Formally, these are two different solutions, because they describe two different events, each taking place in its own time.

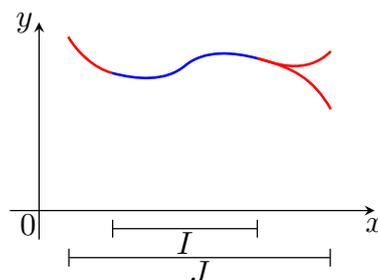
Of course, in both cases we do not fulfill the full potential of our work, it is much better to say that we have a solution  $y(x) = e^{x^2 e^x}$  on  $(0, \infty)$  and it also includes information about the two solutions above. It is also quite clear that we cannot provide an even more encompassing solution due to the condition  $x \neq 0$ : there is no interval  $J$  containing  $(0, \infty)$  as a proper subset on which our formula would be a solution. So in some sense we cannot improve on this solution.

Naturally, when solving an ODE, we always try to find a solution on the largest possible interval, because in this way we provide the best information that is available. We can always pass to a subinterval if we need to. It should be noted that our example was in a sense misleading, because we have a formula for the solution and we simply played with intervals, which looks like there is no depth to it. However, there are situations when things get more interesting, in particular when we do not have solutions given by handy formulas. There are notions that allow us to talk precisely about the ideas that we were now exploring.

**Definition 6.3.**  
 Let  $y, v$  be some solutions of a given ODE on open intervals  $I, J$  respectively. We say that  $v$  is an **extension** of  $y$  if  $I \subset J, I \neq J$  and  $v = y$  on  $I$ . We say that a solution  $y$  of a given ODE on some open interval is a **maximal solution** if it cannot be extended, that is, if there is no other solution that would be an extension of  $y$ .

Going back to example 6.a, we could say that the solution  $y(x) = e^{x^2 e^x}, x \in (1, 23)$  is an extension of the solution  $y(x) = e^{x^2 e^x}, x \in (3, 13)$ . However, none of them is maximal. We can find two maximal solutions based on our formula, namely  $y(x) = e^{x^2 e^x}, x \in (0, \infty)$  and  $y(x) = e^{x^2 e^x}, x \in (-\infty, 0)$ . There may also be other maximal solutions as well, but we do not know enough about our equation to tell.

This general definition allows for some interesting things. For instance, it may happen that a solution has several extensions that differ not just in interval but also in values, as suggested in the picture. We will see such a situation in example 6a.a below. In fact, as far as we know right now, even the solution  $y(x) = e^{x^2 e^x}, x \in (3, 13)$  discussed above could perhaps have some extension that is not based on the formula  $e^{x^2 e^x}$ .



Common sense suggests that when we have a solution on an interval, then either it already is maximal, or it can be extended to a maximal solution. This is in fact true. However, this maximal extension need not be determined uniquely. For instance the two distinct extensions in the picture above give rise to two different maximal solutions, they may even exist on different intervals. Thus proving a statement on maximal extension calls for some advanced mathematics, namely Zorn’s lemma.

It should be noted that when it comes to differential equations coming from applications, then usually the extensions happen in a unique way. We may imagine that the blue common part represents some process, say, a stone falling. It would be really unusual if this falling stone was allowed to decide at some point between two different paths to take, although outer conditions would remain the same. Thus for “practical” equations we expect unique maximal extensions, and this will be true for most ODEs we encounter in this book.

We will return to this situation below.

We have a much better idea of what to expect from solutions of a given ODE if we can capture them all in one formula. We already encountered such situations in chapter 1. This is very desirable.

**Convention 6.4.**  
 If the set of all solutions of a given ODE on some interval  $I$  can be expressed using one formula with parameters, we say that this formula is a **general solution** of this ODE. An individual solution of this equation is called a **particular solution**.

**Example 6.b:** Consider the ODE  $y' = -y$ . We easily check that  $y(x) = C e^{-x}$  is a solution on  $\mathbb{R}$  for any choice of  $C \in \mathbb{R}$ . As we will see later, there are no other solutions, so the formula  $y(x) = C e^{-x}$ ,  $x \in \mathbb{R}$  qualifies as a general solution.

By the way, by taking  $C = 0$  we obtain the constant particular solution  $y(x) = 0$ , we will see later that it is of some interest. It is one of infinitely many possible particular solutions that we can get by choosing a value for  $C$ .

And by the way, note that the dummy variable is not present in the given equation. This sometimes happens. How did we know to use  $x$ ? Actually, we did not know as such, we knew that that name of the variable is not crucial and thus we were free to choose. We could have worked with  $y(t) = C e^{-t}$  and everything would be just fine. Often the name of the variable does not come from the equation itself but from the application that our equations is coming from.

△

Note that we did not introduce the notion of general solution as a definition. The reason is that people sometimes use the name general solution also in situations that almost but not quite fit with our convention.

**Example 6.c:** Consider the ODE  $y' = -y^2$ . We easily check that  $y(x) = \frac{1}{x-C}$  is a solution for any  $C \in \mathbb{R}$ , and this solution is then valid on any interval not containing  $C$ . Since we want the largest possible intervals, we could say that  $y(x) = \frac{1}{x-C}$ ,  $x \in (-\infty, C)$  and  $y(x) = \frac{1}{x-C}$ ,  $x \in (C, \infty)$  are solutions.

However, there is yet another solution, namely the constant function  $y(x) = 0$  (on  $\mathbb{R}$ ), and it cannot be obtained using the formula  $\frac{1}{x-C}$  for any choice of  $C$ .

So is  $\frac{1}{x-C}$  a general solution?

△

Technically the answer should be in the negative, but people often would call this formula a general solution because it captures the substance of behaviour for the problem that we study. Some people would say that the general solutions is as follows:

$$y(x) = \frac{1}{x-C}, x \neq C \text{ or } y(x) = 0, x \in \mathbb{R}.$$

I do not want to take sides here, just warn the reader that there is some ambiguity. The good news is that for most common (and important) types of differential equations we do not have exceptional solutions, so once we get the right formula, it captures all solutions.

Now what should a student do when a question (or algorithm) calls for a general solution? A good general advice is that when in doubt, it never hurts to supply more information. So write the formula and also list all exceptional solutions (if there are any).

For well-behaved differential equations, the number of parameters in a general solution is equal to the order of the equation. For instance, if we were able to find a formula for a general solution in example 6.a, it would have two parameters. We obtain a particular solution by choosing specific values for these parameters.

This is usually not done directly, but by posing additional requirement on our solution. Then we find values for the parameters so that our solution satisfies them. Common sense suggests that when we have a certain number of parameters, we determine them uniquely by we posing as many requirements (assuming that they are independent, it does not help to ask for the same thing several times). Typically, we choose some time or times and we specify values that our solution should have at these times, we can also ask for specific values for derivatives of this function. For some examples we refer to example 1.b. As we noted there, the most popular way to choose one solution out of infinitely many is to specify how the whole event should start.

**Definition 6.5.**

Consider an explicit ODE of order  $n$   $y^{(n)} = f(x, y, y', \dots, y^{(n-1)})$ .

By an **Initial Value Problem (IVP)** or a **Cauchy problem** for this equation we mean any problem of the form

(1) ODE:  $y^{(n)} = f(x, y, y', \dots, y^{(n-1)})$ ;

(2) **initial conditions:**  $y(x_0) = y_0, y'(x_0) = y_1, \dots, y^{(n-1)}(x_0) = y_{n-1}$ ,

here  $x_0, y_0, y_1, \dots, y_{n-1}$  are some fixed real numbers.

Note that these initial conditions represent exactly  $n$  equations, so if everything goes well, this should be just enough to determine the  $n$  parameters in a general solution. There are two reasons which this type of conditions is popular. First, it makes a good sense in applications, and second, one can build a nice theory about it, which we will do later.

Generally speaking, in this book we will focus on two kinds of problems. We may ask for a general solution to a given ODE (in the sense that we want to see all solutions), or we ask for a specific (particular) solution given by some Initial value problem. As we will see, the latter question requires that we solve the former first.

## 6a. Existence and uniqueness

There are two basic questions related to initial value problems (and to particular solutions in general). The first one is natural: Is there a solution to the given problem? The second question is also important: Is there only one solution to the given problem, or is it possible that there are more of them?

We actually touched on this problem earlier. Look at the picture above where the blue solution has two possible extensions. The point where the two red curves branch has certain coordinates  $x_0, y_0$ . When we ask for a solution that passes through this point, we actually ask for a solution  $y$  given by the initial condition  $y(x_0) = y_0$ . The picture thus captures a typical case when an initial condition problem has two distinct solutions.

As we commented then, such a situation is highly unusual in applications (which does not mean impossible, but rare and very special). Our new point of view allows for a slightly different story to explain this: When we have two identical situations (initial conditions) and the same set of natural laws (the ODE), we expect the situation to develop in a unique way.

There is one more question that is asked, although perhaps not as important as existence and uniqueness: Can we always find a solution that makes sense on the same set (for the independent variable) where the equation itself makes sense? We would call this a global solution. In general this need not be true, sometimes we are only able to find a solution on a smaller set (local solution), which again indicates that something interesting is happening.

Every time we focus on some special type of differential equations, the question of existence and uniqueness is among the first we ask. We will now focus on one of the simplest cases: Explicit ordinary differential equations of the first order. That is, we consider differential equations of the form

$$y' = f(x, y).$$

Since they are of order 1, we choose particular solutions with just one initial condition of the form  $y(x_0) = y_0$ .

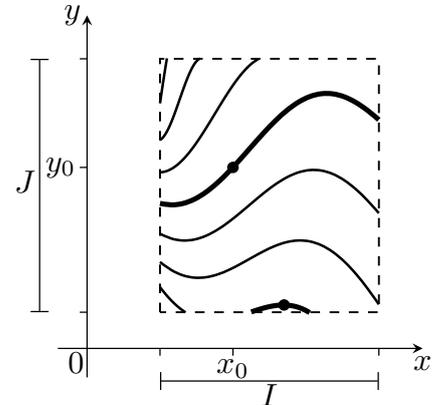
The mathematical playground for such equations is the usual two-dimensional  $xy$ -plane and it actually plays two roles here. The first role is that of the domain for the function  $f$ , because this function sees  $x$  and  $y$  as two independent variables and accepts points from  $\mathbb{R}^2$  (perhaps not all of them, depending on  $f$ ).

The second role of the same plane is that of the place where graphs of solutions appear. If some function  $y = y(x)$  is a solution on some interval  $I$ , then for every  $x \in I$  we obtain a point  $(x, y(x))$

in the plane. Since this point is substituted into the right hand side, it must be in the domain of  $f$ .

The domain of  $f$  may be rather complicated, which makes it difficult to write general theorems. Thus people traditionally focus on “nice” parts of it, and the nicest shape of all is a rectangle. Formally, we have an interval  $I$  for  $x$  and an interval  $J$  for  $y$  (now we treat  $y$  as an independent variable) and we assume that the function  $f$  is defined on the set  $I \times J$ .

If a function  $y = y(x)$  wants to be a solution, in the first place it must be able to enter into  $f$  in the equation, that is, the points  $(x, y(x))$  of its graph must be in  $I \times J$ . The picture on the right shows a typical situation. There are most likely infinitely many solutions in that rectangle, all of a similar kind. When we want to single out one of them, we may try to do it using an initial condition  $y(x_0) = y_0$ . Obviously, the starting time  $x_0$  must be in  $I$  and the value  $y_0$  in  $J$ , that is, point  $(x_0, y_0)$  must be in the rectangle. Note that for the initial point marked  $(x_0, y_0)$  in the picture we obtained a solution defined on the whole  $I$ , so it would be a global solution. There is also another unmarked dot near the lower border. If we choose it as our initial condition, then the corresponding solution leaves the rectangle prematurely, so we end up just with a local solution.



We saw that points of  $I \times J$  also play two roles here: As ordered couples  $(x_0, y_0)$  they denote points in the plane, at the same time they code the initial condition  $y(x_0) = y_0$ .

Now it should be easier to read the following theorem.

**Theorem 6a.1.** (Peano’s thm on existence)

Consider an ODE of the form  $y' = f(x, y)$ . (\*)

Let  $I, J$  be open intervals such that  $f$  is continuous on the set  $I \times J$ .

Then for all  $(x_0, y_0) \in I \times J$  there exists a solution of the IVP (\*),  $y(x_0) = y_0$  on some neighborhood of  $x_0$ .

Note that there are things that this theorem does not claim. First, only a local existence of a solution is guaranteed, and there is no specification of the size of the neighborhood where the solution is supposed to exist. It can be really tiny. Examples show that we cannot really promise more.

Second, it does not say anything about uniqueness, and for a good reason.

**Example 6a.a:**

Consider the initial value problem  $y' = 3y^{2/3}$ ,  $y(0) = 0$ .

The function  $f(x, y) = 3y^{2/3}$  is continuous on the rectangle  $\mathbb{R} \times \mathbb{R}$ , so for any point  $(x_0, y_0)$  in the plane there is some solution of the equation  $y' = 3y^{2/3}$  passing through it. How about the point  $(0, 0)$  corresponding to the given initial condition above?

We can easily see that  $y(x) = 0$ ,  $x \in \mathbb{R}$  is a solution to this problem. But also the function  $y(x) = x^3$ ,  $x \in \mathbb{R}$  solves it, and the function  $y(x) = -x^3$ ,  $x \in \mathbb{R}$  is the third solution. So there are three distinct solutions passing through  $(0, 0)$ .

Note also that all three solutions above have  $y'(0) = 0$ . This means that we can actually “glue” them at the origin. Consider the following functions:

$$u(x) = \begin{cases} 0, & x < 0; \\ x^3, & x \geq 0, \end{cases} \quad v(x) = \begin{cases} -x^3, & x < 0; \\ x^3, & x \geq 0. \end{cases}$$

Since both agree with  $y(x) = x^3$  on  $(0, \infty)$ , they must be solutions of our ODE there. Each of them also agrees with one of our solutions on  $(-\infty, 0)$ . Finally, we easily check that both functions

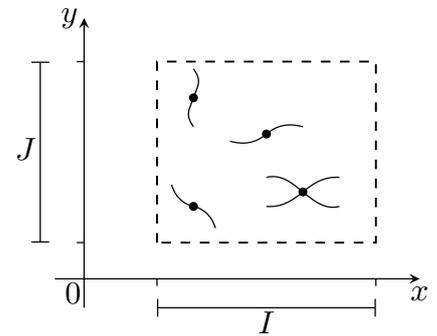
are continuous at 0, moreover,  $u'_+(0) = u'_-(0) = 0$  and  $v'_+(0) = v'_-(0) = 0$ , which means that both functions are differentiable at  $x = 0$  and  $u'(0) = v'(0) = 0$ . Then these two functions also satisfy the given ODE for  $x = 0$ . We conclude that both  $u$  and  $v$  are solutions of the given ODE on  $\mathbb{R}$ .

So this is what gluing solutions means. When two solutions meet at a certain point and they have the same derivative there, we can combine them as we wish and obtain a solution again. From a practical point of view, imagine that you go along one of the two solutions. When you come to that meeting place, you can decide which way to go and all options are valid. Obviously, this is rather unusual and we do not expect this to happen in nature. In other words, the ODE in this example is almost surely artificial, invented by mathematicians to cause troubles.

One last remark: Consider the solution  $y(x) = x^3$ ,  $x \in (0, 13)$ . Then the three functions  $y$ ,  $u$  and  $v$  are distinct extensions of this solution to  $\mathbb{R}$ . We talked about such a possibility when discussing extensions, now we see that it can really happen.

△

This shows that the picture above is not the best representation of the meaning of Peano's theorem. For that we need a different picture, for instance the one on the right. Look closer at the point through which two different solutions pass. It seems that they do not have the same derivative at that meeting point, so we cannot glue the left part of one with the right part of the other and hope to obtain a solution. Indeed, such a function would have a sharp bend at the meeting point, that is, it would not have a derivative there.



We usually hope that our situation is better than this, in other words, we expect our equations to have more “nice” properties than just the continuity in Peano's theorem.

One particular way in which an equation can be “nice” is that the formula we see on the right does not react violently to changes in reality. Imagine an explicit equation  $y^{(n)} = f$ . We can imagine that the function  $y(x)$  represents the state of some system at time  $x$ , and this state is influenced by its environment. The right hand side  $f$  often represents this influence, and because it can feature the state  $y$  (and perhaps its derivatives), it could be seen as a feedback equation. For instance, if  $y$  represents the temperature of some (small) object, then obviously the temperature at a certain time will have very strong influence on its temperature later on.

The effect of environment wants to change the state  $y(x)$  in some way, and the left hand side describes how. For instance, in example 1.b, the effect of environment (gravity) wants to accelerate the object, so we see the second derivative of position on the left. Coming back to our setting, in the equation  $y' = f(x, y(x))$  the environment forces  $y$  to change at a specific rate, and again it could be a feedback type of situation.

We usually expect that this feedback is not violent, that is, small changes in  $y$  do not provoke big changes in  $f$ . This is actually a property that is studied in analysis of real functions and there is a natural way to express it. Consider a function  $g(t)$ . The fact that the change in  $g$  should not be out of proportion compared to change in variable  $t$  can be captured by the inequality

$$|g(t_2) - g(t_1)| \leq K|t_2 - t_1|,$$

When a function satisfies this for all  $t_1, t_2$  from some set  $M$ , we say that it is a **Lipschitz function** on  $M$ . We may also say that it is  $K$ -Lipschitz on  $M$  when we need to work with this  $K$ .

We can rearrange the inequality

$$\left| \frac{g(t_2) - g(t_1)}{t_2 - t_1} \right| \leq K.$$

On the left we have the slope of a secant line connecting two points on the graph of  $g$ , and we require these slopes to have a common upper bound, which explains this property geometrically: The graph of  $g$  should not be too steep at any part.

Going back to our situation, we want the function  $f(x, y)$  to be so nice when we change the second variable while keeping  $x$  constant. But that is not all. When we look at the differences  $|f(x, y_2) - f(x, y_1)|$ , we would like to get an estimate of the form  $K \cdot |y_2 - y_1|$  with  $K$  that does not depend on  $x$ , so this  $K$  should be universal.

We are ready for another theorem.

**Theorem 6a.2.** (Picard's thm on existence and uniqueness)

Consider an ODE of the form  $y' = f(x, y)$ . (\*)

Let  $I, J$  be open intervals such that  $f$  is continuous on the set  $I \times J$  and there exists  $K > 0$  such that for all  $x \in I$ ,  $f$  is  $K$ -Lipschitz as a function of  $y$  on  $J$ .

Then for all  $(x_0, y_0) \in I \times J$  there exists a solution of the IVP (\*),  $y(x_0) = y_0$  on some neighborhood of  $x_0$  and this solution is unique on this neighborhood.

This solution can be extended to the boundary of  $I \times J$ .

The main improvement of this theorem is the uniqueness result. Note that both existence and uniqueness is local in the statement, but in fact one can pass to a global conclusion. Consider a certain solution that passes through a given point  $(x_0, y_0)$ . Direct application of this theorem tells us that this solution may fork, but it could not happen too close to  $x_0$  (local uniqueness). However, we can start looking at other points on this solution (those that are in  $I \times J$ ) and apply this theorem for them. Then we realize that our solution cannot really fork inside  $I \times J$ , so by applying our local theorem repeatedly we realize that we in fact have global uniqueness (on  $I \times J$ ).

Can we apply similar reasoning to existence? Consider some solution. Can it "end" inside  $I \times J$ ? The local existence result tells us that this solution cannot end at some particular point  $(x_0, y_0) \in I \times J$ , because it must exist on some neighborhood of  $x_0$ . This leaves the possibility that a solution may end up in an "open" way, that is, it may exist on some open interval strictly smaller than  $I$ . Such a situation does not contradict local existence. Indeed, when we take some point on an "open-ended" solution, then the solution will exist on some neighborhood. Of course, when we look at points near the "ends", those neighborhoods get really small.

Which brings us to another improvement over the Peano theorem. The proof of the Picard theorem (see the bonus chapter 39) shows that we can in fact guarantee a certain size of those neighborhoods of existence, which allows for extending solutions all the way to the boundary of  $I \times J$ . Unfortunately, we do not know to which boundary, so a solution need not exist on the whole interval  $I$  (see example 6a.b below). This limits the usefulness of this global result and many people state this theorem without the last sentence.

There is also another popular version of this theorem. When we look at assumptions in Picard's theorem, the one about Lipschitz property seems more involved and less easy to check than the others. However, there is a relatively convenient way to see it. Recall that for a function  $g(t)$ , being Lipschitz on some set means that we have to find a global upper bound for the ratio  $|\frac{g(t_2) - g(t_1)}{t_2 - t_1}|$  on this set. This fraction should be familiar to all readers who passed introductory calculus course. Indeed, if  $g$  is suitably nice on the interval between  $t_1$  and  $t_2$ , then

$$\frac{g(t_2) - g(t_1)}{t_2 - t_1} = g'(\xi)$$

for some  $\xi$  between  $t_1$  and  $t_2$ . Thus, to guarantee the Lipschitz property on some interval it is enough to show that the function in question has a bounded derivative there. This is usually easier to check. In the Picard theorem we have a function of two variables and we expect it to be Lipschitz with respect to the second variable, which naturally call for an appropriate partial derivative. We thus arrive at the other popular version of Picard's theorem.

**Theorem 6a.3.**

Consider an ODE of the form  $y' = f(x, y)$ . (\*)

Let  $I, J$  be open intervals such that  $f$  is continuous on the set  $I \times J$  and the partial derivative  $\frac{\partial f}{\partial y}$  exists and is bounded there.

Then for all  $(x_0, y_0) \in I \times J$  there exists a solution of the IVP (\*),  $y(x_0) = y_0$  on some neighborhood of  $x_0$  and this solution is unique on this neighborhood.

Lipschitz functions are fairly common. Some functions are Lipschitz on their whole domains, for instance  $f(x) = \sin(x)$  is 1-Lipschitz on  $\mathbb{R}$ , because its derivative is never larger than 1.

This is not true about the function  $f(x) = x^2$ . To see it, just fix  $u = 0$  and observe that  $\frac{f(u)-f(v)}{u-v} = \frac{-v^2}{-v} = v$ , this can get arbitrarily large on  $\mathbb{R}$ . On the other hand, if we choose some finite interval  $I$ , then we can find a constant  $K$  such that  $|f'| \leq K$  on  $I$ . Indeed,  $f'(u) = 2u$ , so if we take  $u$  from an interval  $I = [a, b]$ , then  $|f'(u)|$  never gets larger than  $2 \max(|a|, |b|)$ .

In general one can show that if  $f$  is a function with a continuous derivative on a closed interval  $M$  that is bounded, then  $f$  must be Lipschitz on  $M$ , therefore also on the interior of  $M$ . Thus we can find many examples of Lipschitz functions, popular functions like  $e^x$ ,  $x^n$ ,  $\arctan(x)$  etc. are all Lipschitz when considered on some finite interval  $[a, b]$  or  $(a, b)$ . Also the function  $\ln(x)$  is Lipschitz on intervals  $[a, b]$  or  $(a, b)$  as long as  $0 < a < b < \infty$ .

When people use the Picard theorem to investigate a certain ODE, they often play an interesting game of local and global points of view.

**Example 6a.b:** Consider the differential equation  $y' = 2xy^2$ . What can we say about existence and uniqueness of its solutions?

Since  $f(x, y) = 2xy^2$  is continuous on  $\mathbb{R} \times \mathbb{R}$ , for any initial condition  $y(x_0) = y_0$  there must be a corresponding solution by the Peano theorem. In other words, solutions pass through all points  $(x_0, y_0)$  in the plane. Could it happen that there is a point through which two (or more) different solutions pass? For that we need the Picard theorem, we will use the more convenient form 6a.3.

1. Local step.

Since  $\frac{\partial f}{\partial y} = 4xy$  is not bounded on  $\mathbb{R} \times \mathbb{R}$ , we cannot apply Picard's theorem to the whole plane. Where is the problem? Boundedness fails when  $f \rightarrow \infty$ , which happens when  $x \rightarrow \pm\infty$  and/or  $y \rightarrow \pm\infty$ . To prevent this we have to restrict the choice of  $x$  and  $y$  (we "cut away infinity"). To this end we choose some  $M, N > 0$  and consider the set  $\mathcal{M} = (-M, M) \times (-N, N)$ . Then for any  $(x, y) \in \mathcal{M}$  we have

$$\left| \frac{\partial f}{\partial y} \right| = 4 \cdot |x| \cdot |y| \leq 4MN,$$

so  $\frac{\partial f}{\partial y}$  is bounded on the set  $\mathcal{M}$ . Consequently, the Picard theorem applies to this set.

We conclude that for any  $M, N > 0$  and any  $(x_0, y_0) \in (-M, M) \times (-N, N)$  there exists some local solution passing through this point and it is (locally) unique.

2. Global step.

When we let  $M, N \rightarrow \infty$ , the rectangles from the local step get larger and larger, eventually swallowing the whole plain  $\mathbb{R}^2$ . Since existence and uniqueness are local, they should therefore be true for the whole plain.

Formally we argue as follows. Consider some point  $(x_0, y_0) \in \mathbb{R} \times \mathbb{R}$ . Then there must be  $M, N > 0$  such that  $x_0 \in (-M, M)$  and  $y_0 \in (-N, N)$ . Consequently  $(x_0, y_0)$  lies in the rectangle  $(-M, M) \times (-N, N)$  and by part 1., there must be some local solution passing through this point and it must also be locally unique.

We have shown local uniqueness at all points, therefore no solution can fork anywhere and we can simply say that solutions are unique.

The end result of our investigation then is that through every point in the plane  $\mathbb{R}^2$  passes some solution and it is unique.

### 3. Extending solutions.

Choose some solution that passes through some  $(x_0, y_0)$ . We obtain it by considering some rectangle  $(-M, M) \times (-N, N)$ . The last sentence in Picard's theorem says that this solutions can be extended all the way to the boundary of this rectangle. However, we can now consider a larger rectangle, and see that this solutions can extend also to the boundary of this larger rectangle. We keep going like that and reach the conclusion that this solution can extend indefinitely. But what does it mean? A nice conclusion would be that such a solution is defined on the whole real line.

Unfortunately, there is no guarantee that this is the case. It may also happen that the solution extends "upward" or "downward", going to infinity while  $x$  stays local. Thus we actually do not really know what to expect from such extended solutions, which is the reason why people do not work with this part of the Picard theorem much.

In fact, we will later learn to solve our equation and find out that its general solution is given by the formula  $y(x) = \frac{1}{C-x^2}$ ,  $x^2 \neq C$  (and also the exceptional solution  $y(x) = 0$ ,  $x \in \mathbb{R}$ ).

If we choose some initial condition, say  $y(1) = \frac{1}{3}$ , then we actually ask for  $\frac{1}{C-1^2} = \frac{1}{3}$ , which is true when  $C = 4$ . We found the solution  $f(x) = \frac{1}{4-x^2}$  that passes through our point  $(1, \frac{1}{3})$ , it is unique and it extends to the boundary of  $\mathbb{R}^2$ . In which way? The region of validity must be an interval, we are not allowed to use  $x = \pm 2$  (so three intervals possible), and we want it to contain the "initial time"  $x_0 = 1$ . We conclude that we have a solution  $f(x) = \frac{1}{4-x^2}$ ,  $x \in (-2, 2)$ .

This solution diverges to infinity at endpoints of its region of validity, so it indeed extends all the way to the border of the region  $\mathbb{R}^2$ , but it exists only on a small interval.

Compare also with example 7.b.

△

In example 6a.a we saw a differential equation that did not have unique solutions. What does Mr. Picard think about it?

**Example 6a.c:** Consider the equation  $y' = 3y^{2/3}$ .

The function  $f(x, y) = y^{2/3}$  is continuous on  $\mathbb{R} \times \mathbb{R}$ , so through every point in  $\mathbb{R}^2$  there passes some solution by Peano's theorem.

The derivative  $\frac{\partial f}{\partial y} = \frac{2}{3y^{1/3}}$  is not bounded on  $\mathbb{R}^2$ , we see a problem when  $y \rightarrow 0$ . We need to cut off small values of  $y$ , more precisely, we need to restrict our attention to  $y$  that satisfy  $|y| \geq a$  for some  $a > 0$ .

We thus choose an arbitrary  $a > 0$  and consider  $(x, y)$  from one of the rectangles  $\mathbb{R} \times (a, \infty)$  or  $\mathbb{R} \times (-\infty, -a)$ . Then  $|y| > a$ , therefore  $\frac{1}{y^{1/3}} < a^{-1/3}$ . We conclude that  $\frac{\partial f}{\partial y}$  is bounded on these two rectangles, therefore for all points in them we have local existence and uniqueness of solution.

When we let  $a \rightarrow 0^+$ , these rectangles include more and more points, eventually they swallow up all points  $(x, y)$  in the plane with exception of those that have  $y = 0$ .

Conclusion: Through every point from  $\mathbb{R} \times (0, \infty)$  and from  $\mathbb{R} \times (-\infty, 0)$  passes a solution that is locally unique.

We see that the Picard theorem could not give any guarantee for points of the form  $(x, 0)$ , and example 6a.a shows that there is a good reason for this.

△

These two examples have shown the basic strategy for investigating differential equations using Picard's theorem. We find the partial derivative  $\frac{\partial f}{\partial y}$  and depending on the formula, we typically find ourselves forced to prohibit one or the other variable from going to infinity or to a certain value. This leads us to consider certain rectangles and obtain results for them.

In the second step we then enlarge these rectangles to cover the largest set possible, obtaining the best possible result on existence and uniqueness.

## 7. Separable differential equations

In this chapter we look at probably the most popular type of a differential equation.

### Definition 7.1.

By a **separable ordinary differential equation** we mean any ODE that can be expressed in the form  $y' = g(x)h(y)$  for some functions  $g, h$ .

For instance, the differential equation  $3xyy' = \frac{1}{xy}$  is separable, because we can write it as  $y' = \frac{1}{3y^2} \frac{1}{x^2}$ . On the other hand, the equation  $y' = x + y$  is not separable, because we cannot change it into an appropriate form.

The popularity of this type is understandable: There is a straightforward way to solve such equations. In many cases, a separable ODE can be solved on a few lines. We will start with one such nice ODE. We will use this opportunity to show how practical people approach separable ODEs. It is not quite mathematically correct but non-mathematicians appreciate that it guides them well in the process.

**Example 7.a:** Consider the equation  $3xyy' = \frac{1}{xy}$ . It is a good idea to start solving an ODE by asking about existence of solutions. We see that  $x = 0$  is not allowed, so the largest possible intervals of existence are  $(0, \infty)$  and  $(-\infty, 0)$ . We also see that  $y = 0$  is a problem, so solutions are not allowed to cross the level  $y = 0$ . We conclude that there will be four distinct families of solutions, each “living” in a specific quadrant of the plane.

We already observed that this equation is separable, but let’s start again from the scratch. People who use differential equations in their work as a tool (in engineering, physics and other natural sciences) typically follow an old-style approach. It starts by writing the derivative in the given equation using the Leibniz notation:

$$3xy \frac{dy}{dx} = \frac{1}{xy}.$$

For practical purposes, an equation is separable if we can move all  $y$  to the left and all  $x$  to the right, including the differential bits (this is the part that makes mathematicians cringe).

$$3y^2 dy = \frac{dx}{x^2}.$$

Yup, it worked, it is a separable equation.

However, from the point of view of mainstream analysis  $dx$  and  $dy$  do not exist as separate objects, the “fraction”  $\frac{dy}{dx}$  is considered one symbol, a picture denoting derivative. We do not want to start lengthy discussions about alternative calculus theories or total differentials to justify it, instead we hide what we did by attaching magical signs to each side.

$$\int 3y^2 dy = \int \frac{dx}{x^2}.$$

Now the formula makes sense. Two remarks before we continue. There used to be a time when mathematics was done in this way. Unfortunately, some wrong results were derived, which is the reason why we no longer play around with  $dy$  and  $dx$  like this. Fortunately, problems happened on a far deeper level, the procedure we outlined here works. It is also very convenient, so it is still widely used in practice. We just do not show the intermediate stage (before attaching the integrals) to anyone, so people do not see what we did.

And doubly fortunately, using the language of analysis we can prove that the outcome of this procedure (the equality with integrals) is correct, we just have to go about it a bit differently. We will return to it below.

Anyway, now we are quite confident that we are doing the right thing and go on. We evaluate the two integrals.

$$y^3 = -\frac{1}{x} + C.$$

Why didn't we put "+C" on the left as well? If done properly, we should have written

$$y^3 + C_1 = -\frac{1}{x} + C_2.$$

We can move  $C_1$  to the right and denote the resulting arbitrary constant  $C_2 - C_1$  as  $C$ , so experienced people skip this intermediate step and write the convenient form right away.

What do we want? We want a formula for  $y$ . The cubic power is invertible, so we can make one more step:

$$y = \sqrt[3]{C - \frac{1}{x}}.$$

Is this any good? Actually, yes, we have ourselves a solution. In fact, we note that there is a parameter in the formula and we are solving a first order ODE, that's a perfect fit. Moreover, all our steps were equivalent (as long as  $y \neq 0$ , but this is prohibited anyway) so there is no other solution. We conclude that this is a general solution. Almost, we always have to include specifications of its validity.

As we explained in previous chapters, we want intervals, and we want them as large as possible. How does it look here? In general, we have to look at three sources of trouble: the equation as given, the procedure we used and the result itself.

We already observed that the equation forces us to demand that  $x \neq 0$ . We also had a condition  $y \neq 0$ , so we have to rule out the case  $C - \frac{1}{x} = 0$ . Once these two are satisfied, the steps we did in our calculations are valid, so no more trouble from there. Finally the formula that we arrived at has just one requirements  $x \neq 0$ , and we already took this into account.

So we have two requirements to work with,  $x \neq 0$  and  $\frac{1}{x} \neq C$ , and we notice that the true meaning of the second one depends on  $C$ . Thus there are several possible types of intervals for our solution:

- For  $C < 0$  we have possible solutions on intervals  $(-\infty, \frac{1}{C})$ ,  $(\frac{1}{C}, 0)$ , and  $(0, \infty)$ .
- For  $C > 0$  we have possible solutions on intervals  $(-\infty, 0)$ ,  $(0, \frac{1}{C})$ , and  $(\frac{1}{C}, \infty)$ .
- For  $C = 0$  we have possible solutions on intervals  $(-\infty, 0)$  and  $(0, \infty)$ .

This is somewhat awkward, so people adopted the following convention: When writing a general solution, we just specify conditions for existence and the understanding is that we can use the formula on any interval that does not contradict these conditions. That is, we then use the largest intervals that do not include the points that we banned.

Thus we would say that the general solution of the given problem is

$$y = \sqrt[3]{C - \frac{1}{x}}, \quad x \neq 0, C - \frac{1}{x} \neq 0.$$

How do we confirm that our answer is correct? By definition, we need to substitute our formula into the given equation. So consider any  $C \in \mathbb{R}$ , then we take some  $x$  that satisfies  $x \neq 0$  and  $\frac{1}{x} \neq C$  and compare the left-hand side and the right-hand side:

$$\begin{aligned} L &= 3xy(x)y'(x) = 3x\left(C - \frac{1}{x}\right)^{\frac{1}{3}} \left[\left(C - \frac{1}{x}\right)^{\frac{1}{3}}\right]' = x\left(C - \frac{1}{x}\right)^{\frac{1}{3}} \left(C - \frac{1}{x}\right)^{-\frac{2}{3}} \frac{1}{x^2} = \left(C - \frac{1}{x}\right)^{-\frac{1}{3}} \frac{1}{x}, \\ R &= \frac{1}{xy(x)} = \frac{1}{x\left(C - \frac{1}{x}\right)^{\frac{1}{3}}} = \frac{1}{x} \left(C - \frac{1}{x}\right)^{-\frac{1}{3}}. \end{aligned}$$

The equation is satisfied.

△

Note that while we are always expected to specify for which  $x$  this solution is valid, we did not

put anything about  $C$ . This is traditional, it is understood that when we see some  $C$  in a formula for a general solution, then it can be any real number.

The procedure we saw in this example is what people mean when they say “solving ODE by separation”. How far can we trust it? Let’s explore it in general. Consider some separable ODE  $y' = g(x)h(y)$ . The procedure above guides us to use the Leibniz notation and then proceed as follows:

$$\frac{1}{h(y)} dy = g(x) dx \implies \int \frac{1}{h(y)} dy = \int g(x) dx.$$

Then we integrated as if  $y$  was just another independent variable, obtaining antiderivatives  $H(y) = G(x) + C$ . If we can solve it for  $y$ , that is, if  $H$  is an invertible function, then we get  $y = H_{-1}(G(x) + C)$ .

How does it work when we return to the world of (mainstream) mathematics again and cannot treat  $dy$  as an independent object? We separate:

$$\frac{1}{h(y)} y' = g(x).$$

Does it make any sense to integrate? Actually it does, because in fact, on both sides we have a function with variable  $x$ .

$$\frac{1}{h(y(x))} y'(x) = g(x).$$

When two functions agree on some interval, then also their integrals must agree, up to some additive constant. The indefinite integral takes care of the constant and we write

$$\int \frac{1}{h(y(x))} y'(x) dx = \int g(x) dx.$$

On the right we use the antiderivative  $G(x)$  as before. The integral on the left has exactly the right form for substitution.

$$\int \frac{1}{h(y(x))} y'(x) dx = \left| \begin{array}{l} w = y(x) \\ dw = y'(x) dx \end{array} \right| = \int \frac{1}{h(w)} dw.$$

We see that we are looking for an antiderivative of  $\frac{1}{h}$ , exactly as in our “wrong” solution. If we call it  $H$ , the integral is

$$\int \frac{1}{h(y(x))} y'(x) dx = \dots = H(w) = H(y(x)).$$

We obtain the equation  $H(y(x)) = G(x) + C$  and we want to solve it for  $y(x)$ , arriving at  $y(x) = H_{-1}(G(x) + C)$  as before.

So we see that the procedure can be done properly, but it takes longer. Since the end result is the same, practical users of mathematics usually prefer the first approach. Experience suggests that it guides students – especially beginners – better.

Our analysis actually provided us with a general formula for the solution.

**Theorem 7.2.** (on existence for separable ODE)

Consider a separable ODE  $y' = g(x)h(y)$ . Assume that  $g$  is continuous on some open interval  $I$  and  $h$  is continuous on some open interval  $J$ . If  $h \neq 0$  on  $J$  then there is a solution to the given equation on  $I$ .

Let  $G(x)$  be an antiderivative of  $g(x)$  on  $I$  and  $H(y)$  be an antiderivative of  $\frac{1}{h(y)}$  on  $J$ . Then a general solution of the given equation can be expressed as  $y(x) = H_{-1}(G(x) + C)$  and it is valid on any subinterval of  $I$  that is mapped by  $G$  into the domain of  $H_{-1}$ .

Here  $H_{-1}$  is the inverse function of  $H$ . Note that the assumptions on  $h$  imply that  $h$  must be always positive on  $J$  or always negative on  $J$ . Then the same applies to  $\frac{1}{h}$ , consequently  $H$  must be strictly monotone on  $J$ . As a strictly monotone function, it is invertible, so we do not have to put this assumption in our theorem.

It would be possible to create an algorithm based on the formula  $y(x) = H_{-1}(G(x) + C)$ . Given a separable ODE, one simply identifies  $g$  and  $h$ , finds the relevant antiderivatives and substitutes into the formula. However, people actually prefer to follow the above process of separation. There are two reasons. First, it is easier to remember a procedure rather than an isolated formula.

Second, the procedure is more flexible than direct application of the theorem and does the necessary work with less fuss about it. For instance, if we wanted to solve example 7.a by applying the theorem, we would have to do it four times, applying it on each quadrant separately.

We have a nice procedure and an obligatory question comes up: What can go wrong? An obvious answer is that integration is known to be tricky. If we cannot evaluate the integrals, the show is over. Another tricky point is the last step, when we try to isolate  $y$  from some formula that we obtain. However, we will see shortly that even if  $H$  is not invertible, we can often still salvage something.

There is a third possible source of trouble, something might have gone wrong when playing with the expressions. Let's review the process.

We took the equation and manipulated it. Using the language of the theorem we changed it into the form  $\frac{y'}{h(y)} = g(x)$ . But that is exactly the moment where we should stop and ask ourselves a crucial question: Is this allowed? And the answer is that yes, we can do it, but only assuming that  $h(y)$  is not zero. Logic of mathematical reasoning tells us that what followed was not wrong, but it was incomplete. We only investigated the solutions that do not make  $h$  zero. However, there is no guarantee that the process we are studying in our equation is so well-behaved.

Thus in order to make a complete study of the given equation we also have to look at the situations when for some  $y_0$  we have  $h(y_0) = 0$ , because our procedure left that case unsolved. It turns out that there is a very simple answer.

**Fact 7.3.**

Consider a separable ODE  $y' = g(x)h(y)$ . If  $y_0$  satisfies  $h(y_0) = 0$ , then the constant function  $y(x) = y_0$  is a solution to the given ODE on any open interval  $I \subseteq D(g)$ .

When a constant solves a differential equation, it is called the **stationary solution** of this equation. We will look at them in detail in chapter 8. Right now the important thing to know is that when we are solving a separable ODE and we forget to check on stationary solutions, then there is a good chance that our general solution is not complete. Actually, we should worry even without knowing anything about differential equations. Already in elementary school teachers told us that when we manipulate an equation, we are not allowed to divide by zero. We should always be on guard when dividing by some expression.

We will show this in another example and this time we also include some Initial value problems to see how we handle initial conditions.

**Example 7.b:** Consider the equation  $y' = y^2 + xy^2$ .

1) Out first task is to find the general solution.

First we observe that there is no restriction coming from the equation itself. This means that unlike example 7.a, here the solutions will not be interrupted by conditions coming from the given equation.

Now we can proceed to the solution process itself and do our secret separation trick:

$$\frac{dy}{dx} = y^2 + xy^2 \implies \frac{dy}{y^2} = (1+x)dx.$$

This is the moment when we should say, wait a second, this is not allowed when  $y = 0$ . According to the fact above, we obtain a stationary solution  $y(x) = 0$  for  $x \in \mathbb{R}$ .

Now we attach integrals and evaluate.

$$\int \frac{dy}{y^2} = \int (1+x)dx \implies -\frac{1}{y} = x + \frac{1}{2}x^2 + C.$$

Thus

$$y = \frac{-1}{x + \frac{1}{2}x^2 + C} = \frac{2}{-2C - 2x - x^2}.$$

The formula will be nicer if we denote  $D = -2C$ .

$$y = \frac{-1}{x + \frac{1}{2}x^2 + C} = \frac{2}{D - 2x - x^2}.$$

The reader may have noticed already that people take certain liberties when it comes to solving ODEs and they like their traditions. One such tradition is that ODE people like to use  $C$  for their constants, and they do not make much fuss about it. In this particular example many would simply write  $C$  again instead of  $-2C$ , obtaining

$$y(x) = \frac{-1}{x + \frac{1}{2}x^2 + C} = \frac{2}{C - 2x - x^2}.$$

This does not mean that we are doing something like a substitution  $C = -2C$ ; rather, we may think that this new  $C$  is not the same as the previous one. If it smacks too much of wizardry to you, ignore it and use the version with  $D$ , I sometimes do.

The reader can check that this indeed is a solution (I swear I did). Where does it exist? First, we did the whole process under the assumption that  $y \neq 0$ . In this particular case this does not force any restrictions, because the formula  $\frac{2}{C-2x-x^2}$  never yields zero. This is very often the case (fortunately, see section 7a for what may happen here if we are not that lucky). There were also no restrictions from the equation itself, so we just need to worry about the resulting formula. There is one possible source of trouble, division by zero. Note the following:

$$C - 2x - x^2 = (C + 1) - 1 - 2x - x^2 = (C + 1) - (1 + x)^2.$$

We have three cases:

- If  $C < -1$ , then  $(C + 1) - (1 + x)^2 < 0 - (1 + x)^2 < 0$ , so we never divide by zero. These solutions exist on  $\mathbb{R}$  and are always negative.
- If  $C = -1$ , then  $y(x) = \frac{2}{-(x+1)^2}$ , it exists on  $(-\infty, -1)$  and on  $(-1, \infty)$  and it is also always negative.
- If  $C > -1$ , then there are two values of  $x$  where  $y$  does not exist and we obtain three intervals for the solution. On intervals  $(-\infty, -1 - \sqrt{C + 1})$  and  $(-\infty, -1 + \sqrt{C + 1})$  the solution is negative, on  $(-1 - \sqrt{C + 1}, -1 + \sqrt{C + 1})$  the solution is positive.

Let's sum it up (and recall the stationary solution): The general solution is

$$y(x) = 0 \text{ for } x \in \mathbb{R} \quad \text{or} \quad y(x) = \frac{2}{C - 2x - x^2},$$

where  $x \in \mathbb{R}$  if  $C < -1$  and  $x \neq -1 \pm \sqrt{C + 1}$  otherwise.

Sometimes people would skip this general analysis, especially when they know that some initial conditions are coming (like in this example). Indeed, very soon we will investigate validity of particular solutions, then  $C$  have concrete values and the reasoning is fairly straightforward. Thus a common way to express our result is

$$y(x) = 0, x \in \mathbb{R} \quad \text{or} \quad y(x) = \frac{2}{C - 2x - x^2}, C - 2x - x^2 \neq 0.$$

2) Now we will try several Initial value problems.

a) Find the solution of the IVP  $y' = y^2 + xy^2$ ,  $y(1) = \frac{2}{3}$ .

The best way to solve such a problem is to first find a general solution, which we did above. That formula actually represents infinitely many solutions and we have to choose the right one. This is done in a natural way, we simply look at what we want.

$$y(1) = \frac{2}{3} \iff \frac{2}{C - 2 \cdot 1 - 1^2} = \frac{2}{3} \iff C = 6.$$

We know the right formula now,  $y = \frac{2}{6 - 2x - x^2}$ , but not the right solution, we still have to determine where this solution lives. The condition  $6 - 2x - x^2 = 0$  yields  $x = -1 \pm \sqrt{7}$ , so the real line is interrupted at points  $-1 - \sqrt{7} \approx -3.6$  and  $-1 + \sqrt{7} \approx 1.6$ . This creates three possible intervals and we have to choose the right one. We recognize it easily, it must allow for the initial condition, which means that it must make the initial time  $x_0 = 1$  available. The middle interval does it for us.

Answer: The solution is  $y_a(x) = \frac{2}{6 - 2x - x^2}$ ,  $x \in (-1 - \sqrt{7}, -1 + \sqrt{7})$ .

b) Find the solution of the IVP  $y' = y^2 + xy^2$ ,  $y(0) = -1$ .

Initial condition:

$$y(0) = -1 \iff \frac{2}{C - 0} = -1 \iff C = -2.$$

The formula  $\frac{2}{-2 - 2x - x^2}$  exists for all real  $x$ , no problem here.

Answer: The solution is  $y_b(x) = \frac{-2}{2 + 2x + x^2}$ ,  $x \in \mathbb{R}$ .

c) Find the solution of the IVP  $y' = y^2 + xy^2$ ,  $y(2) = -1$ .

Initial condition:

$$y(2) = -1 \iff \frac{2}{C - 8} = -1 \iff C = 6.$$

The formula is  $y = \frac{2}{6 - 2x - x^2}$ , we already know the forbidden points  $-1 - \sqrt{7} \approx -3.6$  and  $-1 + \sqrt{7} \approx 1.6$ . The initial time  $x_0 = 2$  is to the right of both.

Answer: The solution is  $y_c(x) = \frac{2}{6 - 2x - x^2}$ ,  $x \in (-1 + \sqrt{7}, \infty)$ .

Note that although the formula is the same as in the problem a), this is not the same solution because their regions of validity differ. Going to our favourite interpretation, these are events that are happening at different times. The flow of information is interrupted at time  $\sqrt{7} - 1$ , so these might be two totally unrelated processes that happen to be given by the same formula.

d) Find the solution of the IVP  $y' = y^2 + xy^2$ ,  $y(-2) = 0$ .

Initial condition:

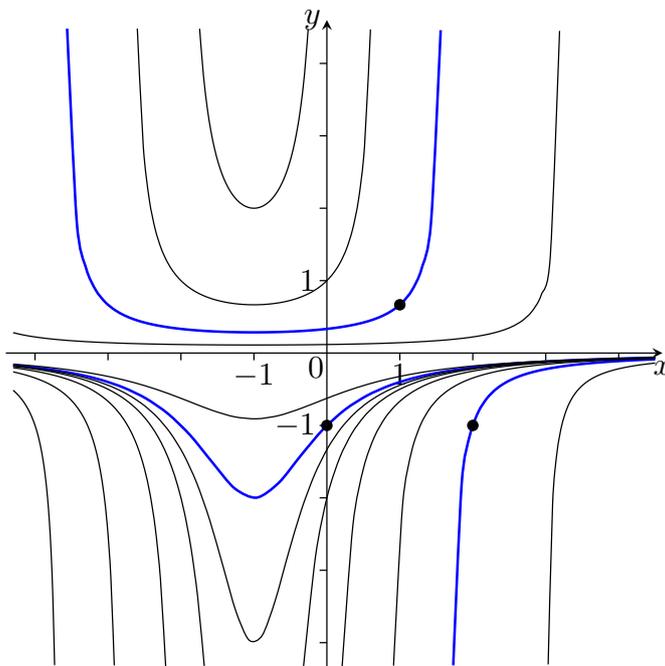
$$y(-2) = 0 \iff \frac{2}{C} = 0 \iff C = ???$$

One would be tempted to say that for this particular initial condition there is no solution (and in many cases this would be true). However, in this case we actually have another chance, the stationary solution. And indeed, it works, we got lucky.

Answer: The solution is  $y_d(x) = 0$ ,  $x \in \mathbb{R}$ .

It is really easy to overlook the stationary solution when solving IVPs, and then we reach a wrong conclusion. Fortunately, in many cases it can be included in the general formula, as we will see in example 7.c, so missing a stationary solution is not fatal then.

To give the reader some idea of what is going on, we show below the first three solutions with their initial conditions and also some other solutions.



△

Let's revisit problem b) with initial condition  $y(0) = -1$ . The answer

$$y_b(x) = \frac{-2}{2 + 2x + x^2}, \quad x \in \mathbb{R}$$

that we gave is “mathematical”. This is how mathematicians see ODEs and there is nothing wrong with it. However, if this problem came from some application, then there is a good chance that the situation is different, namely that there is actually no information what happened before the initial condition. Consequently, for an applied person it makes more sense to state this solution as

$$y_b(x) = \frac{-2}{2 + 2x + x^2}, \quad x \geq 0.$$

The interpretation is natural, we specify how the event started and the formula specifies how it continued, it does not really make real sense to move back in time. Note that we do not have an open interval as the region of validity. The expectation is that our function is a solution in the mathematical sense on the open interval  $(0, \infty)$ , and it is continuous at  $x_0 = 0$  from the right.

In this book we adopt the mathematical point of view, but the reader should be aware that another, equally legitimate viewpoint exists.

**7.4 Remark:** Our differential equation  $y' = (1+x)y^2$  is of the form  $y' = f(x, y)$ . How does it fit in with the general theory from chapter 6?

The function  $f$  is continuous on  $\mathbb{R} \times \mathbb{R}$ , so for every initial condition there will be a local solution (theorem 6a.1). How about the more powerful Picard theorem? We need to inquire about the Lipschitz property.

We use the approach via derivative, see theorem 6a.3. We have

$$\frac{\partial f}{\partial y} = 2(1+x)y.$$

This continuous expression is bounded on any bounded rectangle  $I \times J$ , so the conclusion on uniqueness is true on these. Since one can cover any part of the plane with such rectangles, it follows that for every initial condition we have a local solution and this solution must be unique.

By extending local solutions using such rectangles that eventually cover the whole plane, we argue that local solutions can be extended all the way to the boundary of  $\mathbb{R}^2$ . This implies that solutions to this differential equation extend all the way to infinity (or negative infinity). However, we do not know which one. The solutions may extend to infinity in the horizontal direction, or

in the vertical direction. The picture above (or the analysis of domains depending on  $C$ ) show that both cases do happen: There are solutions that exist on  $\mathbb{R}$ , solutions that exist on bounded intervals and diverge at their endpoints, and solutions that include both types of behaviour.

△

We are ready to outline the general procedure.

**Algorithm 7.5.**

⟨solving separable ODE by separation⟩

Given: a differential equation that can be written as  $y' = g(x) \cdot h(y)$ .

1. Write the equation as  $\frac{dy}{dx} = g(x)h(y)$ . Move all  $x$  (including  $dx$ ) to the right and all  $y$  (including  $dy$ ) to the left, add integration signs:

$$\frac{dy}{dx} = g(x)h(y) \implies \int \frac{dy}{h(y)} = \int g(x) dx.$$

2. Explore the possibility that  $h(y) = 0$ , leading to possible stationary solutions.

3. Assuming that  $h(y) \neq 0$ , integrate both sides.

$$\int \frac{dy}{h(y)} = \int g(x) dx \implies H(y) = G(x) + C.$$

4. If possible, express  $y$  as a function of  $x$ :

$$H(y) = G(x) + C \implies y(x) = H_{-1}(G(x) + C).$$

It may happen that there are more possibilities for  $y$ , each will give rise to one family of solutions.

5. Exploring the given equation and the solution we obtained, determine conditions for its validity.

6. If an initial condition is given, determine the corresponding particular solution and the maximal interval of its validity.

△

The last step is usually very straightforward: Given the condition  $y(x_0) = y_0$ , we substitute  $x_0$  into the general solution, set it equal to  $y_0$  and solve the resulting equation for  $C$ . Then we find the largest interval that contains  $x_0$  and does not violate conditions of validity of the solution.

Sometimes it gets more interesting. It may happen that the initial condition cannot be satisfied by our general solution, then the answer is that the given IVP is not solvable. Or it may happen that the given condition cannot be satisfied by the formula that we found, but it is actually satisfied by some stationary solution. It is really important not to forget to check on their existence. Finally, sometimes separation does not yield one, but several possible formulas. Then we have to choose the right one for our particular solution. This is typically done using  $y_0$ .

We noted above that the general separation approach can handle more situations than the strict theorem. There are also situations that appear often and it is good to know shortcuts. We will show some popular tricks in another comprehensive example.

**Example 7.c:** Consider the differential equation  $2yy' = y^2 - 1$ .

Traditionally we start by checking whether the equation itself already restricts us in some way. We see that the expressions on both sides are always well-defined, so there is no restriction from this source.

Is this equation separable? We try separating using the traditional approach.

$$2y \frac{dy}{dx} = y^2 - 1 \implies \int \frac{2y}{y^2 - 1} dy = \int 1 \cdot dx$$

Yes, we were able to separate, and it is the time to inquire whether our procedure is correct. What we did right now is not allowed if  $y = \pm 1$ , so we have ourselves two stationary solutions,  $y(x) = -1$  on  $\mathbb{R}$  and  $y(x) = 1$  on  $\mathbb{R}$  (there were no restrictions on  $x$  in the equation).

Having this out of the way, we may assume that  $y^2 - 1 \neq 0$  and continue with the equation. The left integral is evaluated using substitution  $w = y^2 - 1$ .

$$\int \frac{2y}{y^2 - 1} dy = \int 1 dx \implies \ln |y^2 - 1| = x + C.$$

Now we would like to solve for  $y$  and the first step is obvious, we apply exponential to both sides.

$$|y^2 - 1| = e^{x+C} = e^C \cdot e^x.$$

We need to get rid of the absolute value, but that is possible only if we know the sign of the expression inside, which we don't. Thus we have to look at possible cases. This part of solution is done under the assumption that  $y^2 - 1 \neq 0$ , so only two cases are left.

If  $y^2 - 1 > 0$ , then we have  $y^2 - 1 = e^C e^x$ . If  $y^2 - 1 < 0$ , we have  $-(y^2 - 1) = e^C e^x$ , that is,  $y^2 - 1 = -e^C e^x$ . Now we use a very specific argument, we join the two versions by writing  $y^2 - 1 = \pm e^C e^x$ . Note that  $C$  was an arbitrary number, so  $e^C$  represents an arbitrary positive number. Then  $\pm e^C$  represents an arbitrary non-zero number, we can call it  $D$ . Thus we can join the two possibilities by writing  $y^2 - 1 = D e^x$ . This is a very handy procedure that is used very often, so often that experienced ODE solvers do it in one step

$$\ln |v(x)| = w(x) \implies v(x) = D e^{w(x)}$$

without going into detail. After you've seen it a few times, you'll very likely do it like that, too. And not surprisingly, many people put  $C$  instead of  $D$  again, but in this book we will keep  $D$  whenever we use this approach to remind the reader of what we did.

Back to our problem, we arrived at the equation  $y^2 - 1 = D e^x$ , that is,  $y^2 = D e^x + 1$ , and we need to express  $y$ . We know that this equation cannot be properly solved for  $y$ ; however, we also know that there are two expressions that satisfy it, namely  $y = \sqrt{D e^x + 1}$  and  $y = -\sqrt{D e^x + 1}$ .

This means that there will be two families of solutions. Now we need to discuss region of validity for them, in this case we need to make sure that  $D e^x + 1 \geq 0$ . We thus have a candidate for the general solution

$$y = \pm \sqrt{D e^x + 1}, \quad D e^x + 1 \geq 0.$$

However, there is still some work to do. We worked under the assumption that  $y^2 \neq 1$ , how about our formula? Recall that the constant  $D$  that we obtained cannot have value zero, so our solutions do not have a problem with this condition. Good.

Another issue to address is this constant  $D$ , because we are not used to have a restriction on its value. And this is exactly the time to recall that we in fact have two stationary solutions that should be listed here as well so that the reader has a complete picture. By a remarkable coincidence, when we take  $D = 0$ , our formula yields exactly those stationary solutions  $y(x) = \pm 1$ . Talk about luck. Actually, experienced ODE solvers would know that this happens when we use the trick with logarithm and absolute value, so it is not so surprising.

The important thing is that we can remove the restriction on  $D$  and at the same time provide a complete answer in one formula (which is actually two formulas).

The general solution is

$$y = \pm \sqrt{D e^x + 1}, \quad D e^x + 1 > 0.$$

It is understood that this expression actually means that there are two formulas to choose from.

2) Let's see how it works with initial conditions.

a) Find the solution of the IVP  $2yy' = y^2 + 1$ ,  $y(1) = 2$ .

This time there is a preliminary round, because there are two candidates provided by the general solution. The initial condition has  $y_0 = 2 > 0$  and it is obviously impossible to reach this value using the formula  $-\sqrt{D e^x + 1}$ , so we will use the other one.

Initial condition:

$$y(1) = 2 \iff \sqrt{D e^1 + 1} = 2 \iff D = \frac{3}{e}.$$

We obtain  $y(x) = \sqrt{3e^{-1}e^x + 1} = \sqrt{3e^{x-1} + 1}$ , the expression inside the root is always at least 1 so no problem with existence there.

Answer: The solution is  $y_a(x) = \sqrt{3e^{x-1} + 1}$ ,  $x \in \mathbb{R}$ .

b) Find the solution of the IVP  $2yy' = y^2 + 1$ ,  $y(0) = -1$ .

This time the initial condition has  $y_0 = -1 < 0$ , so we need the formula that produces negative numbers. Initial condition:

$$y(0) = -1 \iff -\sqrt{De^0 + 1} = -1 \iff D = 0.$$

We get  $y(x) = -1$ . Well, we could have noticed right away that this is the stationary case.

Answer: The solution is  $y_b(x) = -1$ ,  $x \in \mathbb{R}$ .

c) Find the solution of the IVP  $2yy' = y^2 + 1$ ,  $y(0) = 0$ .

This is funny, we cannot decide which version of the general solution to use. We try both, starting with the positive version.

$$y(0) = 0 \iff \sqrt{De^0 + 1} = 0 \iff D = -1.$$

We obtain  $y(x) = \sqrt{1 - e^x}$ , this exists when  $1 - e^x \geq 0$ , that is,  $x \leq 0$ . Now we try the other version.

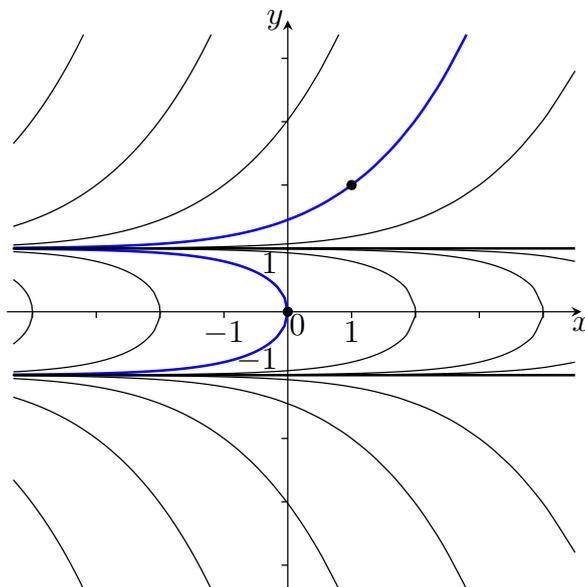
$$y(0) = 0 \iff -\sqrt{De^0 + 1} = 0 \iff D = -1.$$

We obtain  $y(x) = -\sqrt{1 - e^x}$ .

Answer: There are two solutions,  $y(x) = \sqrt{1 - e^x}$ ,  $x \leq 0$  and  $y(x) = -\sqrt{1 - e^x}$ ,  $x \leq 0$ .

It turns out that if we interpret  $x$  as time, then our solutions actually end at time  $x_0 = 0$ , so intuitively we feel as if we had an “ending condition” here. From mathematical point of view these solutions are fine, because “initial condition” is a formal notion and in its definition there is no requirement on the position of  $x_0$  in the region of validity. However, from an applied point of view discussed in the previous example it would seem that the event we study here started at time  $x_0$  and ended immediately.

We again show a picture with some solutions.



△

This equation does not provide unique solutions, which is curious. In particular this must mean that assumptions of the Picard theorem 6a.2 are not all satisfied. What gives? First we should rewrite the given equation into the appropriate form:

$$y' = \frac{y^2 - 1}{2y}.$$

The function  $f(x, y)$  on the right is not defined at  $y = 0$ , so we cannot apply that theorem to our

equation on any neighborhood of points  $(x, 0)$ . In other words, we suspect trouble when the initial condition has  $y_0 = 0$  regardless of the initial time  $x_0$ .

We can also observe that when we choose  $(x_0, y_0)$  so that  $y_0 \neq 0$ , then we can find a bounded rectangle that contains this point and has a certain positive distance from the  $x$ -axis. On such a rectangle the derivative  $\frac{\partial f}{\partial y}(x, y) = \frac{1}{2} + \frac{1}{2y^2}$  is continuous and hence bounded, so by theorem 6a.3, uniqueness is guaranteed.

**7.6 Remark:** Note that there are some sensitive moments in the solution as presented.

When we were dealing with the absolute value, there were two possibilities for  $y^2 - 1$  to choose from,  $e^C \cdot e^x$  and  $-e^C \cdot e^x$ . When we decided to use  $D e^x$ , we in fact forced  $y^2 - 1$  to take one definite sign everywhere on its domain. However, what if some solution  $y$  wants to change the sign of  $y^2 - 1$  at some place? Our procedure did not address this case.

Let's explore this situation. Imagine a solution  $y$  so that  $y^2 - 1$  is positive on some interval  $I$ . Because of continuity, if this solution wants to change the sign of  $y^2 - 1$  into negative somewhere, it should first go through some point where  $y^2 - 1 = 0$ . However, on this interval  $I$  where  $y^2 - 1 > 0$  we can apply our procedure above and learn that  $y^2 - 1 = D e^x$  with  $D > 0$  on  $I$ . Obviously, this can never become zero, so the switch of sign is not possible. If a solution has some place where  $y^2 - 1 \neq 0$ , then it has to keep the chosen sign everywhere where it exists.

This is actually fairly typical. We commented above about a popular trick

$$\ln|v(x)| = w(x) \implies v(x) = D e^{w(x)}$$

It can be shown that when we use this in our solution, then there will be no trouble with changing signs and our procedure yields complete information.

There was another moment when our solutions had to make a choice. There were two possibilities for  $y$ ,  $\sqrt{D e^x + 1}$  and  $-\sqrt{D e^x + 1}$ , and we forced it to stick with just one all the time. Could it be that we actually left out a whole group of solutions, those that are wishy-washy in their choices of signs? This is a very loaded question.

As above, if a solution wants to change its sign, it would have to pass through a point where  $y = 0$ . Is that possible? Unlike the previous case the answer is positive. For instance, if we have a solution that is negative on some interval, then it would have to be given by the formula  $-\sqrt{D e^x + 1}$  there. This can attain value zero when  $D < 0$  and  $x = -\ln(|D|)$ , which could be an opportunity for  $y$  to change signs. However, in this particular example we do not have to worry, because by a remarkable coincidence this  $x_0$  is also the right endpoint of the domain for these solutions. Thus all solutions that can attain value zero do not exist past that point and cannot in fact change signs.

The conclusion is that no "switching" solutions are possible and our solution was complete.

△

The good news is that complications are rare in applications, so the separation algorithm outlined above (and the trick with absolute value) usually lead directly to correct solutions. In fact, many practically oriented courses on differential equations even leave out such discussions entirely, so students are not aware that the separation procedure may sometimes lead to trouble. A reader who is just starting with differential equations may prefer to stay at this level, skip the next part and perhaps solve some problems from the Exercises to see that solving differential equations by separation can be really nice and easy.

For those who wish to look closer at more challenging situations we offer the next section.

## 7a. Gluing solutions

Here we will explore some possible outcomes of complications with the separation procedure, in particular we look at situations when there is no clear separation between solutions obtained by different branches of the separation procedure. Such situation appear mostly in artificial examples, but not exclusively, and they may lead to interesting behaviour of solutions.

**Example 7a.a:** Consider the equation  $y' = 3y^{2/3}$ .

First we note that this expression makes sense everywhere, so there is no restriction as yet. We start with separation.

$$\frac{dy}{dx} = 3y^{2/3} \implies \int \frac{dy}{y^{2/3}} = \int 3dx.$$

We see a stationary solution  $y(x) = 0$ .

Assuming that  $y \neq 0$  we push on.

$$\int y^{-2/3} dy = \int 3dx \implies 3y^{1/3} = 3x - 3C \implies y = (x - C)^3.$$

We used  $-3C$  for integrating constant because obviously the 3 would come handy, and experience told us that working with  $x - C$  is easier than working with  $x + C$  (for instance when solving for zeros).

What can we say about validity of this solution? There is no restriction in the equation itself or the formula, so one would be tempted to claim that the domain is  $\mathbb{R}$ . However, our procedure is only valid for those solutions that satisfy  $y \neq 0$ . Consequently, our procedure can only provide solutions of the form  $y(x) = (x - C)^3$ ,  $x \neq C$ .

Take some  $C \in \mathbb{R}$ . We have solutions  $y_1(x) = (x - 1)^3$  on  $(-\infty, C)$  and  $y_2(x) = (x - 1)^3$  on  $(C, \infty)$ . However, the number  $x = C$  causes no trouble in the given equation and the functions themselves also have no trouble at  $x = C$ , so there does not seem to be a reason why our solutions should be interrupted there, it's just that the separation procedure cannot handle them as a whole.

Is it possible to connect these two solutions into one? We know that  $y_1$  and  $y_2$  are solutions, so the formula  $(x - C)^3$  satisfies the given ODE for  $x \neq C$ . For  $x = C$  we check this directly, and we conclude that in fact this expression satisfies our equation everywhere. We can conclude that  $y(x) = (x - C)^3$  is a solution on  $\mathbb{R}$ .

Is it a general solution? In other words, are these the only solutions? Let's take a closer look at the process of gluing two solutions together. Take  $y(x) = (x - C)^3$  on  $(-\infty, C)$ . Since  $y < 0$  there, the separation procedure applies on this interval, which means that our solution cannot fork on this interval. However, at  $x_0 = C$  the separation procedure that lead to this formula stops being relevant. We now actually have a choice which branch of the separation procedure to use to go on. Above we chose to use the same part of the procedure, which yielded the solution  $y(x) = (x - C)^3$  on  $\mathbb{R}$ .

However, there is another possibility, because the separation procedure also provided us with the stationary solution that passes through the same point. Would it work? Consider a function given by  $y(x) = (x - C)^3$  for  $x < C$  and  $y(x) = 0$  for  $x \geq C$ . We know that this function will solve our equation on the intervals  $(-\infty, C)$  and  $(C, \infty)$ . By checking on one-sided derivatives we learn that

$$[(x - C)^3]'_-(C) = 0, \quad [0]'_+(C) = 0,$$

so  $y'(C) = 0$ . Substituting into our ODE we see that this  $y$  also solves it at  $x = C$ . We conclude that this  $y(x)$  is a solution on  $\mathbb{R}$ .

In general, when some solution reaches the level  $y = 0$  at some  $x_0 \in \mathbb{R}$ , we can decide which way to go next, whether to follow along  $y = 0$  or along  $(x - x_0)^3$ . In particular, when we follow the stationary solution, we can at any point split away and go on along a suitable cubic curve.

Thus there are many possible forms of solutions, in fact there are four possible general scenarios. A solution may or may not start with a cubic part, and it may or may not end with a cubic part, with a constant segment in the middle that can have length zero if two cubic parts follow immediately.

Thus the correct answer to the "general solution problem" goes as follows:

The general solution is

$$y(x) = \begin{cases} (x - C_1)^3, & x \leq C_1; \\ 0, & C_1 \leq x \leq C_2; \\ (x - C_2)^3, & C_2 \leq x \end{cases} \text{ or } y(x) = \begin{cases} (x - C_1)^3, & x \leq C_1; \\ 0, & C_1 \leq x \end{cases} \text{ or } y(x) = \begin{cases} 0, & x \leq C_2; \\ (x - C_2)^3, & C_2 \leq x \end{cases}$$

for any choice  $C_1 \leq C_2 \in \mathbb{R}$ .

Surprisingly, a seemingly simple differential equation resulted in a rather complicated situation. This also has to be taken into account when solving initial value problems.

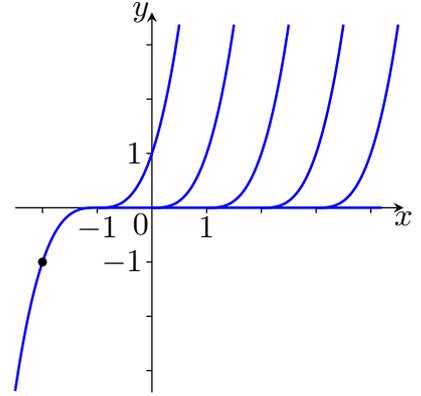
For instance, if we start with an initial condition  $y(x_0) = y_0$  and  $y_0 < 0$ , then we obtain a unique cubic curve passing through this point. But then it reaches the  $x$ -axis and it can follow it as long as it feels like, perhaps going all the way up to infinity. Or it may leave it again using another cubic curve. The picture suggests how this may look.

On the other hand, if our initial condition has  $y_0 > 0$ , then it determines uniquely which cubic function will be chosen to leave the  $x$ -axis, but we can get there using any cubic function we want as long as it gets to the  $x$ -axis in time, before the exiting cubic curve begins.

And if we choose an initial condition with  $y_0 = 0$ , then we can enter and leave the  $x$ -axis whenever we feel like as long as we spent the time  $x_0$  on it.

In short, things are really interesting.

△



Do we really have to be always on our toes, looking for this kind of trouble? Fortunately not. Note that possibility to glue solutions like that means that there are initial value problems with multiple solutions, which we do not expect in nature. In other words, when our differential equations come from applications, we can be almost sure that things like that do not happen. Is there a way to see this mathematically?

Generally, the gluing complications only arise when different solutions meet. When we follow the separation procedure, we end up with one or more algebraic formulas for solutions, and perhaps one or more stationary solutions, some of which can be actually incorporated into the formulas (we do not have to worry about these then) and some stand apart. It is usually fairly simple to check whether solutions of different kind can meet somewhere, and also whether two different solutions of the same kind can meet. In effect, we are asking whether is it possible for some initial condition to have more corresponding particular solutions. If the answer is in the negative, then we do not have to worry.

This approach can be combined with existence and uniqueness analysis, because we can glue different kinds of solutions only in places where forking is possible. Typically, the Picard theorem 6a.2 rules out large areas on uniqueness and leaves only a small suspicious set to check on.

How would this work with our example? We can see from the derivative

$$\frac{\partial f}{\partial y}(x, y) = \frac{2}{3} \frac{1}{y^{1/3}}.$$

that the only trouble is around  $y = 0$ , because this function is bounded on regions of the form  $\mathbb{R} \times (a, \infty)$  and  $\mathbb{R} \times (-\infty, -a)$  for  $a > 0$ . This tells us to inquire about the situation  $y = 0$  for our solutions, which focuses our effort.

It is useful to go back to our previous examples and see how we stand there.

**Example 7a.b:** In example 7.a we had the equation  $3xyy' = \frac{1}{xy}$ . We obtained the formula

$$y = \sqrt[3]{C - \frac{1}{x}}$$

for the general solution. Do we have to worry? We have just one family, so we check whether two such solutions can meet somewhere:

$$\left(C_1 - \frac{1}{x_0}\right)^{1/3} = \left(C_2 - \frac{1}{x_0}\right)^{1/3} \implies C_1 - \frac{1}{x_0} = C_2 - \frac{1}{x_0} \implies C_1 = C_2.$$

We see that when we assumed two function meeting somewhere, it turned out that it was in fact the same function. Thus there is no need to worry.

An alternative approach would be to take arbitrary  $x_0, y_0 \neq 0$  (see the restriction above) and ask how many solutions can pass through this point. Solving the equation

$$\left(C - \frac{1}{x_0}\right)^{1/3} = y_0$$

for  $C$  shows that there is only one possibility for  $C$ , hence just one solution passing through.

For the sake of practice, let's try our sophisticated approach. After isolating  $y'$  we end up with  $f(x, y) = \frac{1}{3x^2y^2}$ . We differentiate:

$$\frac{\partial f}{\partial y}(x, y) = \frac{-2}{3x^3y^3}.$$

We immediately smell trouble around  $x = 0$  and  $y = 0$ . But we also see that this expression is continuous elsewhere, and that it does not grow to infinity as we move  $x, y \rightarrow \infty$ . Thus  $x = 0$  and  $y = 0$  are the only sources of possible trouble. However, the equation itself rules these points out, so no solutions can meet there and we know that we do not have to worry about gluing.

Now what would happen if we were given this equation in the form  $3x^2y^2y' = 1$ ? Then the suspicious cases would not be automatically ruled out and we would have to investigate whether more solutions meet, leading us back to the simple approach solution at the beginning of this example, but more focused, as we would only have to check on what happens when  $x = 0$  and when  $y = 0$ .

△

**Example 7a.c:** Now we return to example 7.b. For the equation  $y' = y^2 + xy^2$  we found the general solution

$$y(x) = 0 \text{ for } x \in \mathbb{R} \quad \text{or} \quad y(x) = \frac{2}{C - 2x - x^2}, \quad C - 2x - x^2 \neq 0.$$

Is there any possibility of gluing? We have a stationary solution here, which could mean danger, but note that the other formula can never yield zero, so the two types of solutions never meet. Could two solutions given by that formula meet? Given a point  $(x_0, y_0)$ , a solution passing through it must have

$$\frac{2}{C - 2x_0 - x_0^2} = y_0 \implies C = \frac{2}{y_0} + 2x_0 + x_0^2,$$

which is unique. There is no opportunity for gluing.

What can we learn from Picard's theorem? The function

$$\frac{\partial f}{\partial y}(x, y) = 2(1 + x)y$$

does not have any trouble at proper points, it is continuous everywhere. Thus the only problem with boundedness happens if  $x$  or  $y$  grow beyond any bound. This means that if we restrict ourselves to any bounded rectangle  $I \times J$ , the derivative  $\frac{\partial f}{\partial y}$  will be bounded there, implying uniqueness of solutions. Since every point of every solution lies inside some bounded rectangle, we see that solutions are unique everywhere, there is no forking (and hence no gluing) possible.

△

**Example 7a.d:** Now we return to example 7.c. For the equation  $2yy' = y^2 - 1$  we obtained two possible solutions,

$$y = \sqrt{De^x + 1} \quad \text{and} \quad y = -\sqrt{De^x + 1}.$$

They also include stationary solution, so at least we do not have to worry about those. Is it possible for those two formulas to meet? Given their form, it is possible if the two roots are equal to zero, that is, if for some  $x_0$  we have  $\sqrt{De^{x_0} + 1} = 0 = -\sqrt{De^{x_0} + 1}$ . As we already saw, it is possible for this to happen, but then this  $x_0$  happens to be the endpoint of the interval of validity, so those two solutions meet at their ends, there is no opportunity for gluing. This example was therefore also safe from trouble.

△

As a bonus we will show one example with even more possibilities for gluing. This example also shows that gluing can have impact on initial value problems.

**Example 7a.e:** Consider the equation  $2y' = 3y^{1/3}$ . The solution proceeds just like in the analogous example 7a.a, with a small change that makes the ending somewhat different.

We start with separation, and as experienced separators we already see that the stationary solution  $y(x) = 0$  is coming and continue under the assumption  $y \neq 0$ .

$$\begin{aligned} 2\frac{dy}{dx} = 3y^{1/3} &\implies \int 2y^{-1/3}dy = \int 3dx \implies 3y^{2/3} = 3x - 3C \\ &\implies y^2 = (x - C)^3 \implies y(x) = \pm\sqrt{(x - C)^3}. \end{aligned}$$

Both formulas are solutions valid for  $x \geq C$  (check).

Now we start investigating possible overlaps. First, if we take  $(x_0, y_0)$  with  $y_0 > 0$ , then only one of these formulas can (potentially) reach this point and we easily find that then  $C$  is determined uniquely:

$$\sqrt{(x_0 - C)^3} = y_0 \implies C = x_0 - y_0^{2/3}.$$

Similarly for  $y_0 < 0$  we have to use the other family and get unique solutions. This leaves us with points of the type  $(x_0, 0)$  as candidates for gluing problems. And indeed, analysis using the Picard theorem would guide us to them as well.

We immediately that when we try to get value zero using our solutions at some point  $x_0$ , we can in fact choose from all three families. However, that does not automatically mean that we are allowed gluing there, we have to show that such a glued function is a solution.

It pays to start by observing that solutions from the two families given by formulas actually reach the value  $y = 0$  at the left endpoints of their domains, that is, they all start on the  $x$ -axis and move away from it. Thus the only gluing available is to start with the stationary solution  $y(x) = 0$  and branch away from it (up or down) at some point  $x_0$ . This branching solution that has to be of the form  $\pm\sqrt{(x - x_0)^3}$  in order to touch the  $x$ -axis at  $x_0$ . What happens there? The stationary solution has derivative 0 there. Since  $[(x - x_0)^{3/2}]' = \frac{3}{2}(x - x_0)^{1/2}$ , we easily see that  $\pm\sqrt{(x - x_0)^3}$  has zero as its derivative from the right at  $x_0$ . The glued function therefore has derivative 0 at  $x_0$  and we check that it satisfies the given equation there. It also satisfies it on intervals  $(-\infty, x_0)$  and  $(x_0, \infty)$ , so we have solutions on  $\mathbb{R}$ .

We conclude that maximal solutions in fact look like this:

$$y(x) = \begin{cases} 0, & x \leq C; \\ \sqrt{(x - C)^3}, & C \leq x \end{cases} \quad \text{or} \quad y(x) = \begin{cases} 0, & x \leq C; \\ -\sqrt{(x - C)^3}, & C \leq x \end{cases} \quad \text{or} \quad y(x) = 0, \quad x \in \mathbb{R}.$$

To illustrate practical impact of gluing, imagine that we are trying to find the solution that satisfies  $y(2) = -8$ . The negative sign points us to the formula with negative square root and we calculate:

$$-8 = -\sqrt{(2 - C)^3} \implies 4 = 2 - C \implies C = -2.$$

It would be natural to claim that  $y(x) = -\sqrt{(x + 2)^3}$ ,  $x \in [-2, \infty)$  is the answer.

However, such answer would not be quite correct, because we always want the maximal solution, that is, the solution on the largest possible interval. We did a proper analysis and thus we know that the function we found starts from the  $x$ -axis, and we can extend it to the left using our stationary solution. The correct answer is therefore

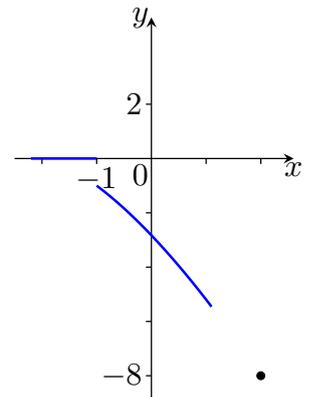
$$y(x) = \begin{cases} 0, & x < -2; \\ -\sqrt{(x + 2)^3}, & -2 \leq x, \end{cases}$$

yielding a solution on the whole  $\mathbb{R}$ .

Aren't you glad that troubles like this are rare?

△

This concludes the last (perhaps a bit scary) section. We once again thank our mother nature that it protects us from such trouble in applications. It is not actually that hard, is it, but it is nice to have things easy in math once in a while.



## 8. Analyzing solutions

It often happens that one encounters a differential equation that cannot be solved analytically, that is, we are not able to obtain an algebraic formula for a solution. However, sometimes we are able to find out things about those unknown solutions just by looking at the equation.

The simplest, but already very useful, is the analysis of trends that solutions have. Do they grow or decrease? For that we need information about the sign of the derivative, and by a lucky coincidence, a differential equation of the form  $y' = f(x, y)$  provides exactly this (did we say that this type is very nice?). We also see that we are in fact getting information about the rate of growth of a certain solution  $y(x)$  depending on the points  $(x, y)$  through which this solution passes.

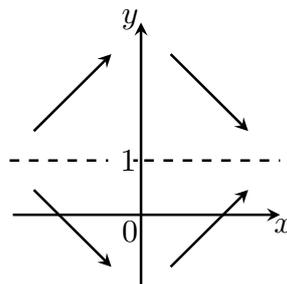
How do we do it? We simply check on the sign of the expression  $f(x, y)$ . Typically,  $\mathbb{R}^2$  (or rather the part of it where  $f$  is defined) splits into regions where the sign of  $f$  is the same. We thus obtain basic information regarding the trends of solutions that pass through such regions. The most efficient way to identify these regions is to find where the sign changes, that is, look at the set of points given by the equation  $f(x, y) = 0$ . This will show us the boundaries of those regions. Actually, the sign can also change at places where  $f$  is not defined, so this should be investigated as well.

**Example 8.a:** Consider the differential equation  $y' = x - xy$ . What can we say about solutions? First, we note that there is no problem with the equation, so no values of  $x$  and  $y$  are forbidden. We therefore expect solutions through every point of the plane. Using theorem 6a.3 we can show that solutions are unique and extend across the whole plane.

On what set does the sign of  $f(x, y) = x - xy$  change? We want to solve  $x - xy = 0$ , that is,  $x(1 - y) = 0$ . We see that in order to satisfy this equation, points have to have  $x = 0$  or  $y = 1$ , so the set of solutions consists of two straight lines. The horizontal line  $y = 1$  and the vertical line  $x = 0$  divide the plane into four regions on which the sign of  $f$  is always constant. What signs are these?

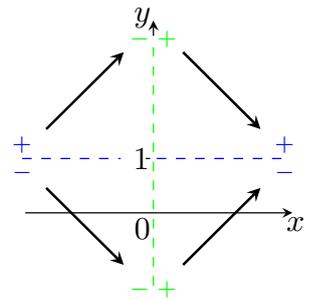
One possibility to answer is to use algebra. For instance, points in the upper-right rectangle satisfy  $x > 0$  and  $y > 1$ , then  $y' = x(1 - y) < 0$ , so solutions in this region are decreasing.

Another popular trick is to argue as follows: Since the sign of  $f$  does not change in the lower-right rectangle, we can simply pick some point from this region and see. I decided to try  $(1, -1)$  and obtain  $f(1, -1) = 2 > 0$ , this extends to the whole region and we see that solutions passing through this region must be increasing there. When we determine trends in all regions, we can express it symbolically as follows.



Such a picture conveys information very effectively. We could say that we “sketched the slope field” of the given equation, we will explain below what this actually means.

There is another way to determine monotonicity and I like to use it in simpler situations, when  $f$  was split into a product (or ratio) of simple factors. When I draw curves for those factors, for each of them I right away determine what signs this particular factor imparts on the regions to the left and right (or up and down, or inside and out), and I mark it in the picture. The for each region I just combine influences of all bordering curves in the usual multiplicative way (two minuses give plus). This can get confusing in more complicate situations, but if you like it, here it is. See also example 8a.b.



By the way, note that  $y = 1$  is not just a dividing line, but also a stationary solution. Sometimes it is useful to notice this, and we will look closer at stationary solutions below.

This picture gives us some idea of what to expect, but note that it leaves some natural questions unanswered. Imagine for instance that we start in the upper right region, which happens with initial condition, say,  $y(1) = 3$ . The corresponding solution then goes down, but we do not quite know what happens next. Could it cross the level  $y = 1$ ? It cannot actually drop **below** this line, because as a continuous function it would have to be decreasing for a while in this lower region, but that is not possible. So in a picture like this we cannot have solutions going from upper rectangles to lower rectangles and vice versa. Could our solution actually touch the line  $y = 1$ ?

In fact, there are equations where such things do happen, see example 6a.a. Our equation does not allow for it, but we have to work a bit to get this conclusion. If our solution came from above and connected with the line  $y = 1$ , then at this connecting point we would actually have two distinct solutions passing through, our  $y$  and the stationary solution. This contradicts uniqueness established above.

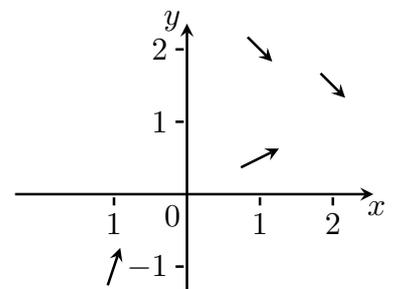
So we know that solutions from the upper and lower half-plane cannot cross or even touch the dividing line, but situation is still not quite clear. Since the solution that started at  $(1, 3)$  cannot reach  $y = 1$ , it must curve right and level off earlier, but how early? Will it converge to 1 or will it level off earlier and converge to some horizontal asymptote  $y(x) = y_a$  with  $y_a > 1$ ? We do not know, since there are cases when exactly this happens.

So there are some interesting questions we cannot answer now, but if this was an equation that we cannot solve, we would be glad for the information that we do have.

△

Can we obtain more information from the equation itself? Note that the equation  $y' = f(x, y)$  provided information not just about the sign of the derivative of solutions, but it also tells us how large the derivative is. What good is it?

Take a point in a plane, for instance  $(1, 2)$ . If we substitute  $x = 1$  and  $y = 2$  into the given equation, we get  $y'(1) = f(1, 2) = -1$ . What does it mean? It means that the solution that passes through the point  $(1, 2)$  has the rate of change equal to  $-1$  there, so this solution should go through that point with slope  $-1$ . We can capture it graphically by drawing a small arrow centred on  $(1, 2)$  going in the right direction.

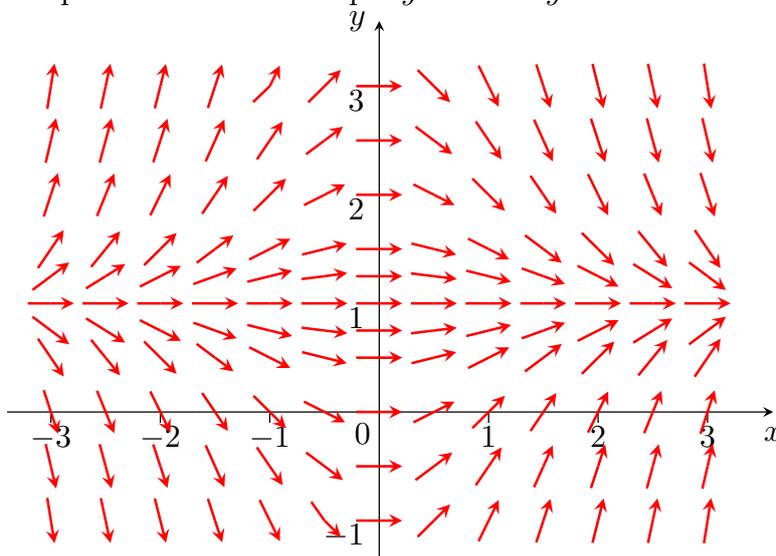


If we look at the point  $(2, 1.5)$ , we get  $y'(2) = -1$  again and add another arrow into our picture. We try a third arrow, the point  $(1, 0.5)$  has solution passing through it with slope  $y'(1) = 0.5$ . Finally, the solution passing through  $(-1, -1)$  does it with the rate of change  $y'(-1) = 3$ .

Things get really interesting if we do this at many points, preferably organized into a regular

mesh. Of course, we ask a computer to do it for us. Note that there are no numerical methods as such involved, we are not trying to approximate anything. The computer simply goes through a given rectangular mesh of points, for each point  $(x, y)$  it directly calculates  $y' = f(x, y)$  and then draws a tiny arrow. Thus there is no error of method, just the usual numerical error in floating point calculations.

We will show one such picture for our example  $y' = x - xy$ .



This is actually a custom job, I added two extra rows of arrows along the stationary solution to get a better feeling for what is going on there. Looking at such a picture we can visualize the paths of various solutions. It seems that their graphs in the upper part are shaped like hills. This visualisation can be very helpful, it is called the **slope field** or **direction field** of the given equation.

**Example 8.b:** This is a bonus example, you may skip it if you want, but I think it's fun.

When I look at the slope field of the equation  $y' = x(1 - y)$ , I have a strong feeling that the shapes are symmetric in two senses, about the  $y$ -axis and about the line  $y = 1$ . Can we confirm it?

To show vertical symmetry we use the following approach. If  $y$  is any function on  $\mathbb{R}$ , its mirror image about the axis  $y = 1$  is given by the formula  $z(x) = 2 - y(x)$ . We will show that if  $y$  is a solution of our equation, then so must be  $z$ . To confirm that  $z$  solves our equation we use the default approach, we choose some  $x$  and substitute  $z(x)$  for the function in both sides.

$$LHS = z'(x) = [2 - y(x)]' = 0 - y'(x) = -(x - xy(x)) = xy(x) - x,$$

$$RHS = x - xz(x) = x - x(2 - y(x)) = -x + xy(x) = xy(x) - x.$$

Since the two sides agree, our equation is satisfied. This means that by flipping a solution that is above the line  $y = 1$  about this line we obtain a solution that is below this line and vice versa. However, this leaves open the possibility that there are also some solutions that cannot be obtained by such flipping, ruining our symmetry. To rule this out we apply a neat trick.

We take a solution, flip it, obtain a solution, and then flip it again. We arrive back at the original solution, and we see that it can indeed be obtained as a flip of another solution, namely of the mirror image of itself. Case closed.

Proving that all solutions are even functions can be done similarly, for a solution  $y(x)$  we consider the function  $z(x) = y(-x)$  and show that it is a solution too, and because these two solutions share the same value at  $x = 0$ , by uniqueness it follows that they must be the same.

We actually managed to learn quite a bit. Given that equations are sometimes very hard to solve, it is nice if we can learn some properties of its solutions just by looking at the equation itself. Actually,  $y' = x(1 - y)$  can be solved on a few lines by separation, but that would defeat

the purpose of this chapter. I swear I did not even think about it until now, so when writing this chapter I was as in the dark regarding this equation as you.

△

Slope field is a useful tool and we ask a computer program to do it for us. Without a computer we can at least “sketch the slope field” as we did in example 8.a.

**Example 8.c:** Consider the equation  $y' = x + y$ . We will sketch its slope field.

The function  $f(x, y) = x + y$  is continuous on  $\mathbb{R}^2$  and the partial derivative  $\frac{\partial f}{\partial y} = 1$  is bounded, so we expect unique solutions at every point. This gives us the starting image of solutions as curves that flow but never touch or cross.

The right hand side cannot be factored as a product of terms, so we have to investigate its sign directly. The dividing curve for the sign is given by  $y' = 0$ , that is,  $x + y = 0$ . This is the equation of a certain line that divides  $\mathbb{R}^2$  into two regions.

In the region above this curve  $y = -x$  we have  $y > -x$ , so  $y' = x + y > 0$ . Below this line the derivative is negative. This determines the sketch of the slope field. We could have also determined the signs by substituting suitable points into  $f(x, y) = x + y$ , or instance  $(1, 1)$  and  $(-1, 1)$ .

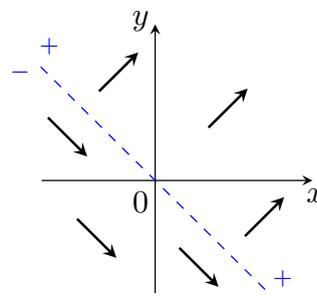
It is not clear how the solutions actually go. If we start in the upper region, solutions will go up and to the right, that seems clear. But what if we start in the lower part? Solutions are supposed to decrease, but if they decrease slowly, they can actually reach the dividing line  $x + y = 0$  (with horizontal tangent line) and then start to increase. Can it happen? We have no way of knowing from the information available to us now.

If we really want to know, we rewrite this equation as  $y' - y = x$  and learn how to solve in chapter 9, because it is not separable.

△

For another useful insight see also example 8a.a

Sometimes we can actually say much more about solutions, but only if the equation is unusually nice and we have to use stronger tools from calculus. We leave it as a bonus section at the end of this chapter for students who like challenge.



## 8a. Equilibria and stability

Some equations allow for a constant solution. This represents the situation when we set up the process described by the equation in some way and it feels good there, so it does not want change. We saw solutions like that in chapter on separable equations, we called them stationary solutions, but the name actually applies also to other types of equations. Even an equation that is not separable may have a stationary (that is, constant) solution.

It happens when there is a value  $y_s$  such that  $f(x, y_s) = 0$  regardless of  $x$ , we easily check that then the constant function  $y(x) = y_s$  solves the differential equation  $y' = f(x, y)$ . For instance, the equation  $y' = (x - y)(y + 1)$  is not separable, but we see that for  $y = -1$  we automatically have  $y' = 0$ . Such stationary solutions are very interesting when analyzing behaviour of various systems and also very simple to find, we simply set  $y' = f(x, y) = 0$  and see whether there is some solution independent of  $x$ .

Going back to our examples in this chapter, the equation  $y' = x - xy$  was found to have the stationary solution  $y = 1$ . On the other hand, the equation  $y' = x + y$  does not have a stationary solution, as there is no special value of  $y$  that would make  $x + y$  always (for all  $x$ ) equal to zero.

We now look at behaviour of solutions from a different viewpoint. Imagine that the given differential equation describes behaviour of some system and  $x$  is time. We fix  $x$  to be some  $x_0$  and

start initiating the system using various starting values  $y_0$ . Mathematically, we consider initial conditions  $y(x_0) = y_0$  for different values  $y_0$ . We want to know how the behaviour of solutions changes depending on the choice of  $y_0$ .

Values  $y_s$  for which  $y' = 0$  for all  $x$  are special, because when we set up the system in this way, it stays the same. As functions  $y(x) = y_s$  they are called stationary solutions, but as simple values used to set up the system they are called equilibria in this context. Note that since we have  $f(x, y_s) = 0$  for all  $x$ , the equilibria that we find are always the same, regardless of what starting time  $x_0$  we use.

**Definition 8a.1.**

Consider a differential equation  $y' = f(x, y)$ . We say that  $y_s \in \mathbb{R}$  is its **equilibrium** if the constant function  $y(x) = y_s$  is a solution of this equation.

Knowing states at which a system is willing to stay is nice, but a much more useful question is what the system does if we move it a bit away from an equilibrium. There are two basic situations. The more popular is a **stable** equilibrium, which means that if we move the system from it by a little bit, the system will return back (eventually). A pendulum is a good example (and by the way, it can be described using a simple differential equation). If left alone it points down (a stationary solution), and if we move it a bit, it eventually settles back down. Even a layperson would call this situation stable.

Then there is another type of equilibrium, namely when we carefully balance the pendulum pointing up. Everyone who ever tried it knows that with the slightest provocation it swings away, never returning back. This would be an unstable situation and engineers, biologists, economists and other applied people are usually less pleased with such situations.

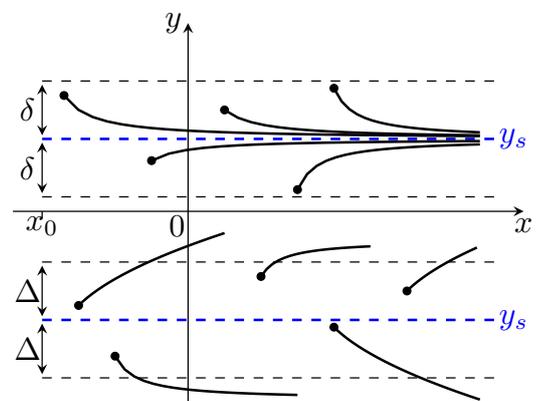
Another good example of a stable equilibrium is a thin band of steel (a leaf spring). When we firmly attach it at one end, the other end stays in place (equilibrium), if we bend the strip a bit, it returns back. However, if we bend it a lot, it stays bent (or breaks). This is something that we have to take into account when formally introducing the notion of stability, namely we cannot allow arbitrary actions when testing stability, just small deviations.

**Definition 8a.2.**

Assume that the differential equation  $y' = f(x, y)$  has an equilibrium  $y_s$ . We say that  $y_s$  is (asymptotically) **stable** if there is  $x_s$  and some  $\delta > 0$  so that for every initial condition  $y(x_0) = y_0$  satisfying  $x_0 > x_s$  and  $|y_0 - y_s| < \delta$ , the corresponding solution  $y(x)$  exists on  $(x_0, \infty)$  and converges to  $y_s$  as  $x \rightarrow \infty$ . We say that  $y_s$  is **unstable** if it is not stable.

Asymptotic stability means that there is a certain strip around the stationary solution  $y(x) = y_s$  (perhaps cut off on the left) and once some solution enters it, it must necessarily converge to  $y_s$  at infinity.

The definition of unstable equilibria is defined as “everything that is not stable” and as such it encompasses all kinds of behaviour, including the possibility that a solution that started a bit away from  $y_s$  keeps oscillating, repeatedly getting close and away from  $y_s$ . Here we show an extreme case, when the stationary solution actually repels those that are close but not equal (if you do not want me then go away). We will pass below to a simpler setting where there will be just two basic types of behaviour, namely those on the right.



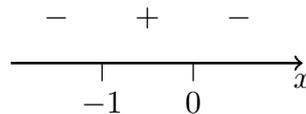
In fact there are several notions that try to capture various aspects of stability, perhaps the most popular is the notion of Ljapunov stability that also applies to solutions that are not constant. However, its analysis is too complicated for this introductory text. We will therefore focus on asymptotic stability that can be often determined rather easily.

So how do we go about it? We use the slope field to make a good guess. For instance, when we look at the slope field in example 8.a, the arrows strongly suggest that if we start with some solution at point  $(x_0, y_0)$  with  $x_0 > 0$ , then this solution will go towards the  $x$ -axis, that is, towards the stationary solution  $y = 0$ . This is actually true in this particular case, but unfortunately, things are not always so simple.

**Example 8a.a:** Consider the differential equation  $y' = \frac{-1}{x(x+1)}y$ . We immediately see the conditions  $x \neq 0$  and  $x \neq -1$ , so there will be three families of solutions. We solve the equation  $\frac{-1}{x(x+1)}y = 0$  and obtain  $y = 0$ . Thus the  $x$ -axis is a dividing line, and  $y(x) = 0$  is a stationary solution. This means that  $y_s = 0$  is an equilibrium.

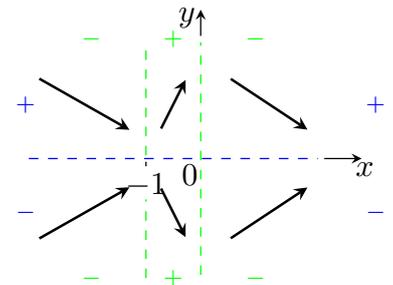
We also notice that  $f$  does not exist when  $x = 0, -1$ , so these two vertical lines can also be a place where  $f$  changes signs. Thus we obtain six regions. Now we want to determine signs in each of them. Note that the variables are separated in  $f$ , so the sign at various regions can be recovered from the influence of  $y$  and the influence of  $x$  in  $f$ . The influence of  $y$  is fairly obvious, it imparts plus sign on the top half-plane and negative sign below the  $x$ -axis.

To see the influence of the  $x$ -component  $\frac{-1}{x(x+1)}$  we have to investigate the sign of  $x(x+1)$  as it depends on  $x$ . This is a standard task from introductory calculus course (for instance when investigating monotonicity), and we apply the popular algorithm. We find dividing points ( $x = 0, -1$ ), split the real line accordingly and determine signs in all three regions, not forgetting the  $-1$  in the numerator.



Now we put all this information together. We sketch the three dividing lines and mark on the outside the signs in various horizontal and vertical strips as determined above.

For each region we then determine the sign of  $y'$  by combining the contribution from the  $y$ -component and the  $x$ -component, and mark it with a suitable arrow. In this way we obtain the slope field.



When it comes to stability, we are interested in behaviour when  $x \rightarrow \infty$ , so we look at the part to the right from  $x = 0$ . The arrows go toward the  $x$ -axis, suggesting that  $y_s = 0$  is a stable equilibrium. However, this is not the case.

The given equation can be solved using separation as follows:

Assuming that  $y \neq 0$  (and noting that it yields the stationary solution  $y(x) = 0$ ) we rearrange the equation:

$$\frac{y'}{y} = \frac{-1}{x(x+1)} = \frac{1}{x+1} - \frac{1}{x}.$$

We integrate and use the standard trick with exponential:

$$\begin{aligned} \ln |y| &= \ln |x+1| - \ln |x| + C = \ln \left| \frac{x+1}{x} \right| + C \implies |y| = e^C \left| \frac{x+1}{x} \right| \\ \implies y &= \pm e^C \frac{x+1}{x} \implies y(x) = C \frac{x+1}{x}. \end{aligned}$$

Right now  $C$  is not allowed to be zero, but we note that choosing  $C = 0$  we obtain the stationary solution.

Conclusion: The general solution is

$$y(x) = C \frac{x+1}{x} = C \left(1 + \frac{1}{x}\right), \quad x \neq -1, 0.$$

It is easy to see that these solutions do not tend to 0 as  $x \rightarrow \infty$ , so  $y(x) = 0$  is not a stable stationary solution, that is,  $y_s = 0$  is not a stable equilibrium.

Indeed, if we take, for instance, the solution passing through  $(1, 1)$ , we easily determine that this solution comes by the choice  $C = \frac{1}{2}$ . The function  $y(x) = \frac{1}{2} \left(1 + \frac{1}{x}\right)$  is indeed decreasing on  $(0, \infty)$ , but converges to  $\frac{1}{2}$  at infinity.

△

This shows that our attempt to learn about solutions just from the equation itself has its limitations. If we want to achieve more, we have to restrict our attention to a special class of equations.

**Definition 8a.3.**

A differential equation is called **autonomous** if the free variable does not appear in it, that is, if it can be written as  $y^{(n)} = f(y, y', \dots, y^{(n-1)})$ .

This looks like a very restrictive requirement—and it is, but in fact many (most?) useful equations satisfy this condition. Why is it so? Imagine that  $x$  stands for time. The value of  $x$  depends on how we measure it. If we start an experiment now, will its outcome depend on whether we forgot to set our clock to the cursed summer savings time? Of course it will not. When I drop a stone, I do not expect the outcome to depend on whether it's Tuesday or Wednesday. So it is in fact natural that when we describe some real-life phenomenon using a differential equation, we obtain an autonomous one. That said, some experiments do require time keeping, for instance if we want to influence it a set time after starting it, so we will definitely also work with non-autonomous equations in this book, but not now.

Note that a first-order autonomous equation can be written as  $y' = h(y)$ , in particular it is separable and there are no restrictions on  $x$ . Equilibria are exactly the roots of  $h$ . For such equations, determining stability is a fairly straightforward procedure, as the sign of  $y'$  now depends only on  $h(y)$ , that is, it is influenced only by location of  $y$ . The plane thus splits into horizontal strips. Moreover, now there are just three distinct types of behaviour.

Indeed, take any continuous function  $h(y)$ . Then its domain splits into open intervals so that on each of them  $h$  has a certain sign. Consider some endpoint  $y_s$  of one such interval. Then  $h$  must be defined on some interval to the left of  $y_s$  or to the right of  $y_s$  (or both), and have a specific sign on each of these two intervals. We look at the case when  $h$  is defined on both sides and realize that there are in fact three cases possible.

One is the situation as in the above example, when  $h$  is decreasing above  $y_s$  and increasing below  $y_s$ . Then solutions are attracted to  $y_s$ , and for autonomous equations it is not possible for them to level off too early like in the example above. They have to go all the way to  $y_s$  and we get a stable equilibrium.

The second distinct example is that the arrows point away from the stationary solution, and then solutions are repelled from these regions. This is the typical **unstable** equilibrium.

Finally, it can happen that the arrows above and below a stationary solution go in the same direction, both up or both down. Then in one region solutions behave as if this equilibrium were stable, and in the other it's the opposite behaviour. By the general definition above this would also be an unstable equilibrium, and for many people this is enough. However, some people actually like to call these

semi-stable equilibria when dealing with autonomous equations.

We will confirm these observations formally.

**Theorem 8a.4.**

Let  $y' = h(y)$  be an autonomous differential equation with  $h$  continuous, and let  $y_s$  be its equilibrium.

(i) If there is  $\delta > 0$  such that  $h(y) < 0$  for all  $y \in (y_s, y_s + \delta)$  and  $h(y) > 0$  for all  $y \in (y_s - \delta, y_s)$ , then  $y_s$  is a stable equilibrium.

(ii) If there is  $\delta > 0$  such that  $h(y) > 0$  for all  $y \in (y_s, y_s + \delta)$ , then  $y_s$  is an unstable equilibrium.

Moreover, there is  $\Delta > 0$  such that for every initial condition  $y(x_0) = y_0$  with  $y_0 \in (y_s, y_s + \Delta)$ , the corresponding solution  $y(x)$  has some  $x_f$  so that  $y(x) \geq y_s + \Delta$  for  $x \geq x_f$ .

(iii) If there is  $\delta > 0$  such that  $h(y) < 0$  for all  $y \in (y_s - \delta, y_s)$ , then  $y_s$  is an unstable equilibrium.

Moreover, there is  $\Delta > 0$  such that for every initial condition  $y(x_0) = y_0$  with  $y_0 \in (y_s - \Delta, y_s)$ , the corresponding solution  $y(x)$  has some  $x_f$  so that  $y(x) \leq y_s - \Delta$  for  $x \geq x_f$ .

**Proof:** (i): Consider some solution  $y(x)$  passing through some  $(x_0, y_0)$  with  $y_0 \in (y_s - \delta, y_s)$ , so it appeared in the  $\delta$ -strip below  $y_s$  where we know  $h < 0$ . By our assumption this solution then must be increasing at  $x_0$  and it has to stay increasing as long as it stays in this strip. Can it actually leave it?

First note that while in the strip, the function can never get smaller than  $y_0$ . Assume the contrary, that is, that  $y(x_f) < y_0$  for some  $x_f > x_0$ . We can arrange it (by taking some  $x_f$  closer to  $x_0$  if necessary) that  $y$  actually stays in the  $\delta$ -strip below  $y_s$  on  $[x_0, x_f]$ . Applying the Mean Value Theorem to the interval  $[x_0, x_f]$  we find that there would have to be some  $\xi \in (x_0, x_f)$  so that  $y'(\xi) < 0$ , but that would contradict our assumption on  $h = y'$ .

Thus the only possibility for  $y$  to leave this strip is up, but for that it would have to first cross the level  $y_s$ . We claim that if this solution  $y$  actually attains the value  $y_s$  at some  $x_f$ , then it necessarily has to join the stationary solution  $y(x) = y_s$  from then on. The argument is similar: We show that if  $y$  left this level downwards, then it would have to have a negative derivative somewhere in the  $\delta$ -strip below  $y_s$ , contradicting our assumption on  $h$ , and if  $y$  went above the level  $y_s$ , then it would have to have a positive derivative somewhere in the  $\delta$ -strip above  $y_s$ , another contradiction.

This confirms our claim that if there is some  $x_f$  so that  $y(x_f) = y_s$ , then  $y(x) = y_s$  for  $x \geq x_f$ . This in particular means that  $y(x) \rightarrow y_s$  as  $x \rightarrow \infty$ , so stability is confirmed for this case.

We are left with one possibility to explore: That  $y(x)$  stays within the  $\delta$ -strip below  $y_s$  but never reaches this value. Then it is always increasing on  $[x_0, \infty)$  and we have to rule out the possibility that it would level off too soon.

If  $y(x) < y_s$  for all  $x \geq x_0$ , then  $y$  is an increasing function bounded from above by  $y_s$  and as such it must have some limit  $y_f \geq y_s$ . We claim that  $y_f = y_s$ .

Indeed, assume that  $y_f < y_s$ . We also have  $y_f \geq y_0$ , so we focus on the interval  $[y_f, y_0]$ . It is a subset of  $(y_0 - \delta, y_0)$ , so  $h > 0$  there, and the interval is closed, so  $h$  as a continuous function must have its minimum there. Thus there must be some  $m > 0$  such that  $h(y) \geq m$  whenever  $y \in [y_f, y_0]$ . By our differential equation it follows that  $y'(x) \geq m$  for  $x \geq x_0$ , that is,  $y$  must increase at a certain guaranteed rate, which seems to contradict the idea that it levels off to have limit equal to  $y_f$  at infinity. We need to confirm this mathematically.

Take any  $x > x_0$ . Applying the mean value theorem to the interval  $[x_0, x]$  we obtain

$$\frac{y(x) - y(x_0)}{x - x_0} = y'(\xi) \geq m,$$

hence  $y(x) \geq y(x_0) + m(x - x_0)$ . The expression on the right tends to infinity as  $x \rightarrow \infty$ , so by comparison also  $y \rightarrow \infty$ , which contradicts assumption that  $y \rightarrow y_f$ . This completes the proof that  $y_f = y_s$ . We confirmed that  $y(x) \rightarrow y_s$  as needed for stability.

The proof for the case when  $y_0 \in (y_s, y_s + \delta)$  is analogous.

(ii): We can choose  $\Delta$  to be any positive number smaller than  $\delta$ .

Indeed, consider some solution  $y(x)$  passing through a point  $(x_0, y_0)$  satisfying  $y_0 \in (y_s, y_s + \Delta)$ . As in (i), we find that  $h(y) \geq m$  on  $[y_0, y_s + \Delta]$  for some positive  $m$ , and conclude that as long as  $y$  stays within the  $\Delta$ -strip, it has to satisfy the lower estimate

$$y(x) \geq y(y_0) + m(x - x_0).$$

However, this very estimate forces it to leave the  $\Delta$  strip sooner or later, that is, there must be  $x_f$  so that  $y(x_f) \geq y_s + \Delta$ . Then using the Mean Value Theorem we show that it is not possible to have  $y(x) < y_s + \Delta$  for some  $x > x_f$  because it would contradict our assumption about  $h > 0$ , just like we did it in (i).

(iii): The proof is analogous to that for (ii). □

For a typical function  $h$  we obtain several zero points (equilibria) and between them we have regions of positive or negative signs. Each such region represents a strip in the plane that sends all solutions it captures away from one edge, towards the other edge. This shows that indeed, there are only two modes of behaviour for stability for a stationary solution, solutions are either attracted or pushed away on each side of it.

**Example 8a.b:** Consider the equation  $y' = y^2(1 - y^2)$ . We want to analyze the behaviour of its solutions.

We note that the right-hand side is continuous with a continuous derivative on  $\mathbb{R}^2$ , so we have unique solutions passing through all points in the plane.

Setting  $y' = 0$  we see that there are three stationary solutions, namely  $y(x) = 1$ ,  $y(x) = 0$ , and  $y(x) = -1$ . Since this equation is autonomous, the numbers 1, 0, -1 are actually equilibria.

Now we will sketch the slope field. We are interested in signs of  $y^2(1 - y^2) = y^2(1 - y)(1 + y)$ . Since having three vertical dividing lines in the plane would make the final analysis of signs tricky, it is better to first determine the sign of  $h$  as a one-dimensional problem. This is in fact a standard question from introductory calculus and we have a procedure for it. We obtain

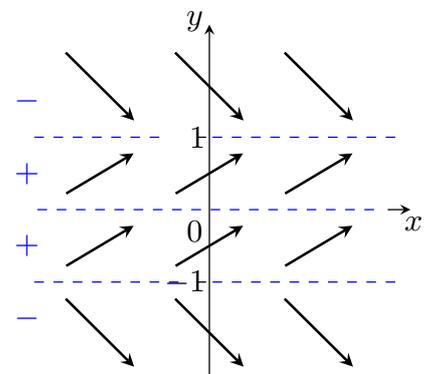
$y:$	$(-\infty, -1)$	$(-1, 0)$	$(0, 1)$	$(1, \infty)$
$y' = y^2(1 - y^2):$	-	+	+	-
$y(x):$	↘	↗	↗	↘

We could actually do with a much simpler picture:

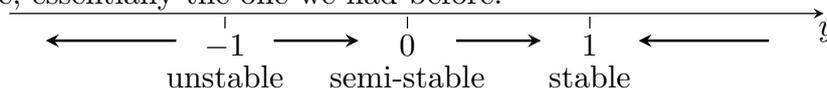
$$\overbrace{-1 \quad 0 \quad 1}^{\text{---} y \text{---}}$$

When we turn it sideways, it will show signs for the corresponding horizontal strips in our sketch of the slope field.

Using Theorem 8a.4, we conclude that  $y = 1$  is a stable equilibrium, while  $y = 0$  and  $y = -1$  are unstable; some may classify  $y = 0$  as semi-stable.



Note that we do not really need the  $x$ -axis in this picture in order to make conclusions. For an autonomous equation, the key information is  $y_0$ , the value where we start our chosen solution, and from that it already follows which way the chosen solution goes. This information can be captured in a simpler picture, essentially the one we had before.



Such a diagram is called a **phase diagram**. In this case we can also call it the **phase line**. It is usually used when analysis a system of two differential equations with two unknown functions, then it is two-dimensional and we will return to it later.

Sometimes people classify stability of equilibria in a different way. In the theorem above we related stability to signs of  $h(y)$ , and the information about signs can be captured locally using derivative. For instance, at a stable equilibria the sign of  $h$  must go from positive to negative when viewed in the natural way, going left to right. This means that  $h$  must be decreasing there. Given that  $h = 0$  at the point of equilibria, this kind of reasoning works also the other way.

**Fact 8a.5.**

Consider an autonomous ordinary differential equation  $y' = h(y)$ . Assume that for some  $y_0$  we have  $h(y_0) = 0$  and  $h$  is differentiable at  $y_0$ .

(i) If  $h'(y_0) < 0$ , then  $y_0$  is a stable equilibrium for the given equation.

(ii) If  $h'(y_0) > 0$ , then  $y_0$  is an unstable equilibrium for the given equation.

Semi-stable equilibria cannot be characterized in this way. There are people who like to using this fact, but I prefer not to remember it and do my analysis of signs as above. I have two reasons. First, I gain more information and get a better feeling for how the solutions behave if I think about regions where solutions increase and where they decrease. Second, unlike the Fact, this way will provide answers also in cases when  $h$  does not have a derivative.

## 8b. Bifurcations

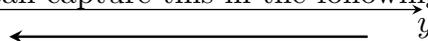
Imagine that a differential equation describes the workings of some system (mechanical or natural) that we can influence by setting it up in a certain way. Mathematically, there is some parameter in the equation that we can set as we want. An interesting question is how does the behaviour of the system depends on the setup, that is, on the value of the parameter.

Looking at equilibria and their stability may provide useful information in this regard.

**Example 8b.a:** Consider the autonomous differential equation  $y' = p - (y - 1)^2$ , where  $p$  is a real parameter. What can we say about equilibria and stability?

We start by asking about equilibria, that is, by solving  $p - (y - 1)^2 = 0$ . We see that there are three possibilities.

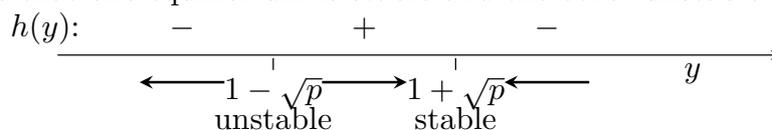
1) If  $p < 0$ , then there is no equilibrium at all. Moreover,  $y' < 0$ , so all solutions are decreasing regardless of initial value  $y_0$ . We can capture this in the following phase diagram.



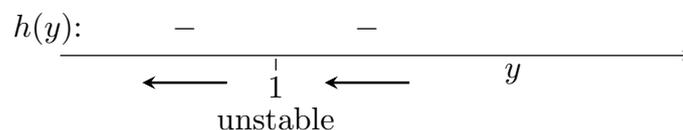
2) If  $p > 0$ , then there are two roots,  $y_s = 1 \pm \sqrt{p}$ . These are equilibria. We write the equation as

$$y' = (\sqrt{p} - (y - 1))(\sqrt{p} + (y - 1)) = -(1 + \sqrt{p} - y)(1 - \sqrt{p} - y),$$

analyze signs and see that one equilibrium is stable and the other unstable.

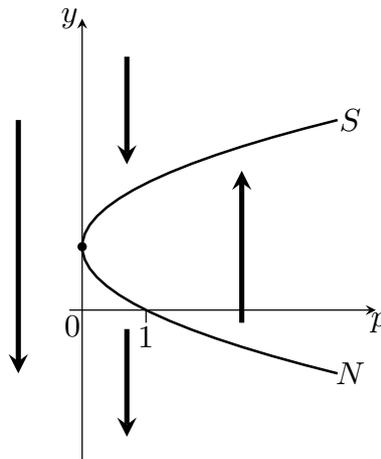
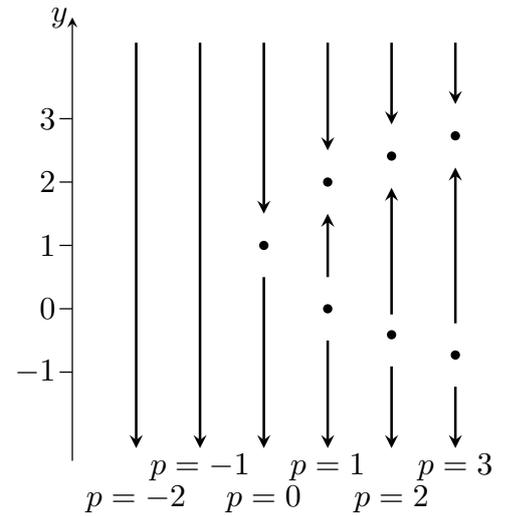


3) If  $p = 0$ , then there is one equilibrium  $y_s = 1$ . Since  $y' = -(y - 1)^2$ , we obtain the following diagram.



We can imagine that there is some lever controlling the value of  $p$  and as we move this lever, at certain point ( $p = 0$ ) the behaviour of the system changes abruptly. Such a point is called a **bifurcation point**.

Now imagine that we prepare many phase diagrams as above, for various values of  $p$ , flip them vertically so that the  $y$ -axis points naturally up and make a chain of them. Then imagine that we have more and more of these phase lines there, eventually they would merge and create a two-dimensional picture where the equilibria would appear as curves, the horizontal axis would show the value of  $p$ . Traditionally, the curves for stable and unstable equilibria are denoted  $S$  and  $N$ .



In this way we arrive at **bifurcation diagram** and in some applications this is a very useful tool. If you want to know how such a system behaves for a certain setup  $p$ , you just slice the picture vertically and see the appropriate phase line.

As you can see, one can learn quite a lot about what is happening without actually solving the given differential equation.

## 9. First order linear ODEs (variation)

Not every first order differential equation can be separated. There is another type that we can handle quite well.

### Definition 9.1.

By a **linear ODE of order 1** we mean any ODE of the form  $y' + a(x)y = b(x)$ , where  $a, b$  are some functions.

This equation is called **homogeneous** if  $b(x) = 0$ .

For instance, the equation  $y' + 2xy = 4x$  is linear, we have  $a(x) = 2x$  and  $b(x) = 4x$ . By the way, note that it is not separable.

It should be noted that later we will look at linear equations of order  $n$ , so this chapter may seem a bit useless. There are two reasons for looking at the special case of first order equations. First, for such equations we can do more. Second, we will get used here to key concepts.

These concepts are related to the word “linear”. There are many different kinds of equations (algebraic, differential, recursive, ...), and those that can be called linear all share common features that make it easier to solve them.

We assume that the reader is familiar with systems of linear algebraic equations that can be written as  $A\vec{x} = \vec{b}$ , where  $A$  is a matrix. We will draw on our experience with such systems.

One general feature is that homogeneous equations are easier to solve.

### Definition 9.2.

Consider a first order linear ODE  $y' + a(x)y = b(x)$ . By its **associated homogeneous equation** we mean the equation  $y' + a(x)y = 0$ .

**Example 9.a:** Consider the equation  $y' + 2xy = 4x$ .

We solve the associated homogeneous equation  $y' + 2xy = 0$ . We can do it by separation.

$$\begin{aligned}y' + 2xy = 0 &\implies y' = -2xy \implies \int \frac{dy}{y} = \int -2x dx \\ &\implies \ln |y| = -x^2 + C \implies |y| = e^{C-x^2} = e^C e^{-x^2}.\end{aligned}$$

Now we use the recommended approach to absolute value,

$$y_h(x) = \pm e^C e^{-x^2} = D e^{-x^2}.$$

△

In general, we can always go from  $y' + a(x)y = 0$  to  $\int \frac{dy}{y} = \int -a(x) dx$ , so we can solve the homogeneous equation as long as we can integrate  $a(x)$ . Theoretically it would be enough to ask for continuity of  $a(x)$ , practically we have to get a bit lucky to actually handle the integration. If we choose an antiderivative of  $a(x)$  and call it  $A(x)$ , then we get

$$\ln |y| = -A(x) + C \implies y = \pm e^C e^{-A(x)}.$$

Denoting  $D = \pm e^C$  and incorporating the stationary solution  $y(x) = 0$  by allowing  $D = 0$  we obtain the following conclusion: The formula  $y_h(x) = D e^{-A(x)}$  is a general solution of  $y' + a(x)y = 0$ . So indeed, the homogeneous case is nice.

We remark that for linear differential equations of higher order we do not have separation available, and thus we will have to solve them in a different way that only works when  $a(x)$  is a constant—a significant loss of generality.

Anyway, we solved  $y' + 2xy = 0$  which is nice, but we actually wanted to solve something else. Is there anything  $y_h(x) = D e^{-x^2}$  is good for? There is a curious idea that goes as follows. If we

allow  $D$  to change, to become a function  $D(x)$ , we may get lucky and with a smart choice of  $D$  obtain a solution to the given equation (the original one). Now there is no obvious reason why this should work, but in fact it does. How do we find the right function  $D(x)$ ? We substitute the general formula  $D(x)e^{-x^2}$  into the given equation and see what works.

**Example 9.b:** We return to the equation  $y' + 2xy = 4x$ . In example 9.a we found the general solution  $y_h(x) = D e^{-x^2}$  of the associated homogeneous equation.

Now we make a wild guess and look for a solution of the form  $y(x) = D(x)e^{-x^2}$ . We substitute this into the given equation:

$$\begin{aligned} [D(x)e^{-x^2}]' + 2xD(x)e^{-x^2} &= 4x \\ [D(x)'e^{-x^2} + D(x)(-2x)e^{-x^2}] + 2xD(x)e^{-x^2} &= 4x \\ D(x)'e^{-x^2} &= 4x \end{aligned}$$

Note that we were incredibly lucky and terms with  $D(x)$  disappeared. This miracle of cancelling is the heart of this procedure, since without it we would face a differential equation featuring both  $D(x)$  and  $D'(x)$ . As it is, we only have  $D'(x)$  that we easily isolate and then pass to  $D(x)$  using everyone's favourite, integration.

$$\begin{aligned} D(x)' &= 4x e^{x^2} \implies \\ D(x) &= \int 4x e^{x^2} dx = 2e^{x^2} + C. \end{aligned}$$

We used substitution  $z = x^2$  for the integration.

We substitute this  $D(x)$  into the formula for  $y$  and obtain

$$y(x) = (2e^{x^2} + C)e^{-x^2} = 2 + C e^{-x^2}.$$

What did we obtain? It is easy to check that this is a solution of the given equation. It features one parameter, so it is a general solution, exactly what we wanted.

Finally, we inquire about the region of validity. There is no restriction on  $x$  or  $y$  in the equation and no restriction on  $x$  in the formula, so we got for the largest possible set.

Conclusion: The given equation has general solution

$$y(x) = 2 + C e^{-x^2}, \quad x \in \mathbb{R}.$$

△

Now were we just incredibly lucky, or is there more to this idea? Let's explore it in general. We already commented above that if we can find an antiderivative  $A(x)$  to  $a(x)$  on some interval  $I$ , then  $y_h(x) = D e^{-A(x)}$  is a general solution of  $y' + a(x)y = 0$  on  $I$ .

Note that it is not always necessary to solve the homogeneous version using separation. When  $a(x)$  is in fact a constant, we will be able to use a much simpler trick that we will learn in chapter 15, or we simply recognize that then we in fact have a case of exponential growth on our hands and write the solution right away (see 10a.1).

Now we try the trick with function instead of a constant. We will therefore ask whether we can find a solution of the form  $y(x) = D(x)e^{-A(x)}$ . We substitute this for  $y$  in the general equation  $y' + a(x)y = b(x)$ :

$$\begin{aligned} [D(x)e^{-A(x)}]' + a(x)D(x)e^{-A(x)} &= b(x) \\ \implies D'(x)e^{-A(x)} + D(x)e^{-A(x)}[-A(x)]' + a(x)D(x)e^{A(x)} &= b(x) \\ \implies D'(x)e^{-A(x)} - D(x)e^{-A(x)}a(x) + a(x)D(x)e^{A(x)} &= b(x) \\ \implies D'(x)e^{-A(x)} &= b(x). \end{aligned}$$

We just saw that for linear equations the miracle of cancelling always happens (is it a miracle

then?) and we are ready to proceed:

$$D'(x) = b(x)e^{A(x)} \implies D(x) = \int b(x)e^{A(x)} dx.$$

We can check that this agrees with the corresponding integral in the above example if we put  $a(x) = 2x$  and  $A(x) = x^2$ .

Now we have to make yet another assumption, namely that this new integral can be evaluated. Since  $e^{A(x)}$  is continuous on  $I$ , it is enough to ask for continuity of  $b(x)$  on  $I$ . Then we can (theoretically) find some antiderivative  $B(x)$  of  $b(x)e^{A(x)}$  on  $I$  and obtain  $D(x) = B(x) + C$ . This function will therefore make our guess into a solution on  $I$ , namely this one:

$$y(x) = D(x)e^{-A(x)} = (B(x) + C)e^{-A(x)}.$$

We just confirmed that the wild idea from our example actually works as long as our equation is linear of order 1 and the functions  $a(x), b(x)$  are continuous on some interval  $I$ . This approach is called **variation of parameter**, because we let the parameter  $D$  vary (change). We have just shown that it works, and incidentally proved the following statement.

**Theorem 9.3.** (on solution of linear ODE of order 1)

Consider a linear ODE  $y' + a(x)y = b(x)$ . Assume that  $a(x), b(x)$  are continuous functions on an open interval  $I$ , let  $A$  be some antiderivative of  $a$  on  $I$ . Then the given equation has a solution on  $I$  of the form  $\left(\int b(x)e^{A(x)} dx\right)e^{-A(x)}$ .

If  $B$  is some antiderivative of  $b(x)e^{A(x)}$  on  $I$ , then a general solution of the given equation on  $I$  is

$$y(x) = (B(x) + C)e^{-A(x)}.$$

Most people actually do not remember these formulas, instead they remember the process as we described above. It has two crucial steps (solve the homogeneous version, then change constant into a function) that are definitely easier to remember than some mathematical formulas, and as a bonus they are applicable also for other types of linear equations, as we will see later.

We usually codify procedures like this with a formal algorithm, but we prefer to make it a bit more general. To appreciate the ideas we first explore some interesting connections with linear algebra.

We start by noting that the solution that we obtained in example 9.b can be written like this:

$$y(x) = 2 + Ce^{-x^2} = 2 + y_h(x)$$

and we easily confirm that  $y_p(x) = 2$  solves the original equation. This should remind the reader of a crucial structural theorem from linear algebra:

- A general solution of the system  $A\vec{x} = \vec{b}$  can be obtained as  $\vec{x} = \vec{x}_p + \vec{x}_h$ , where  $\vec{x}_h$  is a general solution of the associated homogeneous equation and  $\vec{x}_p$  is some particular solution of the given system.

It seems that the same principle works also here. Indeed, we will prove an analogous general theorem for linear differential equations in chapter 16. It has one practical impact, we can choose which way to go in our variation. One way is the one that we followed above, arriving directly at a general solution. The other possibility is not to add integration constant when deriving formula for  $D(x)$  (I know that you always have to put “+C”, here you have a special permission to do without it), then you obtain a particular solution  $y_p$  and get to a general solution using the formula  $y = y_p + y_h$ .

There is another structural theorem that linear algebra knows for systems of linear equations.

- The set of all solutions of a homogeneous system  $A\vec{x} = \vec{0}$  is a vector space.

How does it work for us? We do not have any vectors here. Obviously we need a little sidetrip

### 9.4 Remark vector spaces of functions:

A vector space is a set of objects of just any kind, as long as we have some way to “add” them and “scale” them and some rules are satisfied. For a “vector” we thus can take lots of things, like numbers,  $n$ -tuples of numbers (this is what we commonly see as vectors), matrices, and also functions.

To make this work we first fix some non-degenerate interval  $I$ , once for all. We start by considering the set of all functions on this interval, let's call this set  $V$ :

$$V = \{f : \text{a real function such that } I \subset D(f)\}.$$

We can add functions, and when we add two functions defined on  $I$ , the result is also defined on  $I$ . Similarly, we can multiply functions on  $I$  by real numbers, obtaining again functions on  $I$ . We therefore have operations of addition and scalar multiplication on the set  $V$ .

In linear algebra we learned that being a vector space requires more: some rules must be true, traditionally eight is listed (commutativity, distributive laws and such). Here it is easy to show that they work, because addition and multiplication of functions is based on operations with real numbers.

Thus we have ourselves a nice vector space where vectors are functions. Note that operations and comparisons work globally. To explain what we mean, let's ask about linear independence. What does it mean for two functions  $f, g$  to be linearly independent? Linear algebra has the answer. We have to ask whether it is possible to obtain zero as a non-trivial linear combination of  $g, f$ , that is, whether there are  $\alpha, \beta$  not both being zero such that  $\alpha f + \beta g = 0$ . In this equality,  $f$  and  $g$  are functions as objects, and  $0$  is actually the zero function on  $I$ , and the equality means that for all  $x \in I$  we should have  $\alpha f(x) + \beta g(x) = 0$ .

For instance, polynomials  $1$  and  $x$  are linearly independent, because we cannot find  $\alpha, \beta \neq 0$  such that  $\alpha \cdot 1 + \beta \cdot x = 0$  on some interval  $I$ . Indeed, looking at it another way, it is impossible to obtain the function  $x$  just by multiplying the function  $1$  by some fixed constant. We also know that we cannot obtain  $x^2$  as a polynomial  $\alpha x + \beta 1$  and so on, so it seems that  $\{1, x, x^2, x^3, x^4, \dots\}$  forms an infinite linearly independent set in  $V$ . Consequently, the dimension of  $V$  is infinite. Don't ask us how the basis looks like, that's a tough one and fortunately we do not need to know.

In applications we usually make do with “smaller” spaces. One such popular space is

$$C(I) = \{f : \text{a function continuous on } I\}.$$

Again, we will draw on our knowledge of linear algebra here. We do not need to show all the properties again in order to prove that  $C(I)$  with the usual operations is a vector space. Rather, we observe that it is a subspace of  $V$ , so by a theorem from linear algebra, it becomes a subspace of  $V$  (and therefore a vector space of its own) once we prove that this set is closed under operations. Namely, we have to show the following:

- If  $f, g \in C(I)$ , then  $f + g \in C(I)$ ,
- if  $f \in C(I)$  and  $\alpha \in \mathbb{R}$ , then  $\alpha f \in C(I)$ .

In other words, we have to show that a sum of two functions continuous on  $I$  is a function continuous on  $I$ , similarly for the multiple. This is all true, this time we call on suitable theorems from introductory calculus.

Note that sometimes people combine the two conditions into one, they would show that  $\alpha f + g \in C(I)$ . It is a matter of personal preference whether one does two simpler proofs or one slightly longer.

Thus we have ourselves a nice vector space of functions. There are many “function spaces” that mathematicians use, one type is of interest:

$$C^n(I) = \{f : f \text{ is } n\text{-times differentiable on } I \text{ and } f^{(n)} \text{ is continuous on } I\}.$$

This makes sense for open intervals, for other types of intervals we just require derivatives on the interior of  $I$ .

In this chapter we are looking for solutions of  $y' + a(x)y = b(x)$ , so obviously  $y$  must be differentiable, hence continuous. We can also assume that  $a(x), b(x)$  are continuous, therefore the derivative  $y'$  itself should be continuous. That is, when solving such a differential equation on interval  $I$ , we are in fact searching among functions from  $C^1(I)$ .

Finally, it should be noted that in some applications we need to work with complex functions, then we also use complex scalars for multiplication. Everything works as outlined above, we get vector spaces.

△

Back to our main topic. We observed that given an equation  $y' + a(x)y = 0$ , we can find its general solution in the form  $y_h(x) = D e^{-A(x)}$ . Denote  $u(x) = e^{-A(x)}$ . We easily check that it is a solution of the homogeneous equation, and all other solutions are its multiples. This shows that the set of all solutions of the homogeneous equation  $y' + a(x)y = 0$  is a vector space of dimension 1, with basis  $\{u(x)\}$ . Again, in chapter we will show that an analogous statement is true for all homogeneous linear differential equations.

This observation allows us to take a more general view of variation and state the algorithm in a way that will be useful later on, when we will adapt it also for other types of linear equations.

### Algorithm 9.5.

⟨variation of parameter for linear ODE of order 1⟩

Given: equation  $y' + a(x)y = b(x)$ .

1. Using separation or another approach, find a general solution  $y_h$  of the associated homogeneous equation  $y' + a(x)y = 0$ .

It has the form  $y_h(x) = C \cdot u(x)$ , which includes also stationary solutions.

2. Variation of parameter: Seek a solution of the form  $y(x) = C(x) \cdot u(x)$ .

Either substitute this  $y(x)$  into the given equation  $y' + a(x)y = b(x)$  and cancel, or remember that it leads to the equation  $C'(x)u(x) = b(x)$ . Then  $C(x) = \int \frac{b(x)}{u(x)} dx$

3. Substitute  $C(x)$  into  $y(x) = C(x)u(x)$ .

If you include “+C” when deriving  $C(x)$  by integration, then after substituting it into  $y(x) = C(x)u(x)$  you get the general solution.

If you take for  $C(x)$  one particular antiderivative, then you get one particular solution  $y_p(x)$ , the general solution is then  $y = y_p + y_h$ .

△

In this algorithm, we no longer work with a specific form of the homogeneous solution  $u(x)$ . Will the miracle of cancelling still happen? When we substitute  $C(x)u(x)$  into the given equation, we obtain

$$\begin{aligned} [C(x)u(x)]' + a(x)C(x)u(x) &= b(x) \\ \implies C'(x)u(x) + C(x)u'(x) + a(x)C(x)u(x) &= b(x) \\ \implies C'(x)u(x) + C(x)[u'(x) + a(x)u(x)] &= b(x). \end{aligned}$$

Since  $u(x)$  is a solution of the associated homogeneous equation, the expression  $u'(x) + a(x)u(x)$  must be zero and we obtain  $C'(x)u(x) = b(x)$ . This is a very important observation. It shows that the substance of variation is in working with a solution of the associated homogeneous equation, not in its special form  $e^{-A(x)}$ .

And that's the whole story. If the reader wants to practice understanding of the linear algebra treatment of functions, we offer a proof of the statement about solutions of homogeneous equation. A general version appears in chapter 15.

**Fact 9.6.**

Consider a homogeneous linear differential equation  $y' + a(x)y = 0$ . The set  $W$  of all its solutions on some interval  $I$  is a vector space.

**Proof:** We will show that  $W$  is a subspace of  $V$ . We need to prove closedness under operations: If  $y_1, y_2 \in W$  and  $\alpha$  in  $\mathbb{R}$ , then  $\alpha y_1 + y_2 \in W$ .

Recalling the meaning of  $W$  we translate it as follows: For any two solutions  $y_1$  and  $y_2$  of the given equation on some interval  $I$  and all  $\alpha \in \mathbb{R}$ , the function  $y(x) = \alpha y_1(x) + y_2(x)$  is also a solution to that equation on  $I$ .

So assume that  $y_1, y_2$  solve  $y' + a(x)y = 0$  on  $I$  and that  $\alpha \in \mathbb{R}$ . We confirm that  $\alpha y_1 + y_2$  solves it as well by substituting into the left hand side and arriving at zero. We take any  $x \in I$  and evaluate.

$$\begin{aligned} [\alpha y_1(x) + y_2(x)]' + a(x)(\alpha y_1(x) + y_2(x)) &= \alpha y_1'(x) + y_2'(x) + a(x)\alpha y_1(x) + a(x)y_2(x) \\ &= \alpha[y_1'(x) + a(x)y_1(x)] + [y_2'(x) + a(x)y_2(x)]. \end{aligned}$$

By our assumption,  $y_1$  solves the given equation, so the first bracketed term must be zero. The same treatment can be used for the second term and we conclude that

$$[\alpha y_1(x) + y_2(x)]' + a(x)(\alpha y_1(x) + y_2(x)) = \alpha \cdot 0 + 0 = 0.$$

We confirmed that  $\alpha y_1 + \beta y_2$  is a solution on  $I$ . □

## 9a. Integrating factors

There is another approach to equations of the type  $y' + a(x)y = b(x)$ . It goes as follows. We multiply the equation by a function  $\mu(x)$ :

$$\mu(x)y' + \mu(x)a(x)y = \mu(x)b(x).$$

Now imagine for a moment that  $\mu(x)$  was chosen in such a way that  $[\mu(x)]' = \mu(x)a(x)$ . Then we can proceed as follows:

$$\begin{aligned} \mu(x)y' + \mu(x)a(x)y = \mu(x)b(x) &\implies \mu(x)y' + [\mu(x)]'y = \mu(x)b(x) \implies [\mu(x)y]' = \mu(x)b(x) \\ \implies \mu(x)y &= \int \mu(x)b(x) dx \implies y(x) = \frac{1}{\mu(x)} \int \mu(x)b(x) dx. \end{aligned}$$

We just solved the given equation (assuming we can handle the integral). This very special function  $\mu(x)$  is called the **integrating factor** of this equation and we would obviously like to know whether it actually exists. It turns out that in fact it does if we can find some antiderivative  $A(x)$  of  $a(x)$ . Then we can take  $\mu(x) = e^{A(x)}$  and it is easy to show that it actually does what it should.

When we substitute this  $\mu(x)$  into the formula for solution, we obtain

$$y(x) = e^{-A(x)} \int e^{A(x)} b(x) dx = e^{-A(x)} (B(x) + C).$$

By a remarkable coincidence, this is exactly the formula provided by variation of parameter. Let's see it in action.

**Example 9a.a:** Consider the equation  $y' = x + y$ . We can write it as  $y' - 1 = x$  and see that it is a linear equation of order 1. We have  $a(x) = -1$ , so  $A(x) = -x$  and we remember that the integrating factor is  $\mu(x) = e^{-x}$ . We proceed as outlined above:

$$\begin{aligned} y' - y = x &\implies e^{-x}y' - e^{-x}y = x e^{-x} \implies e^{-x}y' + [e^{-x}]'y = x e^{-x} \\ &\implies [e^{-x}y]' = x e^{-x} \implies e^{-x}y = \int x e^{-x} dx \\ &\implies e^{-x}y = -x e^{-x} - e^{-x} + C y(x) = -x - 1 + C e^x, \quad x \in \mathbb{R}. \end{aligned}$$

We easily check that we obtained a general solution.

△

The reader is encouraged to solve this equation by variation.

So there are two ways to approach essentially the same calculations: the variation way and the integrating factor way. As it applies to equations of order one, the choice is down to personal preferences. To apply variation, one needs to remember the outline of the procedure: First solving the homogeneous equation, then changing the constant, then substitute, enjoy the miracle of cancelling and work it out. To use the integration factor way people usually memorize the formula for  $\mu(x)$  and then the process of multiplying and rewriting derivative. My personal preference is to remember ideas rather than formulas, so I go with variation and present it here.

The difference is more pronounced when we ask whether these two approaches can also extend to other types of equations. It turns out that with integration factor we do not have a natural generalization to richer types of equations. On the other hand, the integrating factor approach can be applied to some specific higher order equations for which we do not really have another approach, and it offers some insight in calculus of more variables. However, since these topics do not concern us here, this does not earn any plus points to integrating factors as far as this book is concerned.

On the other hand, the general idea of the variation approach (first solving homogeneous equation, then finding a particular solution) readily generalizes to linear equations of higher orders and systems of linear equations, as we will soon see in the corresponding chapters.

This concludes our theoretical study of differential equations of order 1.

## 10. Applications of First Order ODEs

In this chapter we look at some applications of differential equations of the first order.

### 10a. Demographics

Already in the middle ages thinkers were thinking about the population growth of mankind (and other animals), trying to see some patterns. One famous result of these attempts is the Fibonacci sequence that allegedly describes how a group of rabbits multiplies. If we denote by  $F_n$  the number of rabbits in this group in month  $n$ , then (according to Fibonacci) it must satisfy the equation  $F_{n+1} = F_n + F_{n-1}$ . This formula then represents a mathematical **model**, that is, a mathematical description of a certain natural phenomenon. Having formed such a model, we may attempt to solve it and then investigate the properties of the solution(s), thus gaining some insight into the phenomenon. In particular, the Fibonacci model can be solved to provide an explicit formula for the rabbit population and we learn that  $F_n$  grows about as fast as  $\varphi^n$ , where  $\varphi = \frac{1+\sqrt{5}}{2} \approx 1.6$  is the golden ratio.

However, we will not investigate the Fibonacci model here, as it is not a differential equation after all. Instead, we will look at some conceptual things related to this model, because already this 700 years old mathematical model exhibits traits that are still present when modelling natural phenomena using the most up-to-date methods.

First, the numbers  $F_n$  are obviously not to be taken literally, rather, they are some averages or typical cases, because rabbits stubbornly refuse to give birth on schedule on the first of each month. This is a typical feature: If we want to capture some behaviour, we often have to simplify it to be able to succeed. In particular, if the model is based on averages, then it can be expected to work reasonably well only for large populations.

We may want to alleviate this to a certain degree by taking smaller time intervals. We can look at numbers of rabbits every week, every day, or even by the hour. When the time intervals get really tiny, differences become differentials and we end up with a differential equation. We can therefore expect that a differential equation will be better at describing changing situations, and this is usually so, indeed. In fact, this is one of the popular ways to derive a differential equation: We start our exploration by looking at the given process over a very short time interval and then pass to a limit.

Another interesting aspect is that Fibonacci did not derive his formula by looking at some data on rabbit population and fitting a curve to it. Instead, after watching rabbits in a neighbor's yard, he thought of the mechanism behind rabbit procreation and derived his rule based on two assumptions: That rabbits younger than one month do not breed, and that those that are older breed with (on average) two little bunnies per month per couple. This exemplifies a true scientific approach: We are not content with merely fitting some formula to known outcomes (although this does have its uses in some situations), but we want to understand what actually makes things do what they do. The underlying mechanism often features some sort of feedback or mutual dependence of quantities, and when we describe this in mathematical language, we usually get an equation, and this equation is very often a differential one.

It is curious that there is no third rule: That rabbits die after a while. It would be possible to determine a rabbit's average lifetime and incorporate this into the model. However, the resulting formula would be significantly more difficult to handle. Again, this is something that appears every time someone tries to create a mathematical model of a real life phenomenon: The better model you create, the less likely it is that you can solve it.

Generally, mathematical models are derived by simplifying the complicated real-life situations. As such, they describe something slightly different, and those slight differences can lead to fundamental differences between the model and the reality that it attempts to describe. This is an important aspect of mathematical modelling, and we always have to carefully judge validity of our models. If we are not careful, then it may happen that all our calculations are correct, but the result is

wrong as a description of reality.

### 10a.1 Exponential growth

The general experience is that the more people (or rabbits, or bacteria, or whatever) there are, the more new people (etc.) will appear (and die). In order to get some information out of this idea, we have to introduce mathematics. We choose some basic time unit (a day, for instance) and ask what is the net change  $\Delta y$  of a population of size  $y$  after the time unit passes.

To answer this we have to say precisely what we mean when we say “the more we have, the more we gain”. The simplest approach is to assume that the net change is directly proportional to the population size, so the change within the time unit is  $\Delta y = ry$ . Here  $r$  is some constant depicting how eagerly the population multiplies and dies, it is called a reproduction rate or a growth rate. Notice that this rate is necessarily an average, which again means that the model we will obtain is likely to work only for large populations.

Now consider some (really small) time interval  $\Delta t$  measured in the time units. Given some population size  $y(t)$  at time  $t$ , then its size at time  $t + \Delta t$  should be about  $ry(t) \cdot \Delta t$ . Again, we are talking averages here, because for really small  $\Delta t$  we cannot expect half a person to be born. In any case, we finally have a formula.

$$\Delta y = ry\Delta t.$$

We write it as

$$\frac{\Delta y}{\Delta t} = ry$$

and decide to pass to a limit, sending  $\Delta t \rightarrow 0^+$ . Then on the left we obtain a derivative of  $y$  with respect to time, which is traditionally denoted as  $\dot{y}$ . If you do not like it, use  $y'$  instead. We obtain

$$\dot{y} = ry.$$

This differential equation therefore captures mathematically the idea that a population change is directly proportional to its size. It is the exponential model for population growth.

This equation is easily solved by separation:

$$\int \frac{dy}{y} = \int r dt \implies \ln |y| = rt + C \implies y(t) = \pm e^C e^{rt}.$$

As usual, we denote  $\pm e^C = D \neq 0$  and notice that by allowing  $D = 0$  we also include the stationary solution  $y(x) = 0$ . We obtain the general solution  $y(t) = D e^{rt}$ , where now  $D \in \mathbb{R}$  can be arbitrary.

It is easy to see that given an initial value  $y(0)$  at time  $t = 0$ , we obtain the particular solution  $y(t) = y(0)e^{rt}$ . In general, an initial condition  $y(t_0) = y_0$  leads to  $y(t) = y_0 e^{r(t-t_0)}$ . As a mathematical solution it is valid on  $\mathbb{R}$ , but in applications we usually consider it only for  $t \geq t_0$ .

Exponential growth appears naturally in many areas and some applied people simply remember that

$$\dot{y} = ry \implies y(t) = y(0)e^{rt}, \quad t \geq 0.$$

I know of some (famous) universities where engineering freshmen were expected to memorize this, that’s how useful it is.

Is this model any good? It is so simple that one gets suspicious, and with a good cause. In reality we rarely have the same multiplicative constant for ever. For instance, if a population grows so much that food becomes scarce, the dying rate shoots up. Still, there are situations where it works quite well. For instance, it is a good fit for growth of bacteria in a Petri dish at its initial stages, when there are enough resources, and scientists use this model with great success. The constant  $r$  is determined experimentally.

Similarly, the exponential growth was a pretty good fit for European population about a hundred years ago, when Europe went through its expansive period. This led many people to worry about the future, and in the mid-20th century the science-fiction community came up with many amusing

stories where people would have to eat each other, stand on each other's heads due to lack of room and such. Needless to say, we no longer think that the exponential growth is a good fit for the mankind.

To sum it up, in biology the exponential growth is a cheap way to obtain reasonable answers under specific conditions.

Animals (including people) are notoriously unreliable, so mathematics has trouble dealing with them. Things are much better with physics and engineering. There are some situations where the response is proportional to the value, and then the exponential model applies. One of the best known examples is that of radioactive decay where due to natural radioactivity, atoms break down with the same probability, so the number of those that decay is directly proportional to the amount we have at the moment. The equation  $\dot{y} = ry$  has a negative  $r$  now and the model is really faithful. It is the basis for radioactive dating that found much use in geology and archeology.

An interesting twist is **exponential growth with harvesting**. We assume that some population breeds freely (exponential rule with  $r > 0$ ), but we keep removing a constant amount  $H > 0$  per time unit from the population. Over some time interval  $\Delta t$  we get the net change

$$\Delta y = ry\Delta t - H\Delta t.$$

We divide by  $\Delta t$  and using  $\Delta t \rightarrow 0^+$  we arrive at the equation

$$\dot{y} = ry - H.$$

This equation is autonomous, so we can investigate stability (see chapter 8). Putting  $y' = 0$  we find one equilibrium  $y_s = \frac{H}{r}$  for our model. When  $y(0) > y_s$ , then  $\dot{y} = ry - H > 0$  and thus the corresponding solution  $y$  is increasing. Similarly we find that solutions with  $y(0) < y_s$  are decreasing. Consequently,  $y_s$  is an unstable equilibrium.

We can interpret our observations in a very useful way: Given some population with initial size  $y(0)$  and reproduction rate  $r$ , we can afford to harvest  $H \leq ry(0)$  if we do not want it to die out.

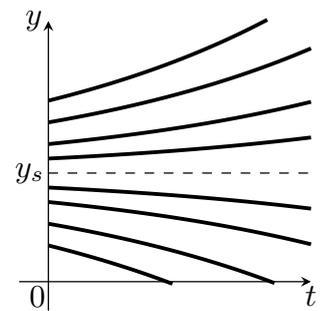
We can actually solve our equation by separation:

$$\int \frac{dy}{ry - H} = \int dt \implies \frac{1}{r} \ln |ry - H| = t + C \implies |ry - H| = e^{rC} e^{rt},$$

the usual tricks lead to  $y(t) = \frac{H}{r} + D e^{rt}$ . The value  $D = 0$  leads to the stationary solution  $y(t) = \frac{H}{r}$ . An initial condition at time  $t = 0$  determines  $D$ , we have

$$y(t) = \frac{H}{r} + \left(y(0) - \frac{H}{r}\right) e^{rt}, \quad t \geq 0.$$

So the outcome is just a shifted exponential.



## 10a.2 Logistic growth

One of the problems with the exponential model is the assumption that the reproduction rate  $r$  always stays the same. Every population lives in some environment that has a limited capacity  $K$ , it will simply not support more than this (in population biology this is called the “carrying capacity”). Experience suggests that under favourable conditions, every population has a natural (base) reproduction rate  $r$  that combines its natural natality and mortality. However, when the population size is near the capacity of environment  $K$ , then the actual reproduction rate drops down due to the “crowding effect” (factors like increased aggression, lack of food etc.).

It is not clear how to capture this mathematically, and one popular way is to introduce a linear dependency of the actual reproduction rate on the population size. The **effective reproduction rate** is then

$$r \left(1 - \frac{y}{K}\right) = \frac{r}{K}(K - y).$$

How does it work? When  $y$  is small compared to the environment capacity, then  $\frac{y}{K} \approx 0$  and the

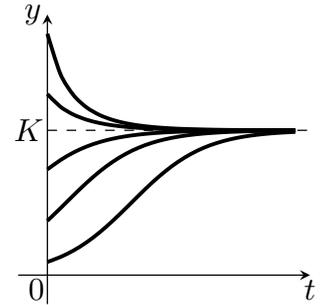
effective rate is about  $r$ . When the population size is near the capacity  $K$ , the reproduction rate is about zero.

Combining this with the basic idea of the exponential growth yields the equation

$$\dot{y} = \frac{r}{K}y(K - y).$$

This is the **logistic equation**. How does it work? When the population is relatively small compared to capacity, then this equation is approximately  $\dot{y} = ry$ , so the population grows exponentially. When the population size is near the capacity, then  $\dot{y}$  is almost zero and the population stagnates. If it happens that  $y > K$  (for instance if we release animals into environment that cannot support them all), then the effective reproduction rate becomes negative and the population size decreases.

We look at the behaviour closer. Setting  $\dot{y}$  to zero we obtain two stationary solutions  $y(x) = 0$  and  $y(x) = K$ , or two equilibria  $y_s = 0, K$  if you prefer. The slope field shows that the equilibrium  $y_s = 0$  is unstable, while  $y_s = K$  is stable. We see several typical solutions on the right. It seems that all populations that start with  $y(0) > 0$  eventually end up with population size about  $K$ , the limit of their environment.



Sometimes one can learn about the behaviour of the system from the equation itself. For instance, we see that the increase in population is proportional to  $y(K - y)$ . It is easy to find that the population grows fastest when  $y = \frac{1}{2}K$ .

We can solve the logistic by separation. It just takes a bit more work, we need to use the partial fractions decomposition.

$$\begin{aligned} \int \frac{K dy}{y(K - y)} &= \int r dt \implies \ln|y| - \ln|K - y| = rt + C \\ \implies \ln\left|\frac{y}{K - y}\right| &= rt + C \implies \frac{y}{K - y} = \pm e^C e^{rt} \\ \implies y &= \frac{\pm K e^C e^{rt}}{1 \pm e^C e^{rt}}. \end{aligned}$$

As usual we denote  $D = \pm e^C$ , which is a non-zero number, and we notice that by allowing  $D = 0$  we also include the stationary solution  $y(x) = 0$ . However, the other one cannot be obtained from this formula, so it has to be listed separately. We conclude that the logistic model has a general solution

$$y(t) = \frac{KD e^{rt}}{1 + D e^{rt}} \quad \text{or} \quad y(t) = K, \quad t \geq 0.$$

This formula behaves in the way predicted by our analysis above. In particular, we note that for  $r > 0$  the exponential  $e^{-rt}$  tends to zero at infinity, so

$$\lim_{t \rightarrow \infty} \left( \frac{KD e^{rt}}{1 + D e^{rt}} \right) = \lim_{t \rightarrow \infty} \left( \frac{KD}{e^{-rt} + D} \right) = \frac{KD}{0 + D} = K.$$

A given initial value  $y(0)$  at time  $t = 0$  determines the solution

$$y(x) = \frac{Ky(0)e^{rt}}{(K - y(0)) + y(0)e^{rt}}, \quad t \geq 0.$$

How good is the logistic growth as a model of population dynamics? It is definitely better than the exponential growth, but the idea that the effective reproduction rate is proportional to population size is still not quite right. Indeed, one would expect that as a population starts small, then its reproduction rate will stay the same for quite some time. Only when the population size crosses a certain threshold will the reproduction rate drop sharply. Indeed, there are alternative models that use non-linear reproduction, but they are significantly harder to handle.

The logistic growth is therefore a reasonable compromise between simplicity and usefulness. It

describes rather well the growth of bacteria in a Petri dish while there is still enough food for them, and it can also work rather well with human populations if we do not expect too much from it. Having some set of data, it is not difficult to derive a best fit estimate for the coefficients  $r, K$ , which can be useful for instance for predicting various wildlife situations.

Perhaps surprisingly, logistic growth found its use not just in biology, but also in medicine (tumor growth), chemistry (autocatalytic reaction) or in economy (diffusion of innovation in economy).

Just like with the exponential growth, we can also introduce harvesting to the logistic model. There are actually two common versions.

### Logistic growth with constant harvesting.

We have the usual parameters  $r, K > 0$  and also a parameter  $H > 0$  describing constant harvesting. The differential equation is

$$\dot{y} = \frac{r}{K}y(K - y) - H.$$

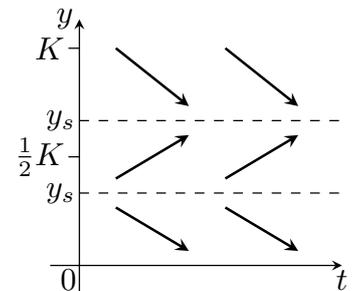
This can be also solved by separation, but this time the partial fractions are less nice, so this is the time to put to practice our analyzing skills from chapter .

Setting the derivative equal to zero we obtain a formula for equilibria

$$y_s = \frac{1}{2}K \pm \sqrt{\left(\frac{K}{2}\right)^2 - \frac{KH}{r}}.$$

There are three distinct cases.

For  $H > 0$  and  $H < \frac{rK}{4}$  we obtain two positive real equilibria. We see the slope field on the right, and we conclude that the larger equilibrium is stable, while the smaller one is unstable. We are always interested in sustainable situations, and we see that now it does not only depend on  $H$ , but also on the initial size of the population. If  $y(0)$  is less than the smaller stationary solution, then the population becomes extinct.



It is more useful to ask about harvesting as it depends on initial data. We start with an imprecise statement. Since the smaller equilibrium is never above  $\frac{1}{2}K$ , we see that if the initial population size satisfies  $y(0) \geq \frac{1}{2}K$ , then any harvesting below  $\frac{rK}{4}$  (so that we have this situation of two equilibria) is sustainable.

More precisely, the sustainability condition

$$y(0) \geq \frac{1}{2}K \pm \sqrt{\left(\frac{K}{2}\right)^2 - \frac{KH}{r}}$$

can be rearranged as

$$H \leq \frac{r}{K}y(0)(K - y(0)).$$

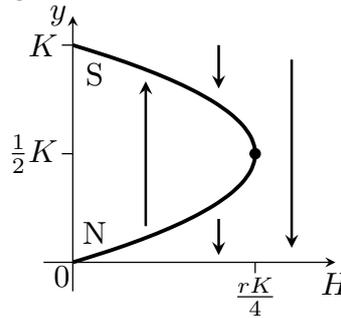
The expression on the right is exactly the right-hand side of the logistic differential equation at time zero, that is, it is the initial rate of growth in the model without harvesting. In words, the situation is sustainable if the harvesting does not exceed the initial growth rate of the population if we did not harvest.

Note that as we increase  $H$  towards  $\frac{rK}{4}$ , then the two equilibria are drawn closer together, towards  $\frac{K}{2}$ . When  $H = \frac{rK}{4}$ , the two equilibria collapse into one,  $y_s = \frac{1}{2}K$ , and the middle strip of increase disappears. Solutions both above and below the equilibrium decrease, so  $\frac{1}{2}K$  is unstable. In particular, all populations that start with  $y(0) < \frac{1}{2}K$  become extinct.

The third case: For  $H > \frac{rK}{4}$  there are no equilibria, and all positive solutions are decreasing. All populations die out.

We have an interesting situation here, a system that depends on a parameter, and the behaviour of the system changes significantly as we change the parameter, with an abrupt change when  $H = \frac{rK}{4}$ . We talked about this in section on bifurcations, and thus we know that we can express

our observations by a bifurcation diagram.



### Logistic growth with proportional harvesting.

Here we introduce a constant  $h > 0$  describing the proportion of the population that we harvest. It is closely related to the effort that we put into this harvesting. The model is

$$\dot{y} = \frac{r}{K}y(K - y) - hy.$$

This can be written as

$$\dot{y} = \frac{r}{K}y\left(\frac{r-h}{r}K - y\right)$$

and analyzed just like the logistic growth. We see two equilibria,  $y_s = 0$  and  $\frac{r-h}{r}K$ . The only way to have a positive equilibrium is to take  $h < r$ , which seems natural, we definitely do not want to take more from the population than its natural growth. For  $h < r$  the positive equilibrium is stable, which is great.

Note that at any given time, the amount that we actually harvest is  $hy$ . How should we harvest to get as much as possible, without destroying the population? This is called the “maximum sustainable yield” or MSI. To find out, consider a sustainable scenario, that is,  $h < r$ . The population size will relatively quickly approach its limit value  $\frac{r-h}{r}K$ , so the harvesting will be about  $h\frac{r-h}{r}K$ . Analyzing the behaviour of  $h(r-h)$  we learn that the maximum value happens when  $h = \frac{1}{2}r$ , then the MSI is  $\frac{1}{4}rK$ .

This type of reasoning found widespread use in wildlife management and related fields. However, it can be dangerous to rely on it too much, because as we observed, the logistic model does have its flaws. In particular, the logistic model with proportional harvesting does not take into account seasonal changes. Again, better models are available, but they are harder to analyze.

## 10b. Free fall

In the previous section we have shown one popular way to arrive at a differential equation: We investigated what happens over some time interval  $\Delta t$  and then we let  $\Delta t$  go to zero. In this section we will show another popular approach: We freeze the situation that we study and inquire about influences. In particular, people often go this way in physics, so we turn there for an example.

Imagine a stone falling down (vertically). We will be interested in its movement, and for convenience we will measure its vertical distance from the place where we released it at time  $t = 0$  and denote it  $y$ . In particular,  $y(0) = 0$  and the  $y$ -axis points down.

Now we freeze the situation, observe the stone hanging in the air and ask about the influences. There is one that is hard to miss, namely the gravity, and we know from physics that as a force it influences acceleration, that is, the second derivative. The resulting differential equation  $\ddot{y} = g$  is very simple and we explored it in chapter 1. However, such a model is reasonably precise only for small velocities. Once the stone picks up speed, another (previously ignored) factor becomes significant, namely the air resistance or drag as it is called in aviation.

Physicists tell us that the force created by the drag is directly proportional to actual velocity, so it should be  $k(\dot{y})^2$ . The constant  $k$  captures influences like the shape of the falling body, its cross-section, air density and other things that can be considered constant (so either the stone is symmetric or it does not tumble).

The physicists (or our common sense) also tell us that this force always acts in the opposite direction from velocity, and this is a problem now. For instance, if we release the stone by actually flipping it up, it will first climb until it loses its energy, then it starts falling, and the drag force will therefore switch directions midflight. It is possible to capture this mathematically, but the resulting equation would be significantly harder to analyze.

We will therefore simplify our situation by assuming that the stone never goes up, in other words, we perhaps simply let it go (that  $\dot{y}(0) = 0$ ) or we may even throw it downwards. The stone will then always go down, and the drag consequently always points up. We obtain the following differential equation for **free fall with air resistance**:

$$\ddot{y} = g - k[\dot{y}]^2.$$

This is a second order ODE. However, we only learned how to solve equations of order 1 so far. Did we introduce this example too soon? Actually, not quite. We will soon learn how to solve differential equations of higher order, but only linear ones, and this one is not linear. In fact, this equation cannot be solved directly by any method that is typically covered in ODE courses, it is one of the good ODEs.

Note that we will also not develop any general existence and uniqueness theory for higher order equations, but this one is really well behaved and its solutions are given uniquely by a pair of initial conditions that specify  $y(0)$  and  $y'(0)$ .

When our analytic methods fail, we usually turn to numerical methods for help, but for this particular ODE we can actually find a simple way to make it tractable. We notice that  $y$  itself is not in the equation, so we switch focus and instead of the elevation we look at velocity  $v = \dot{y}$ . Then the equation becomes

$$\dot{v} = g - kv^2.$$

This is an autonomous first-order ODE and we have tools for dealing with it. First, we notice by putting  $\dot{v} = 0$  that there are two equilibria, namely  $v_s = \pm\sqrt{\frac{g}{k}}$ . Since we assume that the stone is only moving down and we measure the displacement downwards, the velocity cannot be negative and it only makes (physical) sense to investigate the stationary solution  $v(t) = \sqrt{\frac{g}{k}} = v_T$ . This is called the **terminal velocity**.

We notice that for  $v > v_T$  we obtain  $\dot{v} < 0$  and for  $0 \leq v < v_T$  we obtain  $\dot{v} > 0$ , which means that our stationary solution is stable. The vector field suggests that no matter what non-negative velocity our stone starts with, its speed will eventually become close to  $v_T$ . Common sense tells us that in fact all free falls with air resistance behave this way, because even if we flip the stone up, it will soon stop and then start falling down, which can be treated as a new process and the velocity necessarily converges to  $v_T$ .

So this  $v_T$  is really interesting. It is the velocity at which the force of gravity and the force of drag are in perfect balance. It all fits together. If the stone starts with a faster velocity, then the drag gets stronger than gravity and slows down the stone. If the stone starts slower, then the drag is weaker than gravity and the stone will speed up.

Experiments suggest that for a typical human specimen, the terminal velocity can be between 200 and 300 km per hour and it takes about a minute to reach it. If you are ever in a situation when this could be of concern, it helps to know your differential equations. Generally people do not have much to do while falling, so you may as well form and solve this differential equation and then focus on the terminal velocity  $v_T = \sqrt{\frac{g}{k}}$ . Obviously, we want to make it as small as possible. Influencing the gravity constant  $g$  seems to be rather difficult, so it makes more sense to focus on the constant  $k$ . Generally, it pays to make it as large as possible: Increase your cross-section (spread everything you have, including your ears and hanky), and make a shape that is not very aerodynamic. You surely wouldn't have thought of this without knowing your differential equations. If we also add a bit of luck, it may all end up well (and it sometimes did).

We conclude this section by observing that we can actually solve this differential equation by separation. Integration is easier if we introduce a constant  $a = \sqrt{\frac{g}{k}}$ , then the equation can be written as  $\dot{v} = k(a^2 - v^2)$ . We solve:

$$\begin{aligned} \int \frac{dv}{a^2 - v^2} &= \int k dt \implies \int \frac{\frac{1}{2a}}{a+v} + \frac{\frac{1}{2a}}{a-v} = \int k dt \\ &\implies \ln \left| \frac{a+v}{a-v} \right| = 2akt + C \implies \frac{a+v}{a-v} = \pm e^C \cdot e^{2\sqrt{gk}t}. \end{aligned}$$

As usual we denote  $D = \pm e^C$  and note that setting  $D = 0$  yields the stationary solution  $y(t) = \sqrt{\frac{g}{k}}$ . Solving in general for  $v(t)$  we get

$$v(t) = \sqrt{\frac{g}{k}} \frac{D e^{2\sqrt{gk}t} - 1}{D e^{2\sqrt{gk}t} + 1}, \quad t \geq 0.$$

For the simplest initial condition  $v(0) = 0$  we get  $D = 1$ .

Analyzing this formula we confirm our earlier observations about the behaviour of solutions. In particular, letting  $t$  go to infinity we see that

$$\lim_{t \rightarrow \infty} (v(t)) = \sqrt{\frac{g}{k}}.$$

We have a formula for the velocity  $v(t)$  and it is time to return to the elevation function  $y(t)$ . We can recover it as

$$y(t) = \int_0^t v(x) dx.$$

This integral can be handled by the substitution  $u = D e^{2\sqrt{gk}x}$ , leading to partial fractions decomposition.

$$\begin{aligned} \int \sqrt{\frac{g}{k}} \frac{D e^{2\sqrt{gk}x} - 1}{D e^{2\sqrt{gk}x} + 1} dx &= \sqrt{\frac{g}{k}} \int \frac{u-1}{u+1} \frac{du}{2D\sqrt{gk}u} \\ &= \frac{1}{2Dk} \int \frac{u-1}{u(u+1)} du = \frac{1}{2Dk} \int \frac{2}{u+1} - \frac{1}{u} du \\ &= \frac{1}{2Dk} \left( \ln((u+1)^2) - \ln|u| \right) = \frac{1}{2Dk} \ln \left( \frac{(D e^{2\sqrt{gk}x} + 1)^2}{D e^{2\sqrt{gk}x}} \right). \end{aligned}$$

Therefore

$$y(t) = \left[ \frac{1}{2Dk} \ln \left( \frac{(D e^{2\sqrt{gk}x} + 1)^2}{D e^{2\sqrt{gk}x}} \right) \right]_0^t = \frac{1}{2Dk} \ln \left( \frac{(D e^{2\sqrt{gk}t} + 1)^2}{D e^{2\sqrt{gk}t}} \right) - \frac{1}{2Dk} \ln \left( \frac{(D+1)^2}{D} \right), \quad t \geq 0.$$

I admit I do not quite feel like analysing this, and I also did not confirm that it truly is a solution by substituting this formula into our free fall equation. If you feel bad about it, you are welcome to have a go at it.

## 10c. Draining a tank

Imagine a tank, for simplicity a rectangular one with base area equal to  $A$ . There is water in it, and we make a small hole at the bottom to let the water out. What is the water level at time  $t$ ?

As the water pours out, it influences the water level (call it  $h$  as measured from the bottom up) through volume. This is the key moment of our investigation, we need to establish some relationship between the change in water volume  $\Delta V$  and the change in water level  $\Delta h$ . In general (for instance in the case of a regular bathtub) this can be rather complicated, so we will stick here with a simple situation of a tank that has the same cross-section (of area  $A$  throughout), for instance a vertically oriented cylinder or a rectangular tank. Once we know that the sides of tank

are vertical, we can connect its volume  $V$  with the area of the base  $A$  and the water elevation  $h$  as follows:  $V = Ah$ . It is easy to observe that an analogous formula also connects changes:  $\Delta V = A\Delta h$ .

Now we can focus on the change of volume  $\Delta V$  that happens during some short time interval  $\Delta t$  starting at time  $t$ . It is determined by how much water left the tank through the hole. This in turn is determined by two factors: The size of the hole, so let it be  $a$ , and the velocity at which the water pours out.

For that we have to turn to physics, and it supplies us with the Torricelli's law that says: The velocity at which the water pours out is given by  $v = \sqrt{2gh}$ , where  $h$  is the difference between the water level and the level of the hole. We conveniently placed the hole at the bottom, so  $h$  is exactly the water level we are interested in. We therefore obtain the following formula:

$$\Delta V = -a\sqrt{2gh}\Delta t.$$

We had to put minus there, as the water volume is decreasing. Given the regular shape of the tank, we can apply our observation above and write

$$\frac{1}{A}\Delta h = -a\sqrt{2gh}\Delta t.$$

We rewrite it as

$$\frac{\Delta h}{\Delta t} = -aA\sqrt{2g}\sqrt{h}.$$

We collect all the constants into, say,  $2c$  (you will see shortly why), and passing to zero with  $\Delta t$  we obtain a differential equation.

$$\dot{h} = -2c\sqrt{h}.$$

One could also arrive at this equation using the traditional physics approach: We freeze the situation and observe that the key influence is the water pouring out. It does so at the rate  $a\sqrt{2g}\sqrt{h}$ , and this rate directly corresponds to the rate of change of volume, but in the opposite direction (positive water outflow decreases the volume), hence  $\dot{V} = -a\sqrt{2g}\sqrt{h}$ . From the formula  $V = Ah$  we also obtain  $\dot{V} = a\dot{h}$  and arrive at the same differential equation.

This differential equation is obviously separable, so we apply the recommended procedure. First we note that  $h(t) = 0$  is a stationary solution, and for  $h \neq 0$  we proceed as usual.

$$\begin{aligned} \frac{dh}{dt} = -2c\sqrt{h} &\implies \int h^{-1/2}dh = - \int 2c dt \\ &\implies 2h^{1/2} = 2C - 2ct \implies h(t) = (C - ct)^2. \end{aligned}$$

If we know the water level  $H_0$  at time  $t = 0$ , we easily determine that  $C = \sqrt{H_0}$ . Thus we get the solution

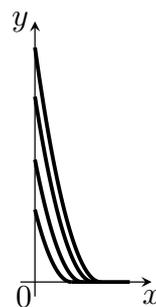
$$h(t) = (\sqrt{H_0} - ct)^2.$$

If we take it as a mathematical problem, then we see this solution valid on  $\mathbb{R}$ , but as a tank problem it only makes sense for  $t \in [0, \frac{\sqrt{H_0}}{c}]$  as the tank becomes empty then. A realistic solution is therefore given as follows:

$$h(t) = \begin{cases} (\sqrt{H_0} - ct)^2, & t \in [0, \frac{\sqrt{H_0}}{c}]; \\ 0, & t \geq \frac{\sqrt{H_0}}{c}. \end{cases}$$

Note that for  $H_0 = 0$  this includes the stationary solution.

It should be noted that the Torricelli's law is a simplified description of the actual process. It only works if the hole is very small, and it ignores the effects of viscosity. Still, it is reasonably close to reality if we use water. You can confirm it with an experiment. Take a rectangular fish bowl and draw regularly spaced vertical lines on its side. These represent time measured in minutes. On the first line, mark the current water level. Then drill a small hole next to the bottom and start your timer. After every minute, mark the current water level on the next vertical line. The marks should follow the shape of a parabola. Disclaimer: No fish were harmed when describing this project.



## 10d. The mixing problem

When it comes to applications of differential equations in chemistry, people usually turn to the mixing problem.

Imagine a tank holding a volume  $V$  of water with salt dissolved in it. We are interested in the concentration  $C(t)$  that changes with time. Why?

There is water pouring into the tank at the rate  $V_I$  that also has salt dissolved in it, with known concentration  $C_I$  that may or may not be constant in time. There is also water pouring out of the tank at the rate  $V_O$ . Obviously, if the incoming and outgoing rates are not the same, then the volume of water in the tank will change.

Intuitively we feel that some salt is getting in, some other salt is going out, so it should be possible to work out the balance somehow. Where do the differential equations come in? Common sense tells us that the concentration of salt in the water pouring out of the tank should be the same as the concentration in the tank (near the outlet), so we actually have not know the solution  $C(t)$  of our problem to work out its working. Such circular situations usually result in differential equations.

We now move to mathematics, and we will start making assumptions in order to be able to do so. The first assumption is that the water in the tank is always perfectly mixed all the time, so at every time there is a unique concentration  $C(t)$  of salt everywhere in the tank, and therefore also in the outgoing water. Of course, this is essentially impossible to achieve, but more realistic models are way beyond our reach.

We are ready to start our mathematical exploration, and it is more convenient to actually talk of the amount of salt rather than the concentration. Obviously, the quantities volume, concentration and mass of salt are connected by the formula  $m = cV$ .

We start by introducing notation. The volume  $V(t)$  of water in the tank is naturally measured in liters (imagine gallons if you prefer), the amount of salt in the tank is  $M(t)$  in kilograms, and for time  $t$  it is reasonable to think in minutes. Then the rates  $V_I, V_O$  of incoming and outgoing flows are measured in l/min, and the concentrations  $C(t)$  and  $C_I$  in kg/l. Obviously, the concentration in the tank is  $C(t) = \frac{M(t)}{V(t)}$ .

Now we will make a balance equation for the amount of salt in the tank. Within some short time interval  $\Delta t$  that starts at some time  $t$ , the change  $\Delta M$  in the amount of salt is determined by two factors. First, during the time  $\Delta t$ , the amount of water coming in is  $V_I \Delta t$ , and it brought  $C_I(t) V_I \Delta t$  kg of salt with it. Here we used a traditional trick: We know that the incoming concentration may change with time, but if the time interval  $\Delta t$  is extremely small, then the changes in the incoming concentration are so small that they can be ignored and we simply use the concentration  $C(t)$  at the start of this time interval.

Similarly we argue that during this time,  $V_O \Delta t$  liters of water went out, taking  $C(t) V_O \Delta t$  kilograms of salt with it. Thus we have the following equation:

$$\Delta M = C_I(t) V_I \Delta t - C(t) V_O \Delta t.$$

This is a **general mixing balance equation** (assuming perfect mixing).

We start with some simple cases. First we will focus on the situation when  $V_I = V_O$ , that is, the water is coming in and out at the same rate, so the volume of water in the tank remains constant.

Then we can divide both sides of the general balance equation by  $V$ , and we get  $\frac{\Delta M}{V} = \Delta C(t)$  on the left. We also divide both sides by  $\Delta t$  to obtain

$$\frac{\Delta C}{\Delta t} = C_I(t) \frac{V_I}{V} - C(t) \frac{V_O}{V}.$$

We deduced it under the assumption that  $V_I = V_O$ , so we simplify, and above all, we let  $\Delta t \rightarrow 0$ . We obtain

$$C'(t) = \frac{V_I}{V} (C_I(t) - C(t)).$$

This equation is still too general to be solved. However, if somebody gives us a specification for the incoming concentration  $C_I(t)$ , then we can try to solve the appropriate equation.

The simplest case is when the incoming concentration is constant. Then we obtain the equation

$$C'(t) = \frac{V_I}{V} (C_I - C(t)).$$

This is the simplest mixing equation, and it can handle all kinds of interesting cases. For instance, we may start with some salt in the tank, so  $C(0) > 0$ , and we may start pouring clean water in, so  $C_I = 0$ . Experience tells us that the water in the tank should gradually clear up. Or we can start with clean water in the tank, that is,  $C(0) = 0$ , and start pouring some water with salt into it. Whatever the initial configuration, the common sense suggests that the concentration in the tank should eventually (perhaps at time infinity) get equal to the concentration in the incoming solution. Will mathematics confirm this?

We solve the above equation easily by separation. We have the obvious stationary solution  $C(t) = C_I$ , and for  $C \neq C_I$  we follow the usual procedure, for convenience we will write  $V_R$  for  $\frac{V_I}{V}$ , the rate at which the water in the tank refreshes:

$$\begin{aligned} \frac{dC}{dt} &= V_R(C_I - C) \\ \implies \int \frac{dC}{C_I - C} &= \int V_R dt \\ \implies -\ln|C_I - C| &= V_R t - c \implies \ln|C_I - C| = c - V_R t \\ \implies C_I - C &= \pm e^c e^{-V_R t} \implies C = C_I - D e^{-V_R t}. \end{aligned}$$

The case  $D = 0$  incorporates the stationary solution, and thus we get a general solution of our equation:

$$C(t) = C_I - D e^{-V_R t}, \quad t \geq 0.$$

Assuming the knowledge of  $C(0)$ , we easily determine  $D = C_I - C(0)$  and write the formula that we have been asking for:

$$C(t) = C_I - (C_I - C(0))e^{-V_R t}, \quad t \geq 0.$$

We easily observe that  $C(t) \rightarrow C_I$  as  $t \rightarrow \infty$ , just as we expected.

In fact, people working in the chemical industry are not really interested in infinity, rather, then need to know things like how to set up their production lines to achieve a certain concentration within a given time. Although our model was not quite faithful to reality, especially with its perfect mixing assumption, it does give a rough idea of what to expect for a small amount of work.

Now we will allow for the incoming and outgoing rates to be different. We start with the general balance equation

$$\Delta M = C_I(t)V_I\Delta t - C(t)V_O\Delta t.$$

To see the concentration in it, we need to divide by the current volume  $V(t)$ . Obviously, if there is volume  $v_0$  in the tank at the start, then after time  $t$  it will be  $V(t) = V + V_I t - V_O t$ . We will not

worry now that this may become negative under some circumstances, divide the ballance equation by it instead.

$$\frac{\Delta C}{\Delta t} = C_I(t) \frac{V_I}{V + (V_I - V_O)t} - C(t) \frac{V_O}{V + (V_I - V_O)t}.$$

Then the corresponding differential equation reads

$$\begin{aligned} C'(t) &= C_I(t) \frac{V_I}{V + (V_I - V_O)t} - C(t) \frac{V_O}{V + (V_I - V_O)t} \\ \implies C' + C \frac{V_O}{V + (V_I - V_O)t} &= C_I(t) \frac{V_I}{V + (V_I - V_O)t}. \end{aligned}$$

This is no longer a separable differential equation, which is a bad news. The good news is that this equation is linear, and thus it can be solved using the method of variation.

We start with the homogeneous equation, and for convenience we will denote  $V_{OR} = \frac{V_O}{V_I - V_O}$ . Note that this assumes that  $V_I \neq V_O$ , otherwise  $V_{OR}$  does not make sense.

$$\begin{aligned} C' + C \frac{V_O}{V + (V_I - V_O)t} = 0 &\implies \frac{dC}{dt} = -C \frac{V_O}{V + (V_I - V_O)t} \\ \implies \int \frac{dC}{C} = - \int \frac{V_O}{V + (V_I - V_O)t} dt & \\ \implies \ln |C| = - \frac{V_O}{V_I - V_O} \ln |V + (V_I - V_O)t| + c & \\ \implies \ln |C| = c + \ln(|V + (V_I - V_O)t|^{-V_{OR}}) & \\ \implies C = \pm e^c |V + (V_I - V_O)t|^{-V_{OR}} & \\ \implies C_h(t) = \frac{D}{|V + (V_I - V_O)t|^{V_{OR}}}. & \end{aligned}$$

Since negative volume is impossible in real life, we can ignore the absolute value in the denominator.

Now we pass to the variation step and assume that  $D = D(x)$ . We obtain the equation

$$\begin{aligned} \frac{D'}{(V + (V_I - V_O)t)^{V_{OR}}} &= C_I(t) \frac{V_I}{V + (V_I - V_O)t} \\ \implies D' &= C_I(t) V_I (V + (V_I - V_O)t)^{V_{OR}-1}. \end{aligned}$$

In the case of variable incoming concentration we would know the function  $C_I(t)$  and perhaps the integral can be evaluated.

We do not want to stop yet, so we will assume that  $C_I$  is a constant and can continue:

$$\begin{aligned} D(t) &= \int C_I V_I (V + (V_I - V_O)t)^{V_{OR}-1} dt = C_I \frac{V_I}{V_{OR}(V_I - V_O)} (V + (V_I - V_O)t)^{V_{OR}} + c \\ \implies D(t) &= C_I \frac{V_I}{V_O} (V + (V_I - V_O)t)^{V_{OR}} + c. \end{aligned}$$

We thus obtain a general solution to our problem:

$$\begin{aligned} C(t) &= \frac{D(t)}{(V + (V_I - V_O)t)^{V_{OR}}} \\ &= C_I \frac{V_I}{V_O} + \frac{c}{(V + (V_I - V_O)t)^{V_{OR}}}. \end{aligned}$$

We deduced it under the assumption that  $V_O \neq V_I$ , and we now also see that we need  $V_O > 0$  to be true. The region of validity depends on the sign of  $V_I - V_O$ . If  $V_I > V_O$ , then the volume goes to infinity and theoretically, the solution exists for  $t \geq 0$ . If  $V_I < V_O$ , then the solution exists for  $t \leq \frac{V}{V_O - V_I}$ , then the tank becomes empty.

How does this solution behave? If  $V_I > v_o$ , then we can ask about the behaviour at infinity. We have  $V(t) \rightarrow \infty$  and we can see this volume in the denominator of expression on the right.

Moreover, for  $V_I > V_O$  we have  $V_{OR} > 0$ , so the fraction on the right has the form  $\frac{c}{\infty} \rightarrow 0$  and consequently,  $C(t) \rightarrow C_I \frac{V_I}{V_O}$ .

If  $V_I < V_O$ , then as  $t$  approaches its largest permitted value  $\frac{V}{V_O - V_I}$ , the volume in the denominator tends to zero. However, the power  $V_{OR}$  is then negative, so the fraction has the form  $c \cdot 0 = 0$  and we again have  $C(t) \rightarrow C_I \frac{V_I}{V_O}$ . But this time this convergence happens as time approaches a certain fixed endtime  $t_e$ , at which moment the tank becomes empty and the concentration stops having any meaning. Quite an interesting behaviour.

## 10e. Building a pillar

Consider a concrete pillar of circular crosssection, where the radius changes with height. Its height is  $H$ , and there is a mass of weight  $M$  on the top. The strength of the material requires that the stress on each horizontal section does not exceed a certain limit value  $\sigma_0$ . We will also assume that the concrete has density  $\varrho$ .

To simplify the situation, we will ask that the stress is exactly  $\sigma_0$  throughout the pillar, and ask what is the shape of the pillar then.

For simplicity, we will first use the distance  $y$  from the top of the pillar as our variable, let  $r(y)$  be the radius of the pillar at the distance  $y$ .

When we consider some section at the distance  $y$ , we see that the force  $F(y)$  acting on it comes from the mass  $m$ , and also from the weight  $m(y)$  of the pillar above this section.

The volume above the section is given as

$$V(y) = \int_0^y A(y) dy = \int_0^y \pi r^2(t) dt,$$

and the mass is  $m(y) = V(y)\varrho$ . The total force acting on this crosssection is therefore

$$F(y) = \varrho m(y)g + mg = \varrho g \int_0^y \pi r^2(t) dt + mg.$$

This causes the stress

$$\frac{F(y)}{S} = \frac{F(y)}{\pi r^2(y)}$$

that is supposed to be  $\sigma_0$ , so  $\sigma_0 \pi r^2(y) = F(y)$ . We therefore obtain the integral equation

$$\pi \sigma_0 r^2(y) = \varrho g \int_0^y \pi r^2(t) dt + mg.$$

We differentiate:

$$2\pi \sigma_0 r(y)r'(y) = \varrho g \pi r^2(y).$$

Assuming that  $r(y) > 0$  we divide and obtain

$$2\sigma_0 r' = \frac{\varrho g}{2\sigma_0} r.$$

This is an exponential equation with a general solution

$$r(y) = C e^{\frac{\varrho g}{2\sigma_0} y}.$$

In order to determine  $C$ , we find the initial value. When  $y = 0$ , then only the mass  $m$  is acting on the pillar, so

$$\sigma_0 \pi r(0)^2 = mg \implies r(0) = \sqrt{\frac{mg}{\sigma_0 \pi}}.$$

But also  $r(0) = Ce^0 = C$ , so we have  $C$ . We can write

$$r(y) = \sqrt{\frac{mg}{\sigma_0\pi}} e^{\frac{gg}{2\sigma_0}y}.$$

Since it is more natural to measure height of the pillar from the bottom, we can write

$$r(h) = \sqrt{\frac{mg}{\sigma_0\pi}} e^{\frac{gg}{2\sigma_0}(H-h)}, \quad h \in [0, H].$$

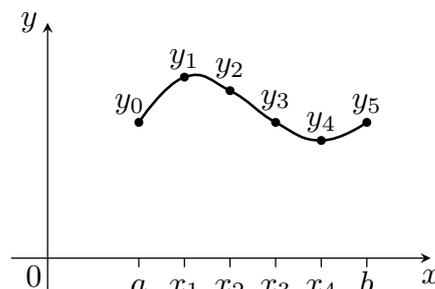
## 11. Euler method (introduction to numerical view of ODEs)

We learned some useful methods for solving first order differential equations, but they depended heavily on the form of these equations and then on our ability to integrate. Thus there is a good reason to ask what to do when we encounter an equation that we cannot handle.

In this chapter we will try to find some answers using a computer. What kind of an answer can we expect? Definitely not a formula, calculations produce numbers. (Yes, there are computer algebra programs that can also work with expressions, but they use algorithms programmed by humans, so if humans cannot solve an equation analytically, neither can a computer.)

This obviously means that we cannot expect a general solution, which narrows down our ambitions. We will use numerical calculations to get the best possible information about a particular solution of some initial value problem. What kind of information? A function is essentially a device that provides values whenever we need them. So the natural idea is that we try to find values of the solution at many points in order to get some idea how it looks. Since things are usually easier when they are regular, it inspires the following approach.

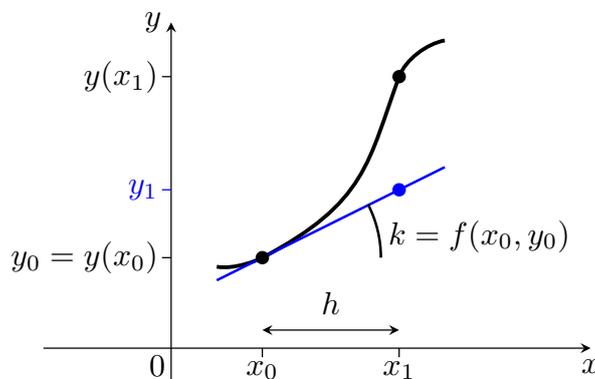
Consider an initial value problem (a differential equation with appropriate initial conditions at time  $x_0$ ) that has a particular solution  $y(x)$  on some interval  $I = [x_0, x_0 + T]$ . We choose  $n$  and divide the interval  $I$  into  $n$  equal parts, obtaining the partition  $x_0 < x_1 < \dots < x_n$  given by  $x_i = x_0 + ih$ , where  $h = \frac{T}{n}$  is the step of our procedure. We want to find numbers  $y_0, y_1, \dots, y_n$  that are as close as possible to values  $y(x_i)$ .



In this chapter we focus on differential equations of the form  $y' = f(x, y)$  with initial condition  $y(x_0) = y_0$ . Then obviously the best choice is to take  $y_0$  as the starting point of our approximation, but what next?

Note that thanks to the differential equation we know something about our solution at time  $x_0$ , namely that it has derivative  $k = y'(x_0) = f(x_0, y_0)$ . Thus we can construct the tangent line to the graph of  $y$  at  $x_0$ , its equation is  $y = y_0 + k(x - x_0)$ . If the step  $h$  is small, then we can hope that as the variable moves from  $x_0$  to  $x_1 = x_0 + h$ , the difference between  $y(x)$  and the tangent line will not be too large, so a good approximation of  $y(x_1)$  can be obtained by taking

$$y_1 = y_0 + k(x_1 - x_0) = y_0 + f(x_0, y_0) \cdot h.$$

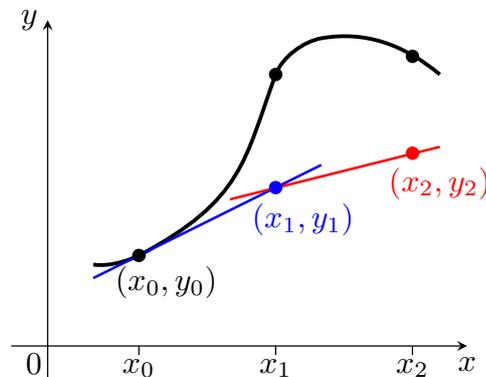


In this way we move to the point  $(x_1, y_1)$ .

What next? If we were at the point  $(x_1, y(x_1))$  of our solution, we could just repeat the above step. We would find the slope of the tangent line there,  $y'(x_1) = f(x_1, y(x_1))$ , and use it to approximate what happens as we pass to  $x_2 = x_1 + h$ . Just like above, we arrive at the approximate value

$$y(x_1) + f(x_1, y(x_1)) \cdot h$$

for  $y(x_2)$ . However, we do not know  $y(x_1)$ . On the other hand, we have its approximation  $y_1$  and we can hope that if we use it as replacement, the error will not be too large. We obtain the formula for the next approximating point,



$$y_2 = y_1 + f(x_1, y_1) \cdot h.$$

Note how this formula compares with the previous one. Now we can repeat these “jumps” along tangent lines until we reach  $x_n = x_0 + T$ .

The procedure we outlined here is natural, and it is the basis for most numerical methods for solving initial value problems of the form  $y' = f(x, y)$ ,  $y(x_0) = y_0$ .

**Algorithm 11.1.**

⟨Euler (forward) formula⟩

Given: ODE  $y' = f(x, y)$  on  $[x_0, x_0 + T]$ , initial condition  $y_0$ , and  $n \in \mathbb{N}$ .

- 0. Set  $h = \frac{T}{n}$ .
  - 1.  $x_0$  and  $y_0$  are given.
  - 2. For  $i = 0, \dots, n - 1$  set  $x_{i+1} = x_i + h$  and  $y_{i+1} = y_i + f(x_i, y_i)h$ .
- △

Note that we can actually prepare all  $x_i$  already in the initialization step by setting  $x_i = x_0 + hi$ . This is definitely a legitimate approach. We wrote our algorithm in this way to make it comparable with adaptive algorithms, where at each step  $i$  we are allowed to change the step size  $h$ . It also reminds us that we really move by stages, at each step we start from some point  $(x_i, y_i)$  and do not care what happened before (and do not know what will come afterwards).

**Example 11.a:** Consider the problem  $y' = x - \frac{1}{5}y^2$ ,  $y(1) = 4$ , on the interval  $[1, 7]$ .

We will use the Euler method to approximate its solution with  $n = 3$ . We have the step size  $h = 2$ , partition  $\{x_0 = 1, x_1 = 3, x_2 = 5, x_3 = 7\}$  and we denote  $f(x, y) = x - \frac{1}{5}y^2$ . We have our first point,  $(x_0, y_0) = (1, 4)$ .

$y_1$ : The slope at  $x_0 = 1$  is  $k = y'(1) = f(1, 4) = -\frac{11}{5}$ . Thus our best guess is

$$y_1 = y_0 + hk = 4 - \frac{11}{5} \cdot 2 = -\frac{2}{5} = -0.4.$$

We have our second point  $(x_1, y_1) = (3, -0.4)$ .

$y_2$ : The slope at  $x_1 = 3$  is  $k = y'(3) = f(3, y(3))$ . Since we do not know  $y(3)$ , we will use our guess  $y_1$  instead. We take  $k \approx f(3, -\frac{2}{5}) = \frac{371}{125}$  and obtain

$$y_2 = y_1 + hk = -\frac{2}{5} + \frac{371}{125} \cdot 2 = \frac{692}{125} = 5.536.$$

We obtained another point for our approximation,  $(x_2, y_2) = (5, 5.536)$ .

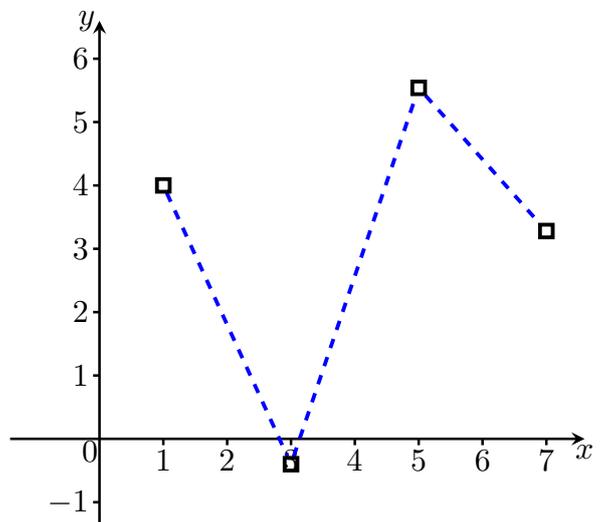
$y_3$ : The slope at  $x_2 = 5$  is  $k = y'(5) = f(5, y(5))$ , we use  $k \approx f(5, \frac{692}{125}) = \frac{-88239}{5 \cdot 125^2}$  instead and obtain

$$y_3 = y_2 + hk = \frac{692}{125} - \frac{88239}{5 \cdot 125^2} \cdot 2 = \frac{256022}{5 \cdot 125^2} = 3.2770816.$$

The result of our work can be expressed for instance in the following table.

$x_i$ :	1	3	5	7
$y_i$ :	4	-0.4	5.536	3.277...

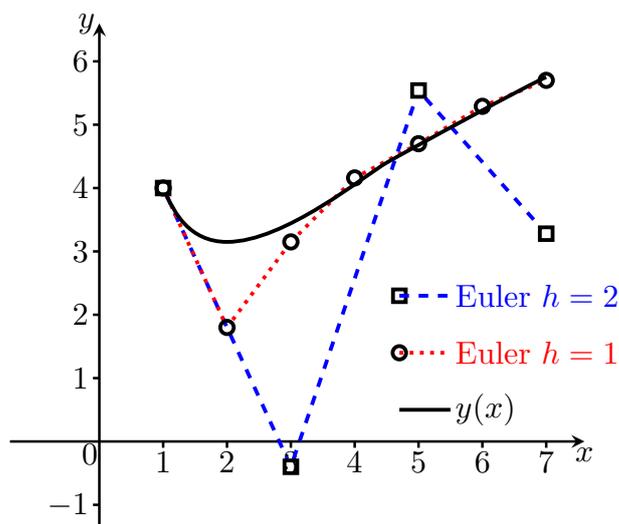
We see a picture on the right. We connected the approximating points with straight segments, but they are there just to make things nicer, we make no claim that they are somehow related to the graph of the solution  $y$ . How good is this approximation? Frankly, most likely very bad. The step size  $h = 2$  is huge; if we construct the tangent line at some point and then move along it that far, your typical function would be miles away by that time.



We actually do know the true solution to our problem and we plotted it on the right for comparison. As expected, our approximation is not good at all. But we do not want you to throw the book away, so we also included the outcome of the Euler method with a smaller step size  $h = 1$  obtained by taking  $n = 6$ . Actually,  $h = 1$  is still uncomfortably large, but the approximation seems surprisingly good. It seems there is some hope after all.

One would guess that by taking larger and larger  $n$  (and consequently smaller and smaller  $h$ ) we get increasingly better approximations. We will explore this idea in the next section.

△



We will shortly meet methods that provide approximations of better quality. This does not mean that looking at the Euler method is pointless. Although it is very basic, it does have some practical advantages. It is very simple to program and it runs fast, in particular because we call the function  $f$  only once to determine  $y_i$ , which can be a big time saver if  $f$  takes long to evaluate. People therefore often use this as a “starter method” to provide a base estimate for the solution which is then improved using more complicated methods. Second, it is intuitive, and its simplicity provides a good environment for learning concepts related to numerical solutions of ODEs.

Which is exactly what we are going to do now. We will outline notions and questions related to a certain group of methods for solving ODEs numerically. The Euler method will serve as a nice exhibit A on which we can show it off and test. So what kind of a method does the Euler formula represent?

First, this method is **iterative**, meaning that we go from one point to another, and use knowledge of what was done before to make the next step. This is a very popular approach not just for solving ODEs and we will see it repeatedly in this book. There are also other approaches, for instance it is possible to construct all points  $y_i$  simultaneously, but we will not look at them now.

An important distinction for iterative methods is how far back they look. Later on we will look (briefly) at multistep methods that look further back, but our main interest in this book lies with **one-step methods** of which the Euler method is a prime example. One-step methods construct  $y_{i+1}$  by looking back only by one step, to  $x_i$  and  $y_i$ . We can see our method as a procedure  $\Phi$  that produces a number  $y_{i+1}$ :

$$(x_i, y_i) \xrightarrow{\Phi} y_{i+1}.$$

What information is actually used? Apart from the point  $(x_i, y_i)$  we also need to know the step size  $h$  and the function  $f$  coming from the given ODE. Thus the general setup we will study now can be expressed in a general algorithm whose main (iterative) step goes as follows:

$$2. \quad x_{i+1} = x_i + h, \quad y_{i+1} = \Phi_f(x_i, y_i, h).$$

For the Euler method, the function  $\Phi_f$  is given by the formula

$$\Phi_f(x_i, y_i, h) = y_i + f(x_i, y_i) \cdot h.$$

In general,  $\Phi_f$  need not be a formula, we accept any well-defined procedure, for instance several calculations that have to be performed before  $y_{i+1}$  is determined.

One important aspect of one-step methods is that they are local. Since they only use the knowledge of  $(x_i, y_i)$ , they have no idea of what happened before. In particular a one-step method does not care how many steps were actually made before coming to  $(x_i, y_i)$ , so working with  $(x_{13}, y_{13})$  or  $(x_{31}, y_{31})$  makes no difference as long as the actual numbers agree. It is therefore more proper to describe the basic step  $\Psi_f$  without referring to any indexing, we should see it as

$\Psi_f : (x^*, y^*) \mapsto \Psi_f(x^*, y^*, h)$ . For instance, for the Euler method we have

$$\Phi_f(x^*, y^*, h) = y^* + f(x^*, y^*) \cdot h.$$

This has an interesting consequence. Since such a one-step method does not know the past, it also does not know which solution we are actually trying to approximate, nor how far we strayed from it before getting to some  $(x_i, y_i)$ . Such a method therefore does the best it can, namely it tries to approximate the solution that passes through  $(x_i, y_i)$ . Therefore, as we go from step to step, we are in fact each time trying to approximate a different solution. This observation will come handy soon.

## 11a. Errors for one-step methods

Assume that we have a method given by the general scheme with  $\Phi_f$  as above. Imagine that we obtained some approximation of a solution to a certain initial value problem. What is the error of this result caused by the method itself, that is, we assume precise calculations in our formulas?

We start with a general definition that is not concerned with differential equations, just with a general problem of approximating function at certain points  $x_i$ .

### Definition 11a.1.

Let  $y(x)$  be some function on an interval  $[x_0, x_0 + T]$ , let  $\{x_0, x_1, \dots, x_n\}$  be some partition of this interval.

For an approximation  $\{y_0, y_1, \dots, y_n\}$  of  $y$  we define the **global error** as

$$E_n = \max\{|y(x_i) - y_i| : i = 0, 1, \dots, n\}.$$

The picture is fairly obvious, we look at the largest (vertical) distance between the approximating points and the graph. Obviously, we appreciate if the global error is small.

Now we apply this idea to approximations of solutions of ODEs obtained by numerical methods. We would like to see the following: Given an initial value problem, we construct a sequence of approximations with  $n$  increasing to infinity, and we want their global errors to go to zero.

However, it does not make much sense demanding that this happens for all differential equations, in particular because some of them need not have any solution at all and then there is nothing to compare with. Thus we will only consider equations where the existence of a solution is guaranteed, and theorem 6a.2 offers a way to recognize such equations.

### Definition 11a.2.

Consider some method for solving initial value problems

$$y' = f(x, y(x)), \quad y(x_0) = y_0$$

that, for a given  $T > 0$  and  $n \in \mathbb{N}$ , creates approximations

$Y_n = \{(x_0, y_0), (x_0, y_1), \dots, (x_n, y_n)\}$  for the solution on interval  $[x_0, x_0 + T]$ .

This method is called **convergent** if the following is true:

For every initial value problem with a solution on an interval  $I$ , where  $f$  is Lipschitz in its second variable on some rectangle that includes this solution, the corresponding global errors  $E_n$  of approximations  $Y_n$  satisfy  $E_n \rightarrow 0$  as  $n \rightarrow \infty$ .

Methods without this property are quite useless, and as expected, all methods we will meet in this book are convergent. But we want more. We would like to know how fast  $E_n$  goes to zero, so that we can compare various methods. This calls for analysis of errors of individual estimates  $y_i$ , and this is quite complicated. In particular, note that our one-step methods work locally. On the other hand, in order to determine error at some  $x_{i+1}$ , that is, to find  $y(x_{i+1}) - y_{i+1}$ , we would

have to take into account how much our approximations strayed away from the true solution even before getting to  $x_i$ .

Thus we do something that we already saw when approximating integrals: We first look at local error in isolation. This is especially suitable to one-step methods. Since the workings of a one-step method represented by the mapping  $\Psi_f$  depends on  $f$ , we will define local error separately for individual differential equations.

**Definition 11a.3.**

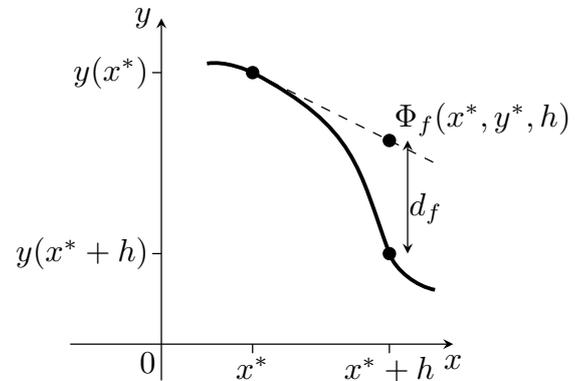
Consider a one-step method  $\Phi_f$  for solving initial value problems of the form  $y' = f(x, y(x))$ ,  $y(x_0) = y_0$  with iteration step  $x_{i+1} = x_i + h$ ,  $y_{i+1} = \Phi_f(x_i, y_i, h)$ . Given an ODE  $y' = f(x, y)$ , we define the **local error** of the method as follows: For all step sizes  $h > 0$  and  $x^*, y^* \in \mathbb{R}$  such that the IVP

$$y' = f(x, y), y(x^*) = y^*$$

has a solution  $y(x)$  on  $[x^*, x^* + h]$  we set

$$d_f(x^*, y^*, h) = y(x^* + h) - \Phi_f(x^*, y^*, h).$$

Consider now the situation that we found some approximations  $(x_i, y_i)$  of a certain solution  $y$  on an interval  $I$ . When we look at the local errors at each  $x_i$  (we take them as  $x^*$  in the definition), what do they tell us? In the definition of the local error we always compare with the solution that passes precisely through  $(x_i, y_i)$ , but this solution is most likely different from the actual solution  $y(x)$ , because it is very likely that by the time we got to  $(x_i, y_i)$ , the approximations  $y_i$  are no longer the same as  $y(x_i)$ . Thus the local error indeed only measures the contribution of the  $i$ th step to the overall error, ignoring the errors made previously.



We note that the local error may not be defined for some ODE and certain values  $(x^*, y^*)$ , in case there is no solution. Thus we prefer to apply the definition only to differential equations for which the existence of solutions is guaranteed, typically using the Lipschitz condition on  $f$ .

How does it work for the Euler method? The formula for  $\Phi_f$  is

$$y_{i+1} = y_i + f(x_i, y_i) \cdot h.$$

Passing to the setting of the local error, we are looking at the difference

$$y(x^* + h) - (y^* + f(x^*, y^*)h),$$

where the solution  $y$  passes through  $(x^*, y^*)$ . To determine the difference we use our popular tool, the Taylor expansion.

$$y(x^* + h) = y(x^*) + y'(x^*)h + \frac{1}{2}y''(x^*)h^2 + O(h^3).$$

Since  $y$  is supposed to solve the given ODE, we have  $y'(x^*) = f(x^*, y(x^*))$ , that is,

$$y(x^* + h) = y(x^*) + hf(x^*, y(x^*)) + \frac{1}{2}y''(x^*)h^2 + O(h^3).$$

By our assumption on initial condition,  $y^* = y(x^*)$  and thus

$$y(x^* + h) = y^* + hf(x^*, y^*) + \frac{1}{2}y''(x^*)h^2 + O(h^3).$$

By a remarkable coincidence, the first two terms on the right are exactly the Euler formula, so we have

$$d_f(x^*, y^*, h) = y(x^* + h) - (y^* + f(x^*, y^*)h) = \frac{1}{2}y''(x^*)h^2 + O(h^3).$$

It is also possible to use Mean value theorem and conclude that there must be some  $\xi$  between  $x_0$  and  $x_0 + h$  si that

$$d_f(x^*, y^*, h) = y(x^* + h) - (y^* + f(x^*, y^*)h) = \frac{1}{2}y''(\xi)h^2.$$

We see that the error is of order  $h$ , but the constant depends on the (unknown) solution  $y$ , which is not very pleasant. If we are to pass from local errors to the global error, we need to be able to control it.

To see the basic idea, let's imagine how this goes. To get from  $x_0$  to  $x_n = x_0 + T$  we have to make  $n$  steps. At each  $x_i$  there is some error  $e_i = y(x_i) - y_i$ , and the global error  $E_n$  is the maximum of  $|e_i|$ .

We can imagine that as we move from  $x_i$  to  $x_{i+1}$ , we would be separated from the solution by the local error  $d_i$  if we started at  $y(x_i)$ . However, we started at  $y_i$  instead, which represents shift by  $e_i$ . This would suggest the formula

$$e_{i+1} = e_i + d_i.$$

If this is correct, then by induction we obtain  $e_i = \sum_{j < i} d_j$ , therefore

$$|e_i| \leq \sum |d_j|. \quad \text{Thus we would get an estimate for the global error } |E_n| \leq \sum |d_j|.$$

If we have an estimate with power  $h^2$  for the local error as in the Euler method, we could estimate like this:

$$|E_n| \leq \sum_{i=0}^{n-1} |d_f(x_i, y_i, h)| = \sum_{i=0}^{n-1} (C_i h^2 + O(h^3)).$$

If we want to go on, we need  $C_i$  to be all equal, but since they are given by the formula  $\frac{1}{2}y''(x_i)$ , this is not to be expected. But we could hope for a common upper bound  $C$ . Then we would have

$$|E_n| \leq C \cdot n \cdot h^2 + n \cdot O(h^3).$$

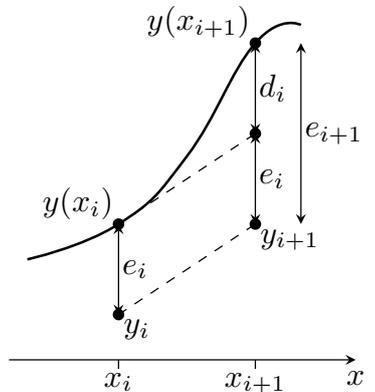
Since  $n = \frac{T}{h}$ , we get

$$|E_n| \leq CT \cdot \frac{1}{h} \cdot h^2 + \frac{T}{h} O(h^3) = CTh + O(h^2).$$

This is interesting: Just like we saw it for numerical integration, we lost one power of  $h$  when passing from local to global error. Indeed, this is one of the basic principles of numerical mathematics.

Unfortunately, things are not that easy, because we did not address one important step properly. We will get to it later, for now it is enough to know that the idea is essentially correct and the procedure can be fixed. Thus we can use it as an inspiration in forming the right definitions.

First, we are interested in controlling the global error, preferably using some power  $h^p$ , and we see that in order to do that we would need some control over local errors with power  $h^{p+1}$ . Second, it shows us that when estimating local errors, we also need some control over the constants  $C$  in  $Ch^{p+1}$ .



**Definition 11a.4.**

Consider a one-step method  $\Phi_f$  for solving initial value problems of the form  $y' = f(x, y(x)), y(x_0) = y_0$  with iteration step  $x_{i+1} = x_i + h, y_{i+1} = \Phi_f(x_i, y_i, h)$ . We say that the method is of **order**  $p$ , or that it has error of order  $p$ , if the following is true: For every differential equation  $y' = f(x, y)$  and rectangle  $I \times J$  such that  $f$  is Lipschitz with respect to  $y$  in  $I \times J$  and sufficiently smooth there is some  $C > 0$  so that

$$|d_f(x^*, y^*, h)| \leq Ch^{p+1}.$$

That is, a method is of order  $p$  if its local error is  $O(h^{p+1})$ , and we actually have a global control over the constant in the “big  $O$ ”. Note that we are not testing the order estimate against all differential equations, but only some of them. Typically, higher order estimates make sense only if we also assume that the solutions of our ODEs have derivatives of certain order, and this can be guaranteed by assuming that the function  $f$  itself is sufficiently smooth. Roughly speaking, in order to get order  $p$  we usually also need to demand that the functions  $f$  in the ODEs we consider have derivatives of order  $p$ .

Now we will use our observations about the Euler method above to determine its order.

**Theorem 11a.5.**

The Euler method is of order 1 with respect to differential equations  $y' = f(x, y)$  such that  $f$  is differentiable on its domain and its partial derivatives are bounded on bounded rectangles.

**Proof:** First some general observations. Consider some bounded rectangle  $I \times J$  on which  $f$  has partial derivatives bounded by some  $K$ . Then it must be  $K$ -Lipschitz there, hence for every point of  $I \times J$  there is some solution passing through it. Moreover,  $f$  must be also bounded on  $I \times J$ , say by some  $M$ .

Consider one such solution  $y(x)$  existing inside  $I \times J$ . Then  $y' = f(x, y(x))$  and thus  $y'(x)$  must be bounded by  $M$  as well. We differentiate,

$$y''(x) = \frac{\partial f}{\partial x}(x, y(x)) + \frac{\partial f}{\partial y}(x, y(x))y'(x),$$

therefore  $|y''(x)| \leq K + KM$ , that is, there is a universal bound on second derivatives of all solutions passing through  $I \times J$ .

Now we return to the formula for the local error of the Euler method derived above. We have

$$d_f(x^*, y^*, h) = \frac{1}{2}y''(\xi)h^2$$

for some  $\xi \in I$ , and therefore

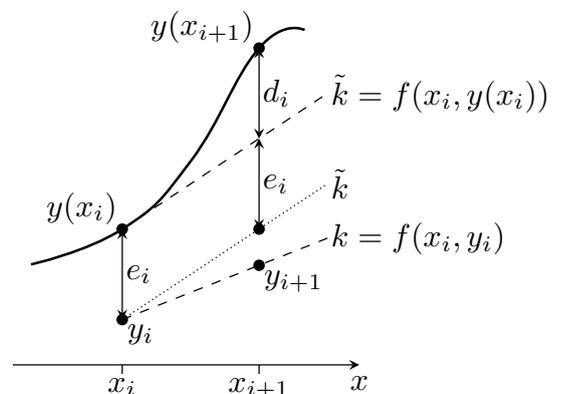
$$|d_f(x^*, y^*, h)| \leq \frac{1}{2}(K + KM)h^2.$$

This shows that there is a common constant  $C$  so that  $|d_f(x^*, y^*, h)| \leq Ch^{1+1}$  on  $I \times J$ , exactly as claimed.  $\square$

Now we return to the problem of global error. We will look closer at the situation explored above when we pass from  $x_i$  to  $x_{i+1}$ .

We start as before and observe that if we started off from  $(x_i, y(x_i))$ , we would arrive at some point  $y^* = \Phi_f(x_i, y(x_i), h)$  whose distance from  $y(x_{i+1})$  is exactly the local error  $d_i = d_f(x_i, y(x_i), h)$ . But we start at a point  $(x_i, y_i)$  that is shifted down by  $e_i$ . If we followed the same slope  $\tilde{k}$  as before, which for the Euler method means the slope given by  $f(x_i, y(x_i))$ , then the new point would indeed have error  $e_i + d_i$ .

However (and this is where the story above got it wrong), we actually follow a different slope  $k$  and end up in the point  $y_{i+1} = \Phi_f(x_i, y_i, h)$ . For the Euler method it is the slope  $f(x_i, y_i)$ . We do not know whether it is larger or smaller than  $k$ , but we have to be ready to face the worst, so we have to admit that  $e_{i+1}$  could actually be larger than  $|e_i| + |d_i|$ .



Thus the story we had above about global error being bounded by an appropriate sum of local errors is not correct. The good news is that the deviations as we see them in the picture can be controlled, and thus one can really derive an estimate with  $h^p$  for the global error, just the constant will be larger. We will prove a very general statement, but we will understand it better if we look at it in a more specific setting.

Many of the methods in this book will follow the basic idea of the Euler method, that is, we will move from the given point along some line, and the magic will be hidden in the selection of the slope. Thus we can now focus on methods with the main step of the form

$$\Phi_f(x^*, y^*, h) = y^* + h \cdot k_f(x^*, y^*, h).$$

In the picture above we decomposed the error  $e_{i+1}$  into two segments, one of them being  $d_i$ . The other segment is the difference

$$\begin{aligned} \Phi_f(x_i, y(x_i), h) - \Phi_f(x_i, y_i, h) &= (y(x_i) + hk_f(x_i, y(x_i), h)) - (y_i + hk_f(x_i, y_i, h)) \\ &= (y(x_i) - y_i) + h(k_f(x_i, y(x_i), h) - k_f(x_i, y_i, h)) \\ &= e_i + h(k_f(x_i, y(x_i), h) - k_f(x_i, y_i, h)). \end{aligned}$$

If we want to control the extra term, we need to have information on how  $k_f$  changes when we change  $y$  in it. For the Euler method we will show that  $k_f$  is actually Lipschitz in  $y$ , which allows for the following estimate:

$$\begin{aligned} |\Phi_f(x_i, y(x_i), h) - \Phi_f(x_i, y_i, h)| &\leq |e_i| + h|k_f(x_i, y(x_i), h) - k_f(x_i, y_i, h)| \\ &\leq |e_i| + hK|y(x_i) - y_i| = (1 + Kh)|e_i|. \end{aligned}$$

This type of estimate turns out to be crucial for our ability to pass from local to global error. We will therefore include it as an assumption on  $\Phi$  in our statement.

**Theorem 11a.6.**

Consider a one-step method  $\Phi_f$  for solving initial value problems of the form  $y' = f(x, y(x))$ ,  $y(x_0) = y_0$  with iteration step  $x_{i+1} = x_i + h$ ,  $y_{i+1} = \Phi_f(x_i, y_i, h)$ . Assume that for every differential equation  $y' = f(x, y)$  and rectangle  $I \times J$  such that  $f$  is Lipschitz with respect to  $y$  in  $I \times J$  and sufficiently smooth, there is  $K_\Phi$  such that for all  $x \in I$ ,  $h > 0$  and  $y_1, y_2 \in J$ ,

$$|\Phi_f(x, y_2, h) - \Phi_f(x, y_1, h)| \leq (1 + K_\Phi h)|y_2 - y_1|.$$

Then if  $\Phi_f$  is of order  $p$ , then for every differential equation  $y' = f(x, y)$  and rectangle  $I \times J$  such that  $f$  is Lipschitz with respect to  $y$  in  $I \times J$  there is a constant  $D$  so that for every solution to the given differential equation passing through  $I \times J$ , the global errors of approximations produced by the method  $\Phi_f$  satisfy  $|E_h| \leq Dh^p$ .

In short, the global error is indeed of order  $p$ . We preferred to give a more detailed statement, sometimes it is good to know that the constant in the  $h^p$  estimate does not depend on some factors, but depends on others.

From the estimate it immediately follows that when a method is of order  $p > 0$  and satisfies that special property (that we will discuss below), then it is automatically convergent.

**Proof:** Take some IVP as in the theorem and its solution  $y(x)$  on some interval  $[x_0, x_0 + T]$  that lies inside  $I \times J$ . Given  $n \in \mathbb{N}$ , we set the step size  $h = \frac{T}{n}$ , apply the method  $\Phi_f$  and obtain the approximation

$$\{(x_0, y_0), \dots, (x_n, y_n)\}.$$

Denote  $e_i = y(x_i) - y_i$ .

We start at a point  $(x_i, y_i)$  for  $i < n$ . We want to estimate the error  $e_i$ . We can write

$$\begin{aligned} e_{i+1} &= y(x_{i+1}) - \Phi_f(x_i, y_i, h) = y(x_{i+1}) - \Phi_f(x_i, y(x_i), h) + \Phi_f(x_i, y(x_i), h) - \Phi_f(x_i, y_i, h) \\ &= d_f(x_i, y(x_i), h) + \Phi_f(x_i, y(x_i), h) - \Phi_f(x_i, y_i, h). \end{aligned}$$

We now use our assumptions:

$$\begin{aligned} |e_{i+1}| &\leq |d_f(x_i, y(x_i), h)| + |\Phi_f(x_i, y(x_i), h) - \Phi_f(x_i, y_i, h)| \\ &\leq Ch^{p+1} + (1 + Kh)|y(x_i) - y_i| = Ch^{p+1} + (1 + Kh)|e_i|. \end{aligned}$$

□

## 11b. Numerical stability

In general, numerical stability means that the calculations do not enlarge errors that appear in it (see error propagation). In the field of differential equations there are specific definitions that capture this general idea in various ways. One rather popular requirement goes as follows:

Assume that some initial value problem has a bounded solution. Numerical stability means that if we change the initial condition by a small amount, then the solution provided by the method is also bounded.

A method is called numerically stable if it has this property for all IVPs with bounded solutions. However, this is really rare. We have a better chance to have a conditional stability, which means that for some IVPs we can prove that nice behaviour, but for others it does not work.

The Euler method is conditionally stable. Given a simpler differential equation, we can often find some  $\delta > 0$  so that if we stick with  $h < \delta$  then the resulting numerical solution does not stray far from the real solution.

This topic is rather involved and requires deeper analysis, so we will not really go into it, we just wanted to give the reader some idea of what people expect from methods. There is one aspect that we should see, however. In the algorithm we let  $x_i$  be defined recursively, at every step we move by  $h$ . This is the right way to think about it, and in fact it will later allow us to look at more general methods that change  $h$  on the fly, but obviously there is a danger that any error in  $h$  gets multiplied many many times, in fact there is a good reason to suspect that  $x_n$  will not be  $x_0 + T$ . So if we indeed do work with fixed  $h$  coming from some partition size  $n \in \mathbb{N}$ , we would be better off to create those  $x_i$  using the formula  $x_i = x_0 + ih$ .

## 11c. Implicit Euler method

Assume that the solution we are looking for is concave up. The the Euler method always underestimates the growth, which means a consistent bias, something we do not like. Here's an interesting idea. Imagine we have  $y_i$  and we want to estimate  $y_{i+1}$ . The standard Euler method uses the information about slope at  $x_i$  and makes a step forward. This is the reason why it is sometimes called the forward Euler method.

The **implicit Euler method** (or backward Euler method) introduces another approach. It tries to get from  $y_i$  to  $y_{i+1}$  using the slope at  $x_{i+1}$ . The trouble is that the slope is calculated as  $k = y'(x_{i+1}) = f(x_{i+1}, y(x_{i+1}))$  which we usually approximate using  $f(x_{i+1}, y_{i+1})$ , but we do not know  $y_{i+1}$  at the moment, that's what we are trying to find! Such a circular reasoning often leads to implicit equations. Indeed, we want the number  $y_{i+1}$  to satisfy

$$y_{i+1} = y_i + f(x_{i+1}, y_{i+1})h.$$

If the function  $f$  collaborates, we might be able to solve this equation for  $y_{i+1}$ , obtaining a workable explicit scheme. However, this does not happen too often, so the implicit approach tends to be troublesome.

**Example 11c.a:** Consider the equation  $y' = x - \frac{1}{5}y^2$ .

The implicit Euler method leads to the formula

$$y_{i+1} = y_i + (x_i - \frac{1}{5}y_{i+1}^2)h \implies \frac{1}{5}y_{i+1}^2 h + y_{i+1} - (x_i h + y_i) = 0.$$

Unfortunately, a quadratic equation does not have a unique solution. There might be a way to decide which of the two solutions is the right one, but obviously this is too much trouble compared to the forward Euler approach.

△

For some problems there is a unique solution to the resulting implicit equation, but we cannot find it analytically (this is actually typical). Then people would use numerical methods for finding the solution. Now what do we get for our trouble?

The implicit Euler method is order 1. Moreover, just like the forward Euler method it has a systematic bias, just the other way around. This explains why people do not use it much. There are known specific problems for which the implicit Euler method works better.

We will show how the implicit Euler method works for one of the simplest differential equations, the exponential one.

**Example 11c.b:** Consider the equation  $y' = -0.5y$  with the initial condition  $y(0) = 1$ .

We will look for the solution on the interval  $[0, 4]$ , we try  $n = 4$ , that is,  $h = 1$ . Thus we will approximate the solution at points  $x_i = i$  for  $i = 0, 1, \dots, 4$ . Both methods start with  $y_0 = 1$

The (forward) Euler method yields the formula

$$y_{i+1} = y_i + f(y_i)h = y_i - 0.5y_i \cdot 1 = 0.5y_i.$$

The implicit Euler method leads to

$$y_{i+1} = y_i + f(y_{i+1})h = y_i - 0.5y_{i+1} \implies y_{i+1} = \frac{2}{3}y_i.$$

The picture shows how the two methods compare with the actual solution. We expected something like this.

△

There is an interesting idea. We have two candidates for the slope to use in our step forward. One is the slope at  $x_i$ , the other the slope at  $x_{i+1}$ ; one systematically overshoots, the other underestimates. Why don't we split the difference and take the average of both?

That's a very good idea and if it wasn't for the troublesome calculations involved in the implicit method, we would get a very neat method. As it is, we will save it for the next chapter, where we try a new approach.

## 11d. ODE's and integrals

Consider the problem  $y' = f(x, y)$ ,  $y(x_0) = y_0$  and the usual setup: We want a solution on the interval  $[x_0, x_0 + T]$ , we chose  $n \in \mathbb{N}$  and obtained a partition  $\{x_i\}$  with step  $h = \frac{T}{n}$ .

Now we choose  $i \in \{0, \dots, n-1\}$  and we integrate the given equation over the interval  $[x_{i-1}, x_i]$ .

$$\int_{x_i}^{x_{i+1}} y'(x) dx = \int_{x_i}^{x_{i+1}} f(x, y(x)) dx.$$

By the Newton-Leibniz formula, the left intrgal is  $y(x_{i+1}) - y(x_i)$ , so we obtain

$$y(x_{i+1}) = y(x_i) + \int_{x_i}^{x_{i+1}} f(x, y(x)) dx.$$

For  $y(x_i)$  we have an estimate, so we would like to set

$$y_{i+1} = y_i + \int_{x_i}^{x_{i+1}} f(x, y(x)) dx.$$

Since we cannot hope to evaluate the integral, we will use one of our favourite numerical methods for estimating it (chapter 5). We actually work here with just one panel. Note that the length of the integration interval is  $h$ .

If we estimate the using left rectangles, we get

$$y_{i+1} \approx y_i + f(x_i, y(x_i))h \approx y_i + f(x_i, y_i)h.$$

Remarkably enough, this is the Euler formula. We know the error over one panel for the rectangle method, it is  $O(h^2)$ , which fits.

The right rectangle method yields

$$y_{i+1} \approx y_i + f(x_i, y(x_{i+1}))h \approx y_i + f(x_i, y_{i+1})h.$$

This is the implicit Euler method, again with local error  $O(h^2)$ .

A better result was obtained by the trapezoid method.

$$y_{i+1} \approx y_i + \frac{1}{2}(f(x_i, y(x_i)) + f(x_i, y(x_{i+1})))h \approx y_i + \frac{1}{2}(f(x_i, y_i) + f(x_i, y_{i+1}))h.$$

This is the idea we had earlier with averaging left and right slope. The local error is  $O(h^3)$ , one better than for the Euler method.

We had another method of the same order and it required just one evaluation of  $f$ , which might be an advantage, it was the midpoint method.

$$y_{i+1} \approx y_i + f\left(\frac{1}{2}(x_i + x_{i+1}), y\left(\frac{1}{2}(x_i + x_{i+1})\right)\right)h.$$

This should also provide a method of order 2 (local error of asymptotic rate  $h^3$ ), but there is a problem here, the value of  $y$  at the midpoint  $\frac{1}{2}(x_i + x_{i+1})$  is not on our list  $\{y_i\}$  of known values of  $y$ . This means that we cannot even form an implicit equation, this is a dead end.

However, all these ideas will be the starting point for our next chapter.

## 12. Basic numerical methods for ODEs

To have it in one place we first review methods that we already know.

### Algorithm 12.1.

⟨Euler (forward) formula⟩

Given: ODE  $y' = f(x, y)$  on  $[x_0, x_0 + T]$ , initial condition  $y_0$ , and  $n \in \mathbb{N}$ ,

set  $h = \frac{T}{n}$ .

0.  $y_0$  is given.

1. For  $i = 0, \dots, n - 1$  set  $x_{i+1} = x_i + h$  and define  $y_{i+1} = y_i + f(x_i, y_i)h$ .

△

### Algorithm 12.2.

⟨implicit (backward) Euler formula⟩

Given: ODE  $y' = f(x, y)$  on  $[x_0, x_0 + T]$ , initial condition  $y_0$ , and  $n \in \mathbb{N}$ ,

set  $h = \frac{T}{n}$ .

0.  $y_0$  is given.

1. For  $i = 0, \dots, n - 1$  set  $x_{i+1} = x_i + h$  and define  $y_{i+1}$  by solving  $y_{i+1} = y_i + f(x_{i+1}, y_{i+1})h$  for it.

△

There are several approaches one can try to improve on the basic Euler method, here we will focus on one fruitful idea. As we noted in the previous chapter, we expect better approximation if we take some care to find a good slope for making the step forward. We observed that taking slopes at the endpoints of  $[x_i, x_{i+1}]$  yielded methods of order 1, but our final thoughts suggested that we might get a method of order 2 if we base the slope of our step on  $y'$  evaluated at midpoint. From the point of the last chapter that was the end, but now comes the time for a new idea.

To make our life easier, we will introduce the notation  $x_{i+1/2}$  for the point  $\frac{1}{2}(x_i + x_{i+1})$  in the middle of  $[x_i, x_{i+1}]$ . This makes sense, because we have  $x_{i+1} = x_i + 1 \cdot h$  and  $x_{i+1/2} = x_i + \frac{1}{2}h$ . Naturally, we denote by  $y_{i+1/2}$  our estimate for  $y(x_{i+1/2})$ . Now let's try the following line of reasoning.

1. To find the slope  $k_2$  at  $x_{i+1/2}$  we need to evaluate  $f(x_{i+1/2}, y(x_{i+1/2}))$ .
2. To evaluate that, we need to find an estimate for  $y(x_{i+1/2})$ .
3. When we wanted to estimate what happens to the right from  $x_i$ , we used approximation by the tangent line (Euler method) using the slope at  $x_i$ . The difference is that now we will move only half a step.

When we take it from 3. to 1., we get a blueprint for our next method. We will emphasise that our estimate of  $y(x_{i+1/2})$  is not a part of the output by putting an asterisk next to it.

### Algorithm 12.3.

⟨**RK2** (midpoint method, modified Euler formula, improved polygon)⟩

Given: ODE  $y' = f(x, y)$  on  $[x_0, x_0 + T]$ , initial condition  $y_0$ , and  $n \in \mathbb{N}$ ,

set  $h = \frac{T}{n}$ .

0.  $y_0$  is given.

1. For  $i = 0, \dots, n - 1$  set  $x_{i+1} = x_i + h$ . Then:

a) Estimate the slope  $y'(x_i)$ :  $k_1 = f(x_i, y_i)$ .

b) Estimate  $y(x_i + \frac{1}{2}h)$ :  $y_{i+1/2}^* = y_i + \frac{1}{2}k_1 \cdot h$ ,

then estimate the slope  $y'(x_i + \frac{1}{2}h)$ :  $k_2 = f(x_i + \frac{1}{2}h, y_{i+1/2}^*)$ .

c) Set  $x_{i+1} = x_i + h$  and define  $y_{i+1} = y_i + k_2h$ .

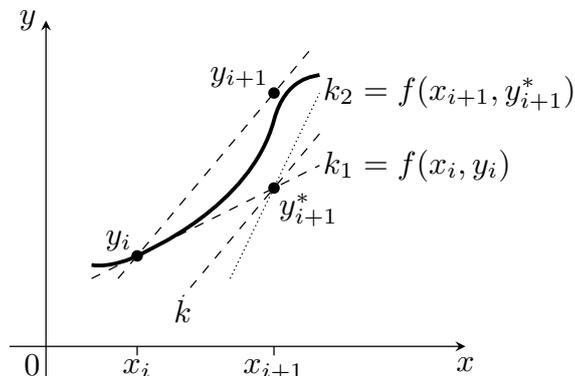
△

It works as we hoped it would.

**Fact 12.4.**

The midpoint (modified Euler) method is consistent and it is a method of order 2.

In a similar way we can make true our dream of averaging left and right slopes, the picture should help in understanding the idea. First we use the known slope  $k_1$  to get a reasonable approximation of  $y(x_{i+1})$ , we denote it  $y_{i+1}^*$  as it is just a supplementary information. Using this we get an approximation  $k_2$  for the slope at  $x_{i+1}$ . Taking the average  $k = \frac{1}{2}(k_1 + k_2)$  we hope to get an estimate for the proper slope that does not systematically overshoot or undershoot, and using this slope we get the approximation  $y_{i+1}$ .

**Algorithm 12.5.**

⟨Heun formula (improved Euler formula)⟩

Given: ODE  $y' = f(x, y)$  on  $[x_0, x_0 + T]$ , initial condition  $y_0$ , and  $n \in \mathbb{N}$ ,

set  $h = \frac{T}{n}$ .

0.  $y_0$  is given.

1. For  $i = 0, \dots, n - 1$  set  $x_{i+1} = x_i + h$ . Then:

a) Estimate the slope  $y'(x_i)$ :  $k_1 = f(x_i, y_i)$ .

b) Estimate  $y_{i+1}$ :  $y_{i+1}^* = y_i + k_1 h$ ,

then estimate the slope  $y'(x_{i+1})$ :  $k_2 = f(x_{i+1}, y_{i+1}^*)$ .

c) Set  $y_{i+1} = y_i + \frac{1}{2}(k_1 + k_2) \cdot h$ .

△

Note that we actually estimated the value  $y(x_{i+1})$  twice. First we did just the basic Euler step with the information we had on hand, and then we did the Euler step again, this time using updated information about slope. There are many methods that first do a rough guess and then try to improve upon it, they are called **predictor-corrector** methods.

The picture suggests that the average should work better than the forward slope or the backward slope. This is indeed true.

**Fact 12.6.**

The Heun method is consistent and it is a method of order 2.

These two methods suggest a general approach. We use Euler method to get preliminary information about  $y$ , we use it to find interesting slopes, we combine this information, perhaps use it to improve our estimates, we can repeat this process many times, we can sample at different points of the segment  $[x_i, x_{i+1}]$ . Eventually we combine slopes that we have at our disposal to form our best educated guess for the right slope to move us forward. If we set it up the right way, every slope we use raises the order of the method by one. One has to be careful to make the flow of information reasonable, with inner consistency, and two people worked out a general scheme.

**Definition 12.7.**

An explicit Runge-Kutta method for solving IVP  $y' = f(x, y)$  is given by fixing parameters

$$\begin{array}{c|cccc}
 0 & & & & \\
 c_2 & a_{21} & & & \\
 c_3 & a_{31} & a_{32} & & \\
 \vdots & \vdots & & \ddots & \\
 c_N & a_{N1} & a_{N2} & \cdots & a_{N,N-1} \\
 \hline
 & w_1 & w_2 & \cdots & w_{N-1} & w_N
 \end{array}$$

Here  $N$  is the number of steps,  $c_j$  defines nodes,  $a_{jl}$  forms the matrix of the method and  $w_j$  are weights.

When determining  $y_{i+1}$  using  $y_i$  we first estimate slopes at various points,

$$\begin{aligned}
 k_1 &= f(x_i, y_i), \\
 k_2 &= f(x_i + c_2 h, y_i + a_{21} k_1 h), \\
 k_3 &= f(x_i + c_3 h, y_i + (a_{31} k_1 + a_{32} k_2) h), \\
 &\vdots \\
 k_N &= f(x_i + c_N h, y_i + (a_{N1} k_1 + a_{N2} k_2 + \cdots + a_{N,N-1} k_{N-1}) h),
 \end{aligned}$$

these estimates are averaged to get the best slope  $k = \sum_{j=1}^N w_j k_j$  and then we set

$$y_{i+1} = y_i + k \cdot h.$$

We already saw one method based on this scheme, RK2. The most frequently used method for solving ODEs is the following one.

**Algorithm 12.8.**

⟨RK4⟩

Given: ODE  $y' = f(x, y)$  on  $[x_0, x_0 + T]$ , initial condition  $y_0$ , and  $n \in \mathbb{N}$ , set  $h = \frac{T}{n}$ .

0.  $y_0$  is given.

1. For  $i = 0, \dots, n-1$  set  $x_{i+1} = x_i + h$ . Then:

a) Estimate the slope  $y'(x_i)$ :  $k_1 = f(x_i, y_i)$ .

b) Estimate  $y(x_i + \frac{1}{2}h)$ :  $y_{i+1/2}^* = y_i + \frac{1}{2}k_1 h$

and then estimate the slope  $y'(x_i + \frac{1}{2}h)$ :  $k_2 = f(x_i + \frac{1}{2}h, y_{i+1/2}^*)$ .

c) Again (to improve?) estimate  $y(x_i + \frac{1}{2}h)$ :  $y_{i+1/2}^{**} = y_i + \frac{1}{2}k_2 h$

and then estimate the slope  $y'(x_i + \frac{1}{2}h)$ :  $k_3 = f(x_i + \frac{1}{2}h, y_{i+1/2}^{**})$ .

d) Estimate  $y(x_i + h)$ :  $y_{i+1}^* = y_i + k_3 h$

and then estimate the slope  $y'(x_{i+1})$ :  $k_4 = f(x_{i+1}, y_{i+1}^*)$ .

e) Set  $y_{i+1} = y_i + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)h$ .

△

This results ties in with something we saw earlier. Note that  $k_2$  and  $k_3$  both refer to slope at the midpoint. If we, just for the moment, denote by  $f_j$  some slope obtained at  $x_j$  by evaluating  $f$  at  $x_j$  and some approximation of  $y_j$ , we get the following:

$$y_{i+1} = y_i + \frac{1}{6}(f_i + 4f_{i+1/2} + f_{i+1}) \cdot h.$$

This is what we get if we use the Simpson formula to estimate the integral in the integral form of iterative methods, see section . The local error of Simpson's rule is  $O(h^5)$  and it does work for RK4 as well.

**Fact 12.9.**

The RK4 method is consistent and it is a method of order 4.

The problem with all numerical methods is that we have nice theoretical estimates for error that converge to zero, but when we run it for a particular  $n$ , those estimates are of no help. Thus it is of great advantage if we can somehow estimate the error we made using numerical methods. For the methods we have shown here there is a way to do this.

**Fact 12.10.**

Consider a grid  $x_0 < x_1 < \dots < x_N$  with step  $h > 0$  and approximations  $\{y_i\}$  of solution produced by the main method of order  $p$  and approximations  $\{z_i\}$  produced by a control method of order higher than  $p$ . Then the values  $|z_{i+1} - y_{i+1}|$  are a sensible estimate of local errors of the main method.

If we want an approximation of solution with global error  $\varepsilon > 0$ , then it is advisable to try the main method with step  $sh$ , where

$$s = \left( \frac{h\varepsilon}{\max |z_{i+1} - y_{i+1}|} \right)^{1/p}.$$

The idea that we can change the step  $h$  to guarantee that error does not get too large leads to an idea of

adaptive methods, where we determine  $h$  on the fly, depending how well the basic method we use works for our problem at that particular place. If  $f$  is “nice” somewhere, we can afford larger  $h$  to speed up the process. When  $f$  starts acting up, we take smaller  $h$  to get a better feeling for what is going on. The most popular method of this kind is based on Runge-Kutta methods.

**Algorithm 12.11.**

(RKF45, adaptive Runge-Kutta-Fehlberg method)

Given: ODE  $y' = f(x, y)$  on  $[x_0, x_0 + T]$ , initial condition  $y_0$ , initial step  $h_0 > 0$  and desired precision  $\varepsilon > 0$ .

0.  $y_0$  is given.

1. For  $i = 0, \dots, n - 1$ :

a) Evaluate  $k_1 = f(x_i, y_i)$ .

b) Evaluate  $k_2 = f(x_i + \frac{1}{4}h_i, y_i + \frac{1}{4}k_1h_i)$ .

c) Evaluate  $k_3 = f(x_i + \frac{3}{8}h_i, y_i + (\frac{3}{32}k_1 + \frac{9}{32}k_2)h_i)$ .

d) Evaluate  $k_4 = f(x_i + \frac{12}{13}h_i, y_i + (\frac{1932}{2197}k_1 - \frac{7200}{2197}k_2 + \frac{7296}{2197}k_3)h_i)$ .

e) Evaluate  $k_5 = f(x_i + h_i, y_i + (\frac{439}{216}k_1 - 8k_2 + \frac{3680}{513}k_3 - \frac{845}{4104}k_4)h_i)$ .

f) Evaluate  $k_6 = f(x_i + \frac{1}{2}h_i, y_i + (-\frac{8}{27}k_1 + 2k_2 - \frac{3544}{2565}k_3 + \frac{1859}{4104}k_4 - \frac{11}{40}k_5)h_i)$ .

g) Estimate  $y_{i+1} = y_i + (\frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4104}k_4 - \frac{1}{5}k_5)h_i$

and  $z_{i+1} = y_i + (\frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6)h_i$ .

If  $\frac{1}{h_i}|z_{i+1} - y_{i+1}| > \varepsilon$ , set  $s = \left( \frac{h\varepsilon}{|z_{i+1} - y_{i+1}|} \right)^{1/4}$  and redo the calculations starting from a) on with the step  $h_i = sh_i$ .

Otherwise set  $x_{i+1} = x_i + h_i$ ,  $h_{i+1} = h_i$  and go to the next cycle, that is, increase  $i$  by one etc.

△

### 13. More methods for solving ODEs

Runge-Kutta methods outlined in chapter 12 are a traditional gateway to numerical approach to ODEs and the high end ones (in particular RKF45) are good enough to be of practical use. However, there are also other approaches that lead to useful methods, in particular because there are types of problems that are known to be troublesome.

We will outline here several basic ideas. Truth be told, most of them truly shine in different settings (for instance with higher order differential equations), but we feel that the reader will take to them easily in the familiar setting of first order differential equations.

#### 13a. Taylor method

Consider a differential equation  $y' = f(x, y)$  and a point  $(x_0, y_0)$  that lies on some solution  $y(x)$ ; and alternative take is that  $y(x)$  is the solution determined by the initial condition  $y(x_0) = y_0$ .

When we asked what error we make if we move from  $y_0$  using the Euler method, we used the Taylor expansion of the solution  $y(x)$ ,

$$y(x_0 + h) = y(x_0) + y'(x_0)h + \frac{1}{2}y''(x_0)h^2 + \frac{1}{6}y'''(x_0)h^3 + \frac{1}{24}y''''(x_0)h^4 + \dots$$

We may decide to take the first two terms on the right as a reasonable approximation for  $y(x_0 + h)$ , which can be done because we know that  $y(x_0) = y_0$  and  $y'(x_0) = f(x_0, y_0)$ , this data is available. In this way we obtain the forward Euler method and we also see that the local error (the remaining part of the series on the right) is of order  $O(h^2)$  as  $h \rightarrow 0^+$ .

A natural way to obtain higher order methods is to use higher order polynomial for our approximation, not just a linear one. For instance, a second degree approximation would be

$$y(x_0) + y'(x_0)h + \frac{1}{2}y''(x_0)h^2 = y_0 + f(x_0, y_0)h + \frac{1}{2}y''(x_0)h^2.$$

In this way we would obtain local error  $O(h^3)$  and a method of order 2, assuming that we can work out the coefficient  $y''(x_0)$ . In a similar way we could obtain methods of arbitrarily high order.

Can we really work out those derivatives? Remember that we have a general formula  $y'(x) = f(x, y(x))$  that is valid on some interval  $I$ . We can differentiate both sides using appropriate rules, and we obtain a formula for  $y''$ . This can be done repeatedly.

$$\begin{aligned} y' &= f(x, y(x)), \\ y'' &= \frac{\partial f}{\partial x}(x, y(x)) + \frac{\partial f}{\partial y}(x, y(x))y'(x), \\ y''' &= \frac{\partial^2 f}{\partial x^2}(x, y(x)) + \frac{\partial^2 f}{\partial y \partial x}(x, y(x))y'(x) + \frac{\partial^2 f}{\partial x \partial y}(x, y(x))y'(x) + \frac{\partial^2 f}{\partial y^2}(x, y(x))[y'(x)]^2 + \frac{\partial f}{\partial y}(x, y(x))y''(x), \\ &\vdots \end{aligned}$$

Well, I've heard enough, you get the drift. Note that each formula features derivatives of  $y$  on the right that are of lower order, so we already know them from previous formulas. We could substitute for them, but then the expressions would be even longer. It seems easier to use these formulas recursively.

Having the derivatives, we obtain a Taylor polynomial  $T(h)$  of the solution  $y(x)$ , with chosen degree and centered at  $x_0$ . We use it to determine  $y_1 = T(h)$ , we have  $x_1 = x_0 + h$  and we can start the whole procedure again.

An example explains this best.

**Example 13a.a:** Consider the equation  $y' = 2x(y - 1)$  and the initial condition  $y(0) = 2$ . We want to approximate the solution  $y(x)$  of this problem using step size  $h = 0.5$ .

We start by approximating the unknown solution using a fifth degree Taylor polynomial centered at  $x_0 = 0$ . First we prepare the derivatives. Keeping in mind that in fact  $y = y(x)$ ,  $y' = y'(x)$ , ...

we differentiate the given equation repeatedly and obtain

$$\begin{aligned}y' &= 2x(y - 1), \\y'' &= 2(y - 1) + 2xy', \\y''' &= 2y' + 2y' + 2xy'' = 4y' + 2xy'', \\y'''' &= 4y'' + 2y'' + 2xy''' = 6y'' + 2xy''', \\y''''' &= 6y''' + 2y''' + 2xy'''' = 8y''' + 2xy''''.\end{aligned}$$

Sometimes we are in luck and a pattern emerges. Here I can see this:

$$y^{(k+1)} = 2ky^{(k-1)} + 2xy^{(k)},$$

and I am fairly confident that I could prove it using induction, but we need not worry about it now. We substitute the initial condition, using recursively the newly obtained derivatives as they appear,

$$\begin{aligned}y(0) &= 2 \\y'(0) &= 2 \cdot 0 \cdot 1 = 0, \\y''(0) &= 2 \cdot 1 + 2 \cdot 0 \cdot 0 = 2, \\y'''(0) &= 4 \cdot 0 + 2 \cdot 0 \cdot 2 = 0, \\y''''(0) &= 12, \\y'''''(0) &= 0.\end{aligned}$$

We obtain the approximation (the desired Taylor polynomial)

$$T_5(t) = 2 + \frac{1}{2!}2t^2 + \frac{1}{4!}12t^4 = 2 + h^2 + \frac{1}{2}t^4.$$

We used  $t$  as a variable because  $h$  now has a fixed value. For  $t$  small we can write that

$$y(0+t) = T_5(t) + O(t^6) \text{ as } t \rightarrow 0^+.$$

We decided on the step  $h = 0.5$ , this is not too large, so we hope that the error is reasonably small as well and decide to approximate

$$y(0.5) \approx T_5(0.5) = 2.28125.$$

We set  $x_1 = 0.5$ ,  $y_1 = 2.28125$  and we are ready to go again.

We want to create the Taylor polynomial of degree 5 of the solution  $y(x)$  that passes through  $(x_1, y_1)$ . Note that the preparatory work we did above with derivatives was in fact general, so we can just substitute a different point, namely  $(x_1, y_1)$ , and things should go nicely.

$$\begin{aligned}y(0.5) &= 2.28125, \\y'(0.5) &= 2 \cdot 0.5 \cdot (2.28125 - 1) = 1.28125, \\y''(0.5) &= 3.84375, \\y'''(0.5) &= 8.96875, \\y''''(0.5) &= 32.03125, \\y'''''(0.5) &= 103.78125.\end{aligned}$$

Now we can form the Taylor approximation (for small  $t$ )

$$\begin{aligned}y(x_1 + t) &= y(x_1) + y'(x_1)t + \frac{1}{2}y''(x_1)t^2 + \frac{1}{3!}y'''(x_1)t^3 + \frac{1}{4!}y''''(x_1)t^4 + \frac{1}{5!}y'''''(x_1)t^5 + O(t^6), \quad t \rightarrow 0^+, \\y(0.5 + t) &= 2.28125 + 1.28125t + \frac{1}{2}3.84375t^2 + \frac{1}{3!}8.96875t^3 + \frac{1}{4!}32.03125t^4 + \frac{1}{5!}103.78125t^5 + O(t^6), \quad t \rightarrow 0^+\end{aligned}$$

This hopefully approximates the real solution well near  $x_1 = 0.5$ , so we use it to move another step to the right. We define  $x_2 = x_1 + h = 1.0$  and our best approximation for  $y(x_2)$  is  $y_2$  obtained by substituting  $t = h = 0.5$  into that formula above,  $y_2 = 3.6996338\dots$

It is quite obvious that we could go on in this way until we have approximations over any desired interval starting at  $x_0 = 0$ .

Remark: If we decide to substitute derivatives recursively already in the general formula, we obtain

$$\begin{aligned}y' &= 2x(y - 1), \\y'' &= 2(y - 1) + 2x \cdot 2x(y - 1) = (4x^2 + 2)(y - 1), \\y''' &= 4 \cdot 2x(y - 1) + 2x(4x^2 + 2)(y - 1) = (8x^3 + 12x)(y - 1), \\y'''' &= 6(4x^2 + 2)(y - 1) + 2x(8x^3 + 12x)(y - 1) = (16x^4 + 48x^2 + 12)(y - 1), \\y''''' &= (64x^3 + 96x)(32x^5 + 160x^3 + 120x)(y - 1).\end{aligned}$$

We obtained formulas that only feature  $x$  and  $y$ , so the initial conditions  $(x_i, y_i)$  can be easily substituted. On the other hand, they seem less friendly and I have trouble seeing any pattern here, so our original approach seems better when it comes to polynomials of higher degree.

△

Note that a large part of this procedure, even the Taylor polynomial itself, can be prepared beforehand with a general point  $(x_i, y_i)$ , then we just keep making those steps. We are ready to specify the method.

### Algorithm 13a.1.

⟨Taylor method⟩

Given: ODE  $y' = f(x, y)$  on  $[x_0, x_0 + T]$ , initial condition  $y_0$ , and  $n \in \mathbb{N}$ ,

set  $h = \frac{T}{n}$  and  $x_i = x_0 + ih$  for  $i = 1, \dots, n$ .

Also given: Degree  $N$  of the approximating polynomial.

-1. Prepare the Taylor polynomial:

a) Find formulas for derivatives  $y'(x) = f(x, y(x))$ ,  $y''(x) = [y'(x)]' = \frac{\partial}{\partial x}[f(x, y(x))]$ ,  $\dots$ ,  $y^{(N)}(x) = [y^{(N-1)}]'(x)$ .

b) Substitute  $x = x_i$ ,  $y(x) = y_i$  into those formulas recursively and use the resulting expressions as coefficients, obtaining a general formula for the Taylor polynomial approximating the solution that passes through the point  $(x_i, y_i)$ :

$$\begin{aligned}T_n(t) &= y_i + y'(x_i)t + \frac{1}{2}y''(x_i)t^2 + \dots + \frac{1}{N!}y^{(N)}(x_i)t^N \\ &= y_i + f(x_i, y_i)t + \dots\end{aligned}$$

0.  $y_0$  is given.

1. For  $i = 0, \dots, n - 1$ , use  $(x_i, y_i)$  to create a concrete polynomial  $T_n$ , then define  $y_{i+1} = T_N(h)$ .

△

We will return to our example and apply this algorithm.

**Example 13a.b:** Consider the equation  $y' = 2x(y - 1)$  and the initial condition  $y(0) = 2$ . We will approximate its solution on  $[0, 2]$  using the Taylor method with  $N = 2$  and  $n = 4$ .

This setup means that  $h = 0.5 = \frac{1}{2}$  and we have the partition  $\{0, 0.5, 1, 1.5, 2\}$  for the interval  $[0, 2]$ .

Since  $N = 2$ , we need formulas for the first two derivatives of the solution:

$$y' = 2x(y - 1), \quad y'' = 2(y - 1) + 2xy' = 2(y - 1) + 4x^2(y - 1) = (4x^2 + 2)(y - 1).$$

Thus the Taylor polynomial of the solution that passes through a point  $(x_i, y_i)$  is

$$T_2(t) = y_i + 2x_i(y_i - 1)t + \frac{1}{2}(4x_i^2 + 2)(y_i - 1)t^2.$$

a)  $y_0 = 2$  is given.

b)  $(x_0, y_0) = (0, 2)$ . We substitute into  $T_2$ , obtaining

$$T_2(t) = 2 + t^2 \implies y_1 = \frac{9}{4} = 2.25.$$

c)  $(x_1, y_1) = (0.5, \frac{9}{4})$ . We substitute into  $T_2$ , obtaining

$$T_2(t) = \frac{9}{4} + \frac{5}{4}t + \frac{15}{8}t^2. \implies y_2 = \frac{107}{32} = 3.34375.$$

d)  $(x_2, y_2) = (1, \frac{107}{32})$ . We substitute into  $T_2$ , obtaining

$$T_2(t) = \frac{107}{32} + \frac{75}{16}t + \frac{225}{32}t^2. \implies y_3 = \frac{953}{128} = 7.445\dots$$

e)  $(x_3, y_3) = (1.5, \frac{953}{128})$ . We substitute into  $T_2$ , obtaining

$$T_2(t) = \frac{953}{128} + \frac{2475}{128}t + \frac{9075}{256}t^2. \implies y_4 = \frac{26599}{1024} = 25.976\dots$$

How good are these approximations? Let's compare.

$x_i$ :	0.0	0.5	1.0	1.5	2
$y_i$ :	2.00	2.25	3.34...	7.44...	25.98...
$y(x_i)$ :	2.00	2.28...	3.72...	10.59...	55.60...
$ E_i $ :	0.00	0.03...	0.37...	3.04...	29.62...

The error grows quite a bit, but this is not surprising, the Taylor method does not work all that well for large  $h$ , so people tend to use other methods for a quick peek at a solution as they are usually faster. Only when  $h$  gets smaller then the Taylor method really starts to shine, so it is used when very high quality approximation is needed. Then people prefer not to go too high with the degree  $N$  (as high degree polynomials tend to oscillate wildly) and get the quality by making  $h$  smaller instead. For instance, Maple does not allow polynomials of order higher than 22 and its default when working with 10 digit accuracy is  $N = 15$ .

△

The way we derived this method immediately shows what the local error is.

**Theorem 13a.2.**

The Taylor method that uses polynomials of degree  $p$  is a method of order  $p$ .

The hardest part—preparing the derivatives—can be handled automatically by a suitable computer algebra system, so this method does not take as much effort as it seems in our example. More importantly, this method can be also used for solving differential equations of higher order, which is something that most other methods we meet here (including RK methods) cannot do. So we definitely expect to meet this method again in the appropriate chapter.

### 13b. Taylor expansions

We return to the Taylor expansion at  $x_0 = 0$  of the solution of  $y' = 2x(y - 1)$  at the beginning of the previous section.

$$y(x) = 2 + x^2 + \frac{1}{2}x^4 + \dots$$

We know that it is no good when we move further away from  $x_0 = 0$ . On the other hand, we easily show by separation that the solution we seek is

$$y(x) = 1 + e^{x^2} = 2 + x^2 + \frac{1}{2}x^4 + \frac{1}{6}x^6 + \frac{1}{24}x^8 + \dots$$

An interesting idea comes up: Would it be possible to actually construct not a Taylor polynomial, but a Taylor series for the solution? This would require that we identify some pattern when deriving formulas for  $y^{(k)}(x_0)$ . We did seem to see one in the example above, but to make things

easier we will look at another equation. And to make things more interesting we will try to solve it in general, with parameters.

**Example 13b.a:** Consider the famous exponential growth, that is, the initial value problem  $y' = ay$ ,  $y(t_0) = A$ . We will attempt to construct a Taylor series for the solution centered at  $t_0$ . We start by preparing derivatives.

$$\begin{aligned}y' &= ay &\implies y'(t_0) &= ay(t_0) = aA, \\y'' &= ay' &\implies y''(t_0) &= a \cdot aA = a^2A, \\y''' &= ay'' &\implies y'''(t_0) &= a \cdot a^2A = a^3A, \\y'''' &= ay''' &\implies y''''(t_0) &= a \cdot a^3A = a^4A, \\y''''' &= ay'''' &\implies y'''''(t_0) &= a \cdot a^4A = a^5A.\end{aligned}$$

The pattern seems clear, and it is easy to show by induction that for any  $k \in \mathbb{N}_0$  we indeed have  $y^{(k)}(t_0) = a^k A$ . Now we are ready to form the Taylor series.

$$y(t) = \sum_{k=0}^{\infty} \frac{y^{(k)}(t_0)}{k!} (t - t_0)^k = \sum_{k=0}^{\infty} \frac{Aa^k}{k!} (t - t_0)^k = A \sum_{k=0}^{\infty} \frac{(a(t - t_0))^k}{k!} = A e^{a(t-t_0)}.$$

And we are not really surprised, this was to be expected.

△

Of course, things can easily get more interesting, even a nice equation can already provide a challenge.

**Example 13b.b:** Consider the IVP  $y = \frac{y}{x}$ ,  $y(1) = 2$ . We will try to find the Taylor expansion of  $y$  centred at  $x_0 = 1$ . First we prepare derivatives.

$$\begin{aligned}y' &= \frac{y}{x}, \\y'' &= \frac{y'x - y}{x^2}, \\y''' &= \frac{(y''x + y' - y')x^2 - (y'x - y)2x}{x^4} = \frac{y''x^2 - 2y'x + 2y}{x^3}, \\y'''' &= \frac{(y'''x^2 + y''2x - 2y''x - 2y' + 2y')x^3 - (y''x^2 - 2y'x + 2y)3x^2}{x^6} = \frac{y'''x^3 - 3y''x^2 + 6y'x - 6y}{x^4}, \\y''''' &= \frac{(y''''x^3 + y'''3x^2 - 3y'''x^2 - 6y''x + 6y''x + 6y' - 6y')x^4 - (y'''x^3 - 3y''x^2 + 6y'x - 6y)4x^3}{x^8} = \frac{y'''''x^4}{x^8}.\end{aligned}$$

Call me a chicken but I've had enough. Do you see any pattern? Well, let's put this one on a back burner for the moment and see what we are getting. We substitute the initial condition,

$$\begin{aligned}y(1) &= 2, \\y'(1) &= \frac{2}{1} = 2, \\y''(1) &= \frac{2 \cdot 1 - 2}{1^2} = 0, \\y'''(1) &= \frac{0 - 4 + 4}{1} = 0, \\y''''(1) &= \frac{0 - 0 + 12 - 12}{1} = 0, \\y'''''(1) &= \frac{-48 + 48}{1} = 0.\end{aligned}$$

It seems that the expansion starts

$$y(x) = 2 + 2(x - 1) + 0 \cdot (x - 1)^2 + 0 \cdot (x - 1)^2 + 0 \cdot (x - 1)^4 + 0 \cdot (x - 1)^5 + \dots = 2x + \dots$$

Of course, we can easily solve this equation by separation and see that the actual solution is  $y(x) = 2x$ , so the expansion is already finished, but we wouldn't know it were we not able to solve the given equation.

So it is time to return to our question about a pattern. Did you think of something?

It seems to me that

$$y^{(k+1)}(x) = \frac{1}{x^{k+1}} \sum_{i=0}^k \frac{k!}{i!} y^{(i)} x^i.$$

It is possible to prove this using induction, and then we can use induction to prove that  $y^{(k)}(1) = 0$  for all  $k \geq 2$ , so the expansion does yield  $y(x) = 2x$ , but I can't help the feeling that I would much rather solve it by separation.

△

Waiting for inspiration from above when looking for patterns does not sound like a reliable approach. After all, if we try it in general, already the third derivative is bad enough to scare all but the strongest among us.

$$\begin{aligned} y(x_0) &= y_0, \\ y'(x_0) &= f(x_0, y_0), \\ y''(x_0) &= \frac{\partial f}{\partial x}(x_0, y_0) + \frac{\partial f}{\partial y}(x_0, y_0) f(x_0, y_0), \\ y'''(x_0) &= \frac{\partial^2 f}{\partial x^2}(x_0, y_0) + \frac{\partial^2 f}{\partial y \partial x}(x_0, y_0) f(x_0, y_0) + \frac{\partial^2 f}{\partial x \partial y}(x_0, y_0) f(x_0, y_0) + \frac{\partial^2 f}{\partial y^2}(x_0, y_0) [f(x_0, y_0)]^2 \\ &\quad + \frac{\partial f}{\partial y}(x_0, y_0) \left( \frac{\partial f}{\partial x}(x_0, y_0) + \frac{\partial f}{\partial y}(x_0, y_0) f(x_0, y_0) \right), \\ &\vdots \end{aligned}$$

When we see no pattern, we hit a dead end and the show is over.

Still, when all else fails, this may be the last hope. While the use of this method for differential equations of first order is limited, it becomes much more viable when it comes to more complicated equations. In fact, the series expansion approach is indispensable in some important applications, but then to make it practical we have to use a different language for it. We will talk about it in chapter 34.

Note that this approach provided us with a formula. Therefore this is not a numerical method but an analytic method, a competitor to separation and other “theoretical” methods. We included it here because it is a natural extension of the Taylor method.

### 13c. Multistep methods

Multistep methods use information about more points in the past when constructing the next step. For simplicity, assume again that we have an equidistant partition  $\{x_0, \dots, x_n\}$  with step  $h$ . To derive  $y_{i+1}$ , a  $m$ -step formula would use the  $m$  last known values  $y_i, y_{i-1}, \dots, y_{i-m+1}$ .

The simplest way to arrive at such a method is to use the approach we already saw earlier, we replace the derivative in  $y' = f(x, y)$  by the central difference and error term. We obtain

$$\frac{y(x+h) - y(x-h)}{2h} + O(h^2) = f(x, y) \implies y(x_{i+1}) = y(x_{i-1}) + 2hf(x_i, y_i) + O(h^3).$$

This shows that if we define  $y_{i+1}$  using the formula on the right, we obtain a method of local order  $O(h^3)$  and global order 2. It is a nice explicit method that can be easily programmed.

### 13d. Finite differences

## 14. Interpolation

We saw how solutions of ODEs can be approximated using various numerical methods. Such an resulting approximation is given as a list of pairs  $(x_i, y_i)$ . This is in fact a familiar situation. When we measure development of some quantity in time (or with respect to some other variable like position), we always obtain a list of values, because instruments cannot provide us with formulas. They just take measurements at (perhaps regular) intervals and record the values.

This is the starting point for our investigation. We are given a list of pairs  $(x_i, y_i)$  that describe our function. However, this is not the most convenient form if we want to investigate this function, and often we would also like to know what happens between the points where the function is given to us. The natural approach is to “fit a curve”, that is, provide a formula (or formulas) that best fit the data. Interpolation is a part of mathematics that suggests possible answers. There are quite a few approaches one can follow, we will just outline the most important ones here.

There is also a related field called extrapolation. While interpolation focuses on what is happening between the supplied data points, extrapolation looks outside. Basically, after we fit a curve to our data, we use it to make a guess about future development. This is undoubtedly very useful, but as we will see below, fitting a curve to data is not a deterministic process. There are always many candidates to choose from, such a selection is in a way arbitrary, and therefore we cannot with any confidence truly predict what comes next. As such, this topic is sometimes closer to black magic than mathematics and we will not address it here.

This is a survey chapter, its purpose is to give the reader an intuitive understanding of concepts. We will therefore prove only some results and leave details of some topics for independent study.

### 14a. Interpolation by polynomials

Polynomials are unique among functions in one key practical aspect: They can be evaluated using just the basic algebraic operations, which is exactly what computers can do best and fast. This gives them an edge when it comes to practical applications. They are also easy to differentiate and integrate, in short they are the ideal object to use when attempting to interpolate known data. By a happy coincidence, we learn in linear algebra that when given  $n + 1$  points in the plane whose  $x$ -coordinates are distinct, we can fit a unique polynomial of degree  $n$  through them. This sounds like an invitation to interpolation.

#### Example 14a.a:

Consider function given by the following chart.

$x:$	-2	0	1	2
$y:$	-5	4	4	5

We have four points in the plane, so there must be a polynomial  $p$  of degree 3 passing through them. How do we find it? We want a polynomial of the form

$$p(x) = Ax^3 + Bx^2 + Cx + D$$

that satisfies  $p(-2) = -5$ ,  $p(0) = 4$ ,  $p(1) = 4$ , and  $p(2) = 5$ . This translates to a system of linear equations.

$$-8A + 4B - 2C + D = -5$$

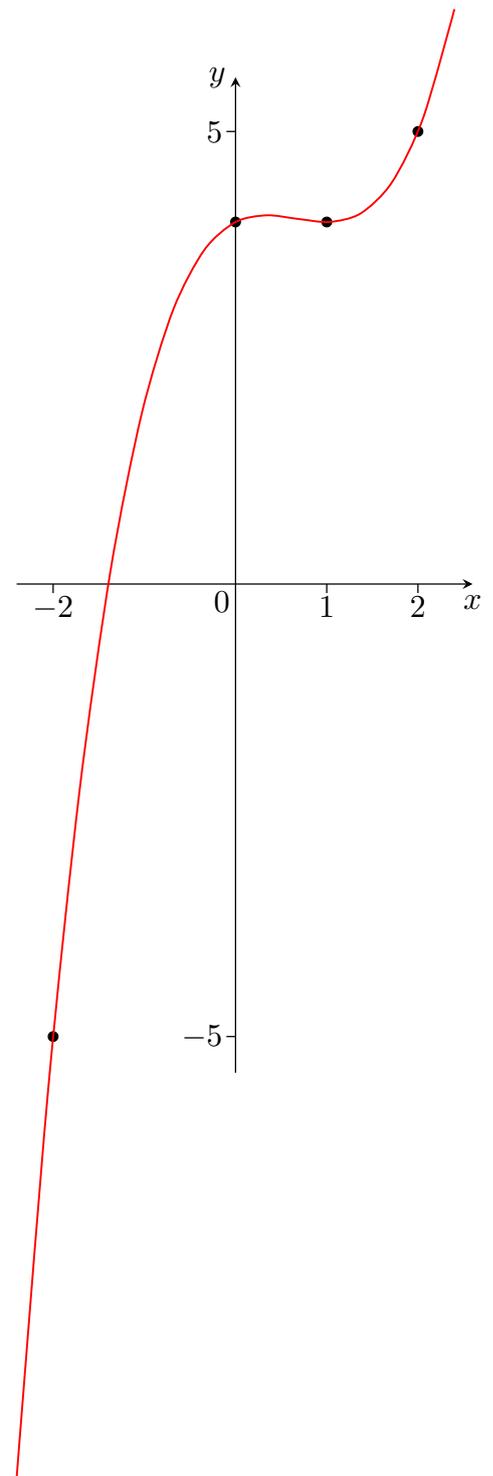
$$D = 4$$

$$A + B + C + D = 4$$

$$8A + 4B + 2C + D = 5$$

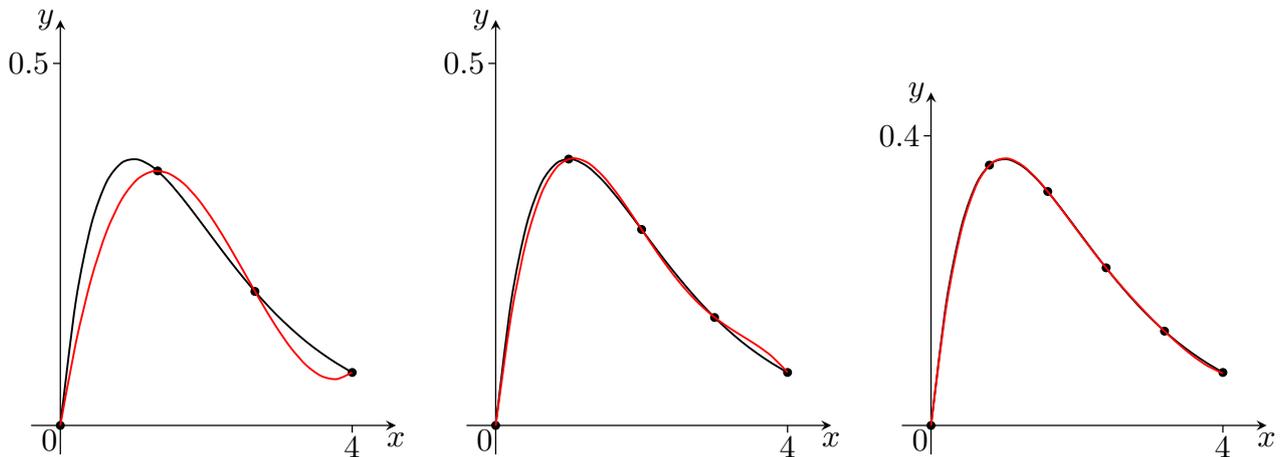
We obtain  $A = \frac{1}{2}$ ,  $B = -1$ ,  $C = \frac{1}{2}$ ,  $D = 4$ . We conclude that the polynomial  $p(x) = \frac{1}{2}x^3 - x^2 + \frac{1}{2}x + 4$  passes through the given points. We could use it to approximate the given function between the given values for  $x$ .

Is this approximation actually faithful? We have no way to know, as we do not know how the actual function looks like between the given points.



△

Since we do not know the original function apart from the given points, we cannot judge accuracy of the polynomial interpolation. But we can at least try to get some feeling for what to expect. We take a known function, pick some points on it and then use polynomial interpolation. In the following pictures we consider the function  $f(x) = x e^{-x}$  on the interval  $[0, 8]$ . We split the interval into 4, 6, and 8 segments, that is, we consider 5, 7, and 9 regularly spaced points on this graph, and each time we put an interpolating polynomial through it.



The pictures suggest that the more data points we choose, the better the interpolation. That has been the practical experience, and there are also some theoretical results that support this expectation in cases when we know some information about the function from which our data comes.

**Theorem 14a.1.**

Let  $f$  be a function defined on a closed interval  $I$ . For some  $n \in \mathbb{N}$ , let  $x_0, x_1, \dots, x_n$  be distinct points from the interval  $I$ .  $p(x)$  be the unique interpolating polynomial of degree at most  $n$  for data points  $(x_i, f(x_i))$ .

If the function  $f$  is  $n + 1$  times continuously differentiable on the interior of  $I$ , then for every  $x \in I$  there is some  $\xi \in I$  such that

$$f(x) - p(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^n (x - x_i).$$

In general we cannot say anything about this error of approximation. However, if we have a reason to believe that there unknown function  $f$  has a universal upper bound for all derivatives on  $I$ , then the error  $f(x) - p(x)$  tends to zero as we increase the number of data points to infinity.

Given good properties of polynomials and a seemingly good behaviour of polynomial interpolation, it is not surprising that it has been used in many applications. For instance, when we send some text to a printer, the shape of individual letters is not specified precisely, but just few key points are sent, and the shape inbetween is determined using polynomial interpolation. Such practical applications brought forth practical question, and an important one is the amount of work it takes to create an interpolating polynomial.

In the above example we used an intuitive approach, and it can be always used. However, It requires that we solve a system of linear equation of dimension  $n + 1$ , and while this is a standard task, it is also known that it is quite labor-intensive, see chapter 22. It is therefore natural to look for an easier approach.

We take a clue from linear algebra. Polynomials of degree at most  $n$  form a linear space  $P_n$ , and the powers  $\{1 = x^0, x, \dots, x^n\}$  form a basis of this space. Every polynomial of degree at most  $n$  is a linear combination of this basis. The key idea is that we can find a better basis for our interpolating polynomials. What kind of a basis would prove suitable?

In our example we work with polynomials of degree three, and the corresponding space  $P_4$  is four-

dimensional. Imagine that we find four polynomials of degree three with the following properties:

$$\begin{aligned}l_0(-2) &= 1l_0(0) = 0l_0(1) = 0l_0(2) = 0 \\l_1(-2) &= 0l_1(0) = 1l_1(1) = 0l_1(2) = 0 \\l_2(-2) &= 0l_2(0) = 0l_2(1) = 2l_2(2) = 0 \\l_3(-2) &= 0l_3(0) = 0l_3(1) = 0l_3(2) = 3\end{aligned}$$

It is quite obvious that such polynomials would be linearly independent. The key property is that when we form a linear combination, then the polynomials do not interfere with one another at given points  $x_i$ . For instance, at the point  $x_2 = 1$  only the polynomial  $l_2$  is not zero. Consequently, when we form a linear combination

$$a + 0l_0(x) + a_1l_1(x) + a_2l_2(x) + a_3l_3(x),$$

then we readily conclude that its value at  $x_i$  is  $a_i$ . We will use this the other way around. We want to find a polynomial with values  $y_i$  at points  $x_i$ , and we find it as  $\sum y_i l_i(x)$ . We obtain the polynomial

$$p(x) = -5 \cdot l_0(x) + 4 \cdot l_1(x) + 4 \cdot l_2(x) + 5 \cdot l_3(x).$$

Then we have, for instance,

$$p(-2) = -5 \cdot 1 + 4 \cdot 0 + 4 \cdot 0 + 5 \cdot 0 = -5.$$

So this seems to work well, we just need to figure out how to get such polynomials  $l_i$ .

### Example 14a.b:

We continue with the example 14a.a. Can we find a polynomial  $l_0(x)$  such that  $l_0(-2) = 1$  and  $l_0(0) = l_0(1) = l_0(2) = 0$ ? We could try it as above, by solving a system of equations, but that is exactly what we wanted to avoid. We therefore apply a different approach.

We start by preparing a temporary polynomial  $q$  that would be zero at  $x = 0, 1, 2$ . Now this is easy,  $q_0(x) = (x - 0)(x - 1)(x - 2)$ . Will this do? We check:  $q_0(-2) = -24$ , that's not it. But we are almost there, we take

$$l_0(x) = \frac{1}{-24}q_0(x) = -\frac{1}{24}x(x - 1)(x - 2).$$

The values at  $x = 0, 1, 2$  are still zero and now also  $l_0(-2) = 1$  as required.

Similarly we identify the other polynomials.

$$q_1(x) = (x - (-2))(x - 1)(x - 2) = (x + 2)(x - 1)(x - 2), \quad q_1(0) = 4$$

$$\implies l_1(x) = \frac{1}{4}(x + 2)(x - 1)(x - 2);$$

$$q_2(x) = (x - (-2))(x - 0)(x - 2) = (x + 2)x(x - 2), \quad q_2(1) = -3$$

$$\implies l_2(x) = -\frac{1}{3}(x + 2)x(x - 2);$$

$$q_3(x) = (x - (-2))(x - 0)(x - 1) = (x + 2)x(x - 1), \quad q_3(2) = 8$$

$$\implies l_3(x) = \frac{1}{8}(x + 2)x(x - 1).$$

Now we can form the interpolating polynomial.

$$\begin{aligned}p(x) &= -5 \cdot \left(-\frac{1}{24}\right)x(x - 1)(x - 2) + 4 \cdot \frac{1}{4}(x + 2)(x - 1)(x - 2) \\ &\quad + 4 \cdot \left(-\frac{1}{3}\right)(x + 2)x(x - 2) + 5 \cdot \frac{1}{8}(x + 2)x(x - 1).\end{aligned}$$

It takes a bit of manual work to show that this polynomial is in fact the same one that we deduced in example 14a.a. It is just written in a different way, that is, instead of the basis  $\{1, x, x^2, x^3\}$  we expressed it with respect to the basis  $\{l_0, l_1, l_2, l_3\}$ .

△

The new version of the interpolating polynomial definitely looks less friendly. However, for larger sets of points it is significantly easier to construct. The polynomials that we introduced here are so useful that they have a name. Before we state a formal definition, note that the polynomials in such a basis can be constructed using a universal formula that features a certain symmetry. For instance, the polynomial  $l_1$  above can be obtained as follows.

$$l_1(x) = \frac{(x+2)(x-1)(x-2)}{(0+2)(0-1)(0-2)}.$$

This pattern is general.

**Definition 14a.2.**

For given distinct numbers  $x_0, x_1, \dots, x_n \in \mathbb{R}$  we define the **Lagrange basis polynomials** as

$$l_i(x) = \frac{1}{\prod_{j \neq i} (x_i - x_j)} \prod_{j \neq i} (x - x_j) = \prod_{j \neq i} \frac{x - x_j}{x_i - x_j}$$

for  $i = 0, 1, \dots, n$ .

For given data  $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n) \in \mathbb{R}^2$  such that  $x_0, \dots, x_n$  are distinct we define the **Lagrange interpolating polynomial** as

$$p(x) = \sum_{i=0}^n y_i l_i(x),$$

where  $l_i$  are Lagrange basis polynomials corresponding to  $\{x_i\}$ .

We easily confirm that given any set of  $n+1$  pairs  $(x_i, y_i)$  with  $x_i$  distinct, then the Lagrange interpolating polynomial  $p$  is just that: A polynomial of degree  $n$  that passes through all of these points. This is in fact a simple proof of the existence of an interpolating polynomial.

Now imagine that somebody comes up with another interpolating polynomial  $q(x)$  of degree (at most)  $n$  passing through the same points. Then the polynomial  $p - q$  has degree at most  $n$ , and it is obviously zero at all points  $x_i$ . There are  $n+1$  of them, so the polynomial  $p - q$  has at least  $n+1$  distinct roots, while being of order at most  $n$ . The only polynomial that works like this is the identically zero polynomial. This shows that  $p - q = 0$ , that is,  $p = q$ . We just proved that given distinct  $n+1$  points, there is just one polynomial of degree (at most)  $n$  passing through them.

As we saw in the examples, this unique polynomial can be written in different ways, and the Lagrange form is quite useful.

Very often the values  $x_i$  are ordered, that is,  $x_0, x_1 < \dots < x_n$ . This is not necessary, but it can be convenient, especially if the numbers  $x_i$  are spaced regularly, with mutual distance  $h$ . For instance, this is the case when we solve some ODE numerically using a standard Runge-Kutta method. Then we can use the ordering to our advantage and derive the following formula for the Lagrange polynomials:

$$l_i(x) = \frac{1}{h^n i! (n-i)!} \prod_{j \neq i} (x - x_j) = \frac{\binom{n}{i}}{n! h^n} \prod_{j \neq i} (x - x_j).$$

However, this brings us to one of the disadvantages of the Lagrange basis. It sometimes happens in applications that we feel like adding some points along the curve, so that the new polynomial fits better. However, the formula for the Lagrange polynomial uses knowledge of all points in the data. This means that if we add even a single data point, we then have to recalculate everything.

People therefore wondered: Is there some basis for polynomials that could be easily extended when we add more points, preferably just by adding another polynomial to it? This sounds like a lot to ask, but there actually are such polynomials.

**Example 14a.c:** We return to the example 14a.a. We have the points  $x_0 = -2$ ,  $x_1 = 0$ ,  $x_2 = 1$ ,  $x_3 = 2$ , but now we will assume that they come one at a time.

First we get the point  $x_0$  and we are looking for a polynomial  $n_0(x)$  that would make it easy to approximate any value at  $x_0$ . The obvious candidate is the constant polynomial  $n_0(x) = 1$ , because we easily determine interpolating polynomial  $y_0 n_0(x)$  for any data point  $(x_0, y_0)$ . In particular, the first given datapoint  $(-2, -4)$  from our example leads to the polynomial  $p_0(x) = -5 \cdot 1$ .

Now we add  $x_1$  into the mix. We want a polynomial  $n_1(x)$  that could be added to  $p_0(x)$  without spoiling our successful interpolation at  $x_0$ . The simplest answer is  $n_1(x) = x - x_0 = x + 2$ . We want to add a new term based on  $n_1(x)$  to  $p_0(x)$  so that the new polynomial also interpolates at  $x_1$ . That is, we are looking for a polynomial of the form

$$-5 \cdot 1 + a \cdot (x + 2)$$

that would work with the first two data points. It obviously has the right value at  $x_0 = -2$ , but we also expect it to pass through  $(0, 4)$ . This sets up just one equation that we easily solve:

$$-5 \cdot 1 + a \cdot (0 + 2) = 4 \implies a = \frac{9}{2}.$$

We now have the interpolating polynomial  $p_1(x) = -5 \cdot 1 + \frac{9}{2}(x + 2)$ . Note that when we defined  $n_1(x)$ , we did not actually make any reference to  $x_1$ . We just wanted it to be zero at the previous point, which is a general pattern here. One could also ask for some condition at  $x_1$ , for instance we could insist that  $n_1(x_1) = 1$ , and in fact this is easy to arrange, but it turns out that it would not really help us, so we take the easy path here.

It is time to add the third point  $(1, 4)$ . We want a polynomial  $n_2(x)$  that can be added to  $p_1(x)$  without spoiling its value at  $x_0$  and  $x_1$ , and the natural choice is  $n_2(x) = (x - x_0)(x - x_1) = (x + 2)x$ . The polynomial

$$-5 \cdot 1 + \frac{9}{2}(x + 2) + a(x + 2)x$$

has the right value at  $x_0 = -2$  and  $x_1 = 0$ , we need to fix the value at  $x_2 = 1$ :

$$-5 \cdot 1 + \frac{9}{2}(1 + 2) + a(1 + 2) \cdot 1 = 4 \implies a = -\frac{3}{2}.$$

Now we have the polynomial  $p_2(x) = -5 \cdot 1 + \frac{9}{2}(x + 2) - \frac{3}{2}(x + 2)x$ .

The last member in our basis is

$$n_3(x) = (x - x_0)(x - x_1)(x - x_2) = (x + 2)x(x - 1).$$

When we extend  $p_2(x)$  using this basic polynomial, we have one requirement on value at  $x_3 = 2$ :

$$-5 \cdot 1 + \frac{9}{2}(2 + 2) - \frac{3}{2}(2 + 2) \cdot 2 + a(2 + 2) \cdot 2 \cdot (2 - 1) = 5 \implies a = \frac{1}{2}.$$

We arrive at the interpolating polynomial for the given data,

$$p(x) = -5 \cdot 1 + \frac{9}{2}(x + 2) - \frac{3}{2}(x + 2)x + \frac{1}{2}(x + 2)x(x - 1).$$

Multiplying out we confirm that this is just another way to express the original interpolating polynomial from example 14a.a.

△

The pattern for creating the right polynomials seems clear.

**Definition 14a.3.**

For given distinct numbers  $x_0, x_1, \dots, x_n \in \mathbb{R}$  we define the **Newton basis polynomials** as

$$n_i(x) = \prod_{0 \leq j < i} (x - x_j)$$

for  $i = 0, 1, \dots, n$ .

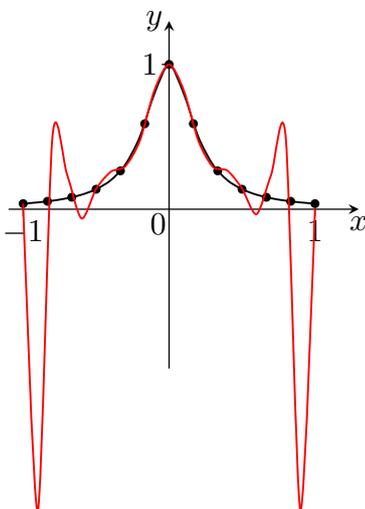
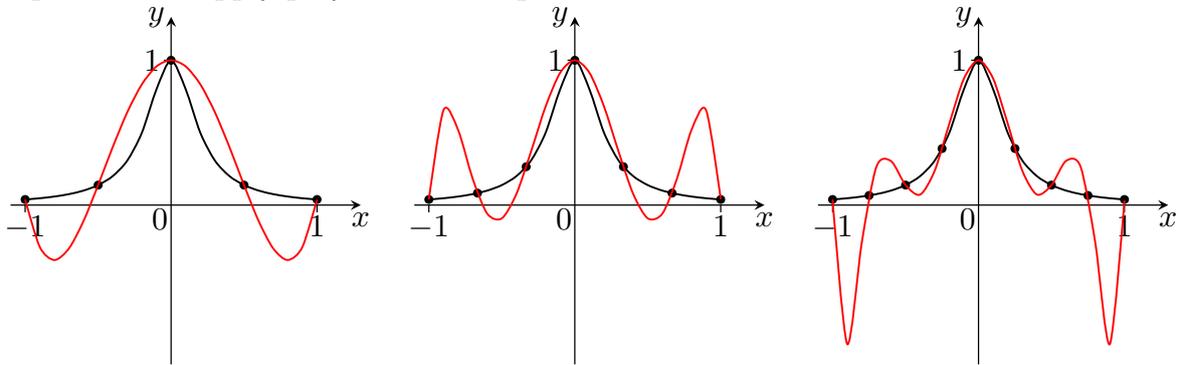
The notation  $\prod$  for product of terms is defined to yield 1 when there are no terms to be multiplied, which is exactly the case for  $i = 0$ .

The Newton basis has several advantages. Besides the fact that such a basis can be easily extended if more data is added, which was our motivation for introducing them, they are also obviously very easy to set up. ON the other hand, while for a Lagrange interpolation  $\sum a_i l_i(x)$  the coefficients  $a_i$  we lifted directly from the data, with a Newton interpolating polynomial  $\sum a_i n_i(x)$  we have to do some work. Actually, as we saw, it is not so bad, we can find them recursively by solving a simple linear equation for each. Still, it is natural to ask whether there is some formula for these coefficients. It turns out that there is.

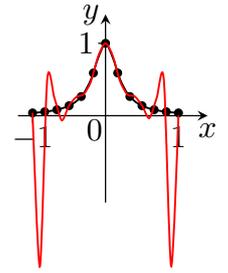
When we want to interpolate given data points  $(x_i, y_i)$  for  $i = 0, \dots, n$  by a polynomial of the form  $\sum_{i=0}^n a_i n_i(x)$ , we should use the coefficients  $a_i = [y_0, \dots, y_i]$ , these are the divided differences. We refer a curious reader to the ample literature on this topic, as we want to focus our attention in a different direction.

As we saw, the same polynomial can be expressed in many different ways, depending on the basis we choose. There are in fact quite a few different bases, chosen to fulfill various practical purposes. One important concern is related to quality of interpolation. Our experiment above suggested that by increasing the number of data points we improve quality, and it simplifies things if we take equidistant points. It therefore came as a surprise when at the beginning of the 20th century people found out that this is not a rule, and things can go wrong. At the root of the problem is the tendency of polynomials of high degree to oscillate.

In the following pictures we will interpolate the function  $f(x) = \frac{1}{25x^2+1}$  on the interval  $[-1, 1]$ . We split this interval into 4, 6, and 8 segments, that is, we will sample it with regularly spaced 5, 7, and 9 points, and apply polynomial interpolation.



We notice that as we increase the number of data points, the interpolation is more faithful around the middle, which was expected, but near the endpoint the situation actually seems to get worse. And indeed, as we increase the number of data points, the middle region of good approximation widens, while near the endpoint there are two zones that are getting progressively more narrow (good news) but on which the interpolating polynomial oscillates more and more (bad news). On the right we see the situation when 13 data points were used.



This is called the **Runge phenomenon**. It shows that the polynomial interpolation is not as good as we hoped for, and inspires inquiries into how to avoid such problems. One possible venue is to choose data points smartly, namely, it is better not to use an equidistant spread, but make the data points denser near the endpoints. There is a rather complicated theory regarding the best choice of interpolating data points, with important applications in numerical integration and other areas of numerical analysis. A curious reader may want to look up Chebyshev polynomials (which is yet another basis for the space of polynomials) and the Gauss method for numerical integration.

Another interesting approach is to limit ourselves to polynomials of low degree. This brings us to the next section.

## 14b. Splines: Local interpolation by polynomials

Imagine a situation when we have some data points and want to use polynomials for interpolation, but we do not want to use high degree polynomials. One possible strategy is to split the data points into smaller groups and apply polynomial interpolation to each. For instance, given five data points, we can connect the first three with a (unique) quadratic polynomial and the last three with another quadratic polynomial. The middle point is used in both cases, and serves as a connecting point for the two quadratic segments. We could say that we did piecewise interpolation.

This is a viable strategy, but our freedom is severely restricted by the number of given points. For instance, if there are four data points, like we have in our introductory example, then there are only two options. Either we connect them all with one cubic polynomial, as we did in the previous section, or we split the given points into three groups of two, in effect connecting the first two, then the second with the third, and finally the third with the last. No other piecewise interpolation is possible if we expect to have all polynomials of the same degree, which is a very sensible requirement.

To cut down on complications, people adopted the simple strategy of connecting the neighboring points, and then the obvious candidate is a first degree polynomial, that is, a straight segment.

### Example 14b.a:

Consider function given by the following chart, see example 14a.a.

$x:$	-2	0	1	2
$y:$	-5	4	4	5

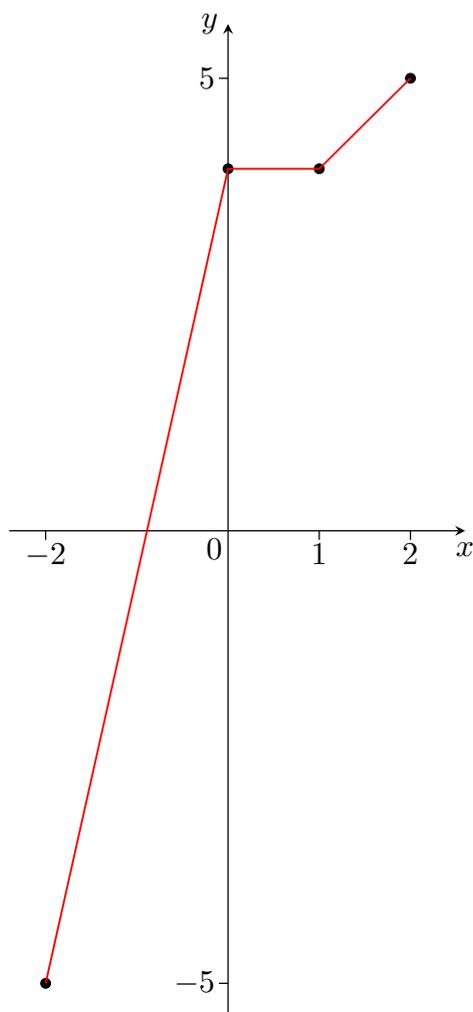
We first find the straight line connecting points  $(-2, -5)$  and  $(0, 4)$ . This is a standard task, we obtain

$$y = (-5) + \frac{4 - (-5)}{0 - (-2)}(x - (-2)) \implies y = \frac{9}{2}x + 4.$$

Similarly we proceed with the pairs  $(0, 4)$ ,  $(1, 4)$  and  $(1, 4)$ ,  $(2, 5)$ , obtaining the following interpolating function:

$$f(x) = \begin{cases} \frac{9}{2}x + 4, & x \in [-2, 0]; \\ 4, & x \in [0, 1]; \\ x + 3, & x \in [1, 2]. \end{cases}$$

The picture shows the outcome of this interpolation.



△

This kind of interpolation is very popular. For instance, when we ask some computer algebra system like Maple or Mathematica to plot a function, then it will simply evaluate it at some points and connect them with segments. However, it also has some disadvantages, namely that this curve is definitely not smooth, in particular it does not have first derivative at the “nodes”  $x_i$ . We can fix this by using polynomials of higher degree.

This brings us to the first possible meaning of the word spline. Given some data points  $(x_i, y_i)$  ordered so that  $x_i$  are increasing, by a spline we mean a function that interpolates these data points and on each interval  $[x_{i-1}, x_i]$  it is a polynomial. It is expected that the polynomials join at  $x_i$ , that is, we expect splines to be continuous. We can therefore also see a spline as a collection of polynomials. Typically we decide on polynomials of a certain degree.

**Definition 14b.1.**

Consider data  $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n) \in \mathbb{R}^2$  with  $x_0 < x_1 < \dots < x_n$ . Let  $n \in \mathbb{N}$ .

A function  $s(x)$  is called a **spline of degree  $n$**  if it

- is continuous on  $[x_0, x_n]$ ,
- satisfies  $s(x_i) = y_i$  for every  $i = 0, 1, \dots, n$ ,
- it is a polynomial of degree at most  $n$  on every interval  $[x_{i-1}, x_i]$  for  $i = 1, \dots, n$ .

For every interval  $[x_{i-1}, x_i]$ , let  $p_i(x)$  denote the corresponding polynomial restriction of  $s(x)$ . Then these polynomials must satisfy the following condition:

- (i)  $p_i(x_{i-1}) = y_{i-1}$  and  $p_i(x_i) = y_i$  for every  $i = 1, \dots, n$ .

Typically we would ask for more, but how much we can ask depends on the degree of the spline.

**Example 14b.b:** Consider again our favorite example 14a.a.

We will now find a quadratic spline. This means that we are looking for three quadratic polynomials:

$$p_1(x) = a_1x^2 + b_1x + c_1,$$

$$p_2(x) = a_2x^2 + b_2x + c_2,$$

$$p_3(x) = a_3x^2 + b_3x + c_3.$$

The baseline requirement is that each polynomial segment has the right value at its endpoints.

$$a_1 \cdot (-2)^2 + b_1 \cdot (-2) + c_1 = -5, a_1 \cdot 0^2 + b_1 \cdot 0 + c_1 = 4,$$

$$a_2 \cdot 0^2 + b_2 \cdot 0 + c_2 = 4, a_2 \cdot 1^2 + b_2 \cdot 1 + c_2 = 4,$$

$$a_3 \cdot 1^2 + b_3 \cdot 1 + c_3 = 4, a_3 \cdot 2^2 + b_3 \cdot 2 + c_3 = 5.$$

We have 6 equations, and nine parameters to play with. This gives us a chance to ask for more, and the natural requirement is that also the first derivatives of these polynomial segments should match at the joints, so that the resulting curve does not have kinks (sharp breaks). In other words, we want the resulting spline to be differentiable on the whole interval  $(-2, 2)$ . This yields the following conditions:

$$2a_1 \cdot 0 + b_1 = 2a_2 \cdot 0 + b_2,$$

$$2a_2 \cdot 1 + b_2 = 2a_3 \cdot 1^2 + b_3.$$

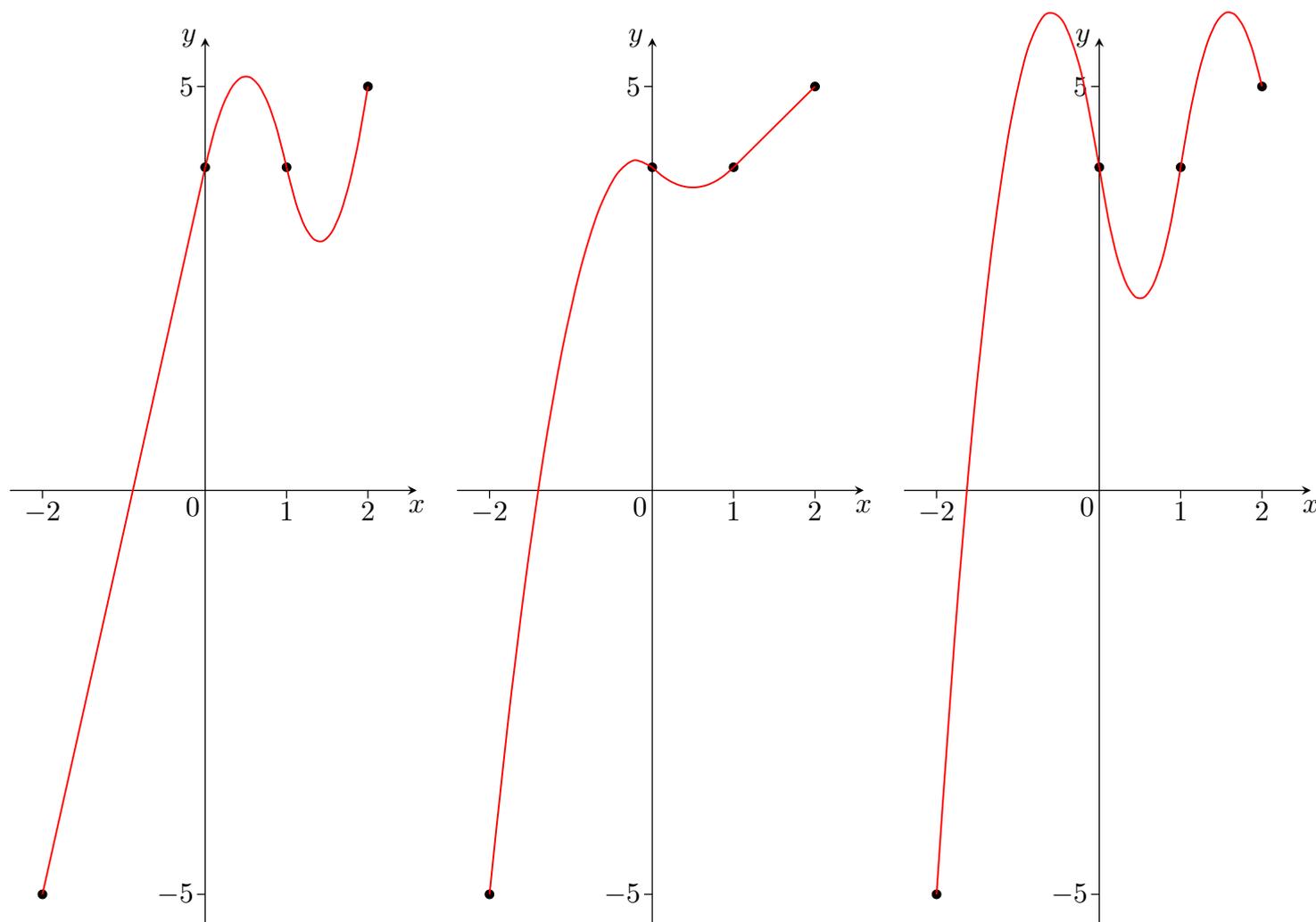
This makes eight conditions, there is still one more and there is no natural candidate. People try all kinds of things. One possibility is to ask that the curve starts flat, without curving, that is, we want the second derivative equal to zero at the left endpoint. In our case this means the condition  $a_1 = 0$ , which forces the first segment to be a straight line. This has some unpleasant consequences.

Another possibility is to make the same demand, but at the end of the curve, forcing the last segment to be a straight line. Finally, an interesting idea is to ask that the resulting spline has the same second derivative at the endpoints.

Whatever additional condition we use, we obtain a system of nine linear equations with nine unknowns, and we know how to solve it. It is actually possible to use the eight core equations to determine a solution depending on just one parameter:

$$a_1 = \frac{1}{2}b_1 - \frac{9}{4}, c_1 = 4, a_2 = -b_1, b_2 = b_1, c_2 = 4, a_3 = 1 + b_1, b_3 = -3b_1 - 2, c_3 = 2b_1 + 5.$$

When we decide on the ninth condition, we easily get the coefficients for our spline. In the following pictures we see three quadratic splines, corresponding to the three possibilities for the ninth condition outlined above.



△

The quadratic splines are not used much, most people think of cubic splines when they hear the word spline. There we can also ask for continuity in the second derivative, so the resulting spline function is smooth. This means that at every join of neighboring polynomials we ask for three conditions: that they have the prescribed value, and that their first and second derivatives agree.

**Definition 14b.2.**

Consider data  $(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n) \in \mathbb{R}^2$  with  $x_0 < x_1 < \dots < x_n$ . Let  $n \in \mathbb{N}$ .

A function  $s(x)$  is called a **spline of degree  $n$**  if it

- is continuous on  $[x_0, x_n]$ ,
- satisfies  $s(x_i) = y_i$  for every  $i = 0, 1, \dots, n$ ,
- it is a polynomial of degree at most  $n$  on every interval  $[x_{i-1}, x_i]$  for  $i = 1, \dots, n$ .

For every interval  $[x_{i-1}, x_i]$ , let  $p_i(x)$  denote the corresponding polynomial restriction of  $s(x)$ . Then these polynomials must satisfy the following condition:

- (i)  $p_i(x_{i-1}) = y_{i-1}$  and  $p_i(x_i) = y_i$  for every  $i = 1, \dots, n$ ;
- (ii)  $p'_i(x_i) = p'_{i+1}(x_i)$  for every  $i = 1, \dots, n-1$ ;
- (iii)  $p''_i(x_i) = p''_{i+1}(x_i)$  for every  $i = 1, \dots, n-1$ .

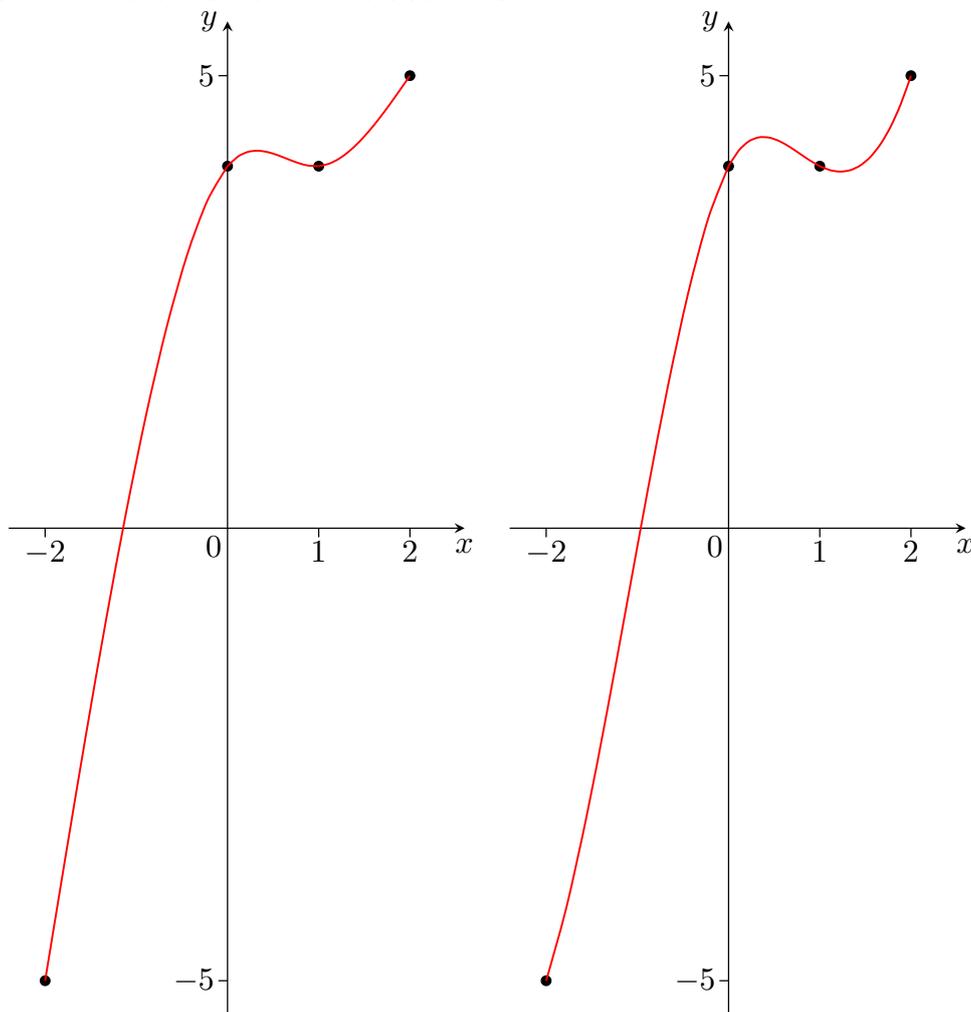
Some people simply say “spline”, so this is the second common meaning of this word.

What is the situation? Given  $n + 1$  data point, we are trying to find  $n$  cubic polynomials, that is, we have  $4n$  coefficients to determine. The condition (i) on values at endpoints provide us with  $2n$  equations. Each of the next two conditions yield  $n - 1$  equations, for the total of  $4n - 2$  equations. To determine the cubic polynomials we need two more conditions. Some popular choices are:

- $p_1''(x_0) = 0$  and  $p_n''(x_n) = 0$  (“natural spline”, for a cubic spline this condition does not force the end segments to be straight);
- $p_1'(x_0) = d_0$  and  $p_n'(x_n) = d_n$  when the values  $d_0, d_n$  of first derivatives at endpoints are known (“clamped spline”);
- $p_1'(x_0) = p_n'(x_n)$  and  $p_1''(x_0) = p_n''(x_n)$  (this one makes good sense when  $y_0 = y_n$ , then we can ask for a periodic spline).

Finding cubic splines intuitively leads to a system of  $4n$  linear equations. More efficient schemes for determining coefficients were found, in particular for the natural spline.

In the picture below we show two cubic splines, the first is the natural cubic spline and the one on the right is the third condition in the above list.



Due to its usefulness, the spline theory has been developed considerably. There are several classifications of splines based on different viewpoints, and a lot of special terminology. An interested reader will easily find many sources on splines.

## 14c. Some other ideas

## 15. (Homogeneous) linear differential equations

There are many kinds of equations: differential, algebraic, Diophantine, recurrent, congruences, etc., and regardless of the kind, the best type to meet is a linear equation. Generally, a linear equation involves the unknown (or unknowns) only in the form of a linear combination. Here the situation is somewhat specific, as the coefficients of our linear combinations are allowed to change with time (or depend on the independent variable in general).

### Definition 15.1.

By a **linear ordinary differential equation of order  $n$**  (LODE) we mean any ODE that can be written in the form

$$y^{(n)} + a_{n-1}(x)y^{(n-1)} + \cdots + a_1(x)y' + a_0(x)y = b(x),$$

where  $a_{n-1}, \dots, a_0, b$  are some functions.

This equation is called **homogeneous** if  $b(x) = 0$ .

For instance, the equation  $y' - 13y = x$  is linear, and also  $y'' + 13y' - e^x y = \ln(x)$  is a linear equation as the coefficient in front of  $y$  is allowed to depend on  $x$ . On the other hand, the equations  $y'' + 3y' - y^2 = x$ ,  $y'' - \sin(y') + y = 0$  and  $y'' + y' \cdot y = 0$  are not linear, as the definition does not allow us to do such things with  $y$  and its derivatives.

In general we prefer to have the highest derivative isolated, and some authors stick to this general pattern also when dealing with linear equations, so they prefer this form:

$$y^{(n)} = a_{n-1}(x)y^{(n-1)} + \cdots + a_1(x)y' + a_0(x)y + b(x).$$

However, the arrangement that we use here seems more practical and is more common. We will sometimes write the general equation as

$$y^{(n)} + \sum_{k=0}^{n-1} a_k(x)y^{(k)} = b(x).$$

for convenience, especially when doing some calculations with this expression. If the reader feels uneasy about some steps in such calculations, it usually helps to rewrite them in the long form as in the definition, then we see better what is happening.

The reader may have noticed a certain inconsistency in our notation. Normally we either write all functions with indicated variable (as in  $f(x) + g(x)$ ) or do not indicate it at all ( $f + g$ ). However, here we write  $y$  without it but the right-hand side  $b$  and the coefficients  $a_i$  do have  $(x)$  next to them. This is not exactly proper, but people do it because it serves a purpose. We do not want to waste time (which is money) on writing  $x$  with  $y$  and its derivatives because everybody knows that it is there, but it is helpful to remind ourselves that the coefficients and the right-hand side do depend on  $x$ . This will be all the more useful later on, when some of these do become constants.

As usual with ODEs, we start with a theorem that reassures us that our attempts to find a solution are not futile. We will be talking about particular solutions, so apart from the differential equation we also need to specify some conditions. We hinted before that when an ODE of order  $n$  is well behaved, then we can expect  $n$  parameters in its general solution, and consequently we need  $n$  initial conditions. We also prefer to ask about situation at a given time (initial time)  $x_0$ , so we essentially have no other choice but to involve derivatives in our conditions. This makes sense, as we saw in chapter 1.

Thus, to obtain a unique solution, we will make a specification for  $y(x_0)$ , and also for  $y'(x_0)$ ,  $y''(x_0)$ ,  $\dots$  until we get  $n$  specifications. It is easy to see that the last one will feature the derivative of order  $n - 1$ . The following theorem, among other things, confirms that this reasoning was sound.

**Theorem 15.2.** (on existence and uniqueness for LODE)

Consider a linear ODE

$$y^{(n)} + a_{n-1}(x)y^{(n-1)} + \cdots + a_1(x)y' + a_0(x)y = b(x). \quad (L)$$

If  $a_{n-1}, \dots, a_0, b$  are continuous on an open interval  $I$ , then for all  $x_0 \in I$  and  $y_0, y_1, \dots, y_{n-1} \in \mathbb{R}$  there exists a solution to the initial value problem

$$(L), y(x_0) = y_0, y'(x_0) = y_1, \dots, y^{(n-1)}(x_0) = y_{n-1}$$

on  $I$  and it is unique there.

The proof of this theorem is not easy, in fact the traditional proof uses the analogous result for systems of first order linear differential equations. We will discuss this in more detail in chapter 26.

When we encountered the existence and uniqueness theorem before, it did not impact directly our exposition. However, this time the above theorem plays a crucial part in proofs of theorems that follow.

All linear equations share similar features and properties, and the statements and their proofs follow along the same path, just the language adjusts to the type of equation at hand. Differential equations are not different, so what follows should be familiar to any student who already met systems of linear equations in a linear algebra class. Here we go.

If linear equations are nice, then homogeneous linear equations are usually even better. We start with the usual structural theorem.

**Theorem 15.3.** (on structure of solution set for hLODE)

Consider a homogeneous linear ODE

$$y^{(n)} + a_{n-1}(x)y^{(n-1)} + \cdots + a_1(x)y' + a_0(x)y = 0. \quad (L)$$

If  $a_i$  are continuous on an open interval  $I$ , then the set of all solutions of this equation on  $I$  is a linear space of dimension  $n$ .

Let  $M$  be the set of all solutions of  $(L)$  on the interval  $I$ .

A linear (or vector) space is, by definition, a set endowed with operations that satisfy certain eight (or seven, depending how one counts them) properties. Our set  $M$  consists of functions on an interval  $I$  and we have natural operations for them, namely addition and multiplication by a number (real in this case, but the theorem also works in a complex setting).

However, rather than checking on all the properties, the preferred way is to show that the investigated set is in fact a subspace of a larger linear overspace. In this case the natural overspace is the space of all functions defined on  $I$ . It is a standard fact that this set with the usual algebraic operations forms a linear space.

There is another interesting candidate. Every solution of our differential equation is, by necessity,  $n$ -times differentiable on  $I$ . It is also common knowledge that the set of all  $n$ -times differentiable functions on an interval, endowed with addition and scalar multiplication, forms a linear space.

We will thus show that  $M$  is a linear space by proving that it is a linear subspace of one of the popular overspaces. Linear algebra tells us how to do it: It is sufficient to confirm that the set  $M$  is closed under the two operations. Some people prefer to do it in one step as follows:

- $\vec{y}_1, \vec{y}_2 \in M, \alpha \in \mathbb{R} \implies (\alpha\vec{y}_1 + \vec{y}_2) \in M.$

Other people prefer to treat each operation separately:

- $\vec{y}_1, \vec{y}_2 \in M \implies (\vec{y}_1 + \vec{y}_2) \in M;$

- $\vec{y} \in M, \alpha \in \mathbb{R} \implies \alpha\vec{y} \in M.$

Choosing between them is a matter of personal preference, we will follow the first approach. The “vectors” are now functions, and we will need to be able to answer two questions. First, how can we tell whether a function  $y$  belongs to our set  $M$ ? The answer is that by the definition of  $M$ , we simply check whether this function is a solution to the equation  $(L)$ . And how do we check

whether it is a solution? We use the general approach: We simply substitute this function into the equation and see whether it becomes a true statement. As usual, it is easier to just substitute into one side and see the other come out of it.

**Proof:** Consider the set  $M$  of all solutions of the equation (L) on the given interval  $I$ . By the existence theorem, this set is not empty.

1. We will show that it is closed under addition and scalar multiplication.

To this end, consider functions  $y_1, y_2 \in M$  and a scalar  $\alpha \in \mathbb{R}$ . We claim that the function  $\alpha y_1 + y_2$  is in  $M$  again, that is, that this function solves the equation (L) on  $I$ . To this end, we substitute  $\alpha y_1 + y_2$  into the left-hand side of the equation and evaluate at any  $x \in I$ , using linearity of differentiation:

$$\begin{aligned} L &= [\alpha y_1 + y_2]^{(n)}(x) + \sum_{i=0}^{n-1} a_i(x)[\alpha y_1 + y_2]^{(i)}(x) \\ &= \alpha y_1^{(n)}(x) + y_2^{(n)}(x) + \sum_{i=0}^{n-1} a_i(x)[\alpha y_1^{(i)}(x) + y_2^{(i)}(x)] \\ &= \alpha y_1^{(n)}(x) + y_2^{(n)}(x) + \sum_{i=0}^{n-1} [\alpha a_i(x)y_1^{(i)}(x) + a_i(x)y_2^{(i)}(x)] \\ &= \alpha y_1^{(n)}(x) + y_2^{(n)}(x) + \sum_{i=0}^{n-1} \alpha a_i(x)y_1^{(i)}(x) + \sum_{i=0}^{n-1} a_i(x)y_2^{(i)}(x) \\ &= \alpha \left[ y_1^{(n)}(x) + \sum_{i=0}^{n-1} a_i(x)y_1^{(i)}(x) \right] + \left[ y_2^{(n)}(x) + \sum_{i=0}^{n-1} a_i(x)y_2^{(i)}(x) \right]. \end{aligned}$$

Now the first group is the left-hand side of our equation (L) with  $y_1$  substituted into it. However, as an element of  $M$ , this function is a solution of (L) on  $I$ , therefore substituting into the  $L$ -side of the equation yields zero. The same argument applies to the second group and function  $y_2$ , which leads us to the conclusion

$$L = \alpha \cdot 0 + 0 = 0,$$

confirming our claim. Note that this proof did not use any advanced theory or tricks, it was just a straightforward application of known principles.

2. Now we want to prove that the dimension of  $M$  is  $n$ . To this end we use the Existence and uniqueness theorem above. Take any  $x_0 \in I$  and consider the following collection of initial conditions:

- $y(x_0) = 1, y'(x_0) = 0, y''(x_0) = 0, \dots, y^{(n-1)}(x_0) = 0;$
- $y(x_0) = 0, y'(x_0) = 1, y''(x_0) = 0, \dots, y^{(n-1)}(x_0) = 0;$
- $y(x_0) = 0, y'(x_0) = 0, y''(x_0) = 1, \dots, y^{(n-1)}(x_0) = 0;$

and so on. In general, the  $i$ th copy of initial conditions has  $y^{(i-1)}(x_0) = 1$  and the other values are zero. The last set of initial conditions is

- $y(x_0) = 0, y'(x_0) = 0, y''(x_0) = 0, \dots, y^{(n-1)}(x_0) = 1$

so there is really  $n$  of them.

Now we apply the existence theorem to the first, second, etc. set of initial collections and obtain solutions of (L) on  $I$ , call them  $y_1, y_2, \dots, y_n$ .

Because  $y_i$  are solutions of (L) on  $I$ , they belong to  $M$ . We claim that they form a linearly independent set. To this end, consider some null linear combination  $\sum \alpha_i y_i = 0$  on  $I$ . This must be in particular true at  $x_0$ , so  $\sum \alpha_i y_i(x_0) = 0$ . However, only one of the functions  $y_i$  has a non-zero value at  $x_0$ , namely  $y_1(x_0) = 1$  (see the initial conditions that were used to create them). The equation  $\sum \alpha_i y_i(x_0) = 0$  therefore reads  $\alpha_1 = 0$ .

When we differentiate the equation  $\sum \alpha_i y_i = 0$ , we get  $\sum \alpha_i y_i' = 0$  on  $I$ . We again substitute  $x_0$  and notice that  $y_2'(x_0) = 1$  is the only non-zero function value there, so  $\alpha_2 = 0$ . Another derivative yields  $\alpha_3 = 0$  and so on, in the end we arrive at the conclusion that all  $\alpha_i$  must be zero. In other words, only the trivial linear combination of  $\{y_i\}$  can produce the zero vector, therefore they are linearly independent. In particular, the dimension of  $M$  is at least  $n$ .

Now we will prove that these  $y_i$  also span the space  $M$ ; so in fact they form a basis. Take any  $y_0 \in M$ , that is, any solution of (L) on  $I$ . Then we can look at values  $\beta_1 = y_0(x_0)$ ,  $\beta_2 = y_0'(x_0)$ ,  $\dots, \beta_n = y_0^{(n-1)}(x_0)$ . Obviously,  $y_0$  is a solution of (L) on  $I$  that satisfies initial conditions  $y^{(i-1)}(x_0) = \beta_i$ .

Now consider the function  $y_c = \sum \beta_i y_i$ . This function is a linear combination of solutions  $y_i$  and we already proved that it therefore must also be a solution of (L) on  $I$ . Moreover, due to our choice of initial conditions that gave rise to  $y_i$ , we have

$$y_c^{(i-1)}(x_0) = \sum \beta_i y_i^{(i-1)}(x_0) = \beta_i \cdot 1 + 0 = \beta_i.$$

Consequently,  $y_c$  solves exactly the same initial value problem as  $y_0$ . By the uniqueness theorem, these two must agree, that is,  $y_0 = \sum \beta_i y_i$  on  $I$ .

We just proved that the set  $\{\vec{y}_i\}$  generates the space  $M$ , so it is in fact a basis and  $\dim(M) = n$ .  $\square$

#### 15.4 Remark (differential equations as mappings):

In section we introduced the idea that the left-hand side of the differential equation determines a mapping, an operator. For instance, for the equation  $y'' - 3y' + 2y = x^{13}$  we have the operator  $L[y] = y'' - 3y' + 2y$ . Here is an example of its action:

$$L[e^{x^2}] = [e^{x^2}]'' - 3[e^{x^2}]' + 2 \cdot e^{x^2} = (4x^2 + 2)e^{x^2} - 3 \cdot 2x e^{x^2} + 2e^{x^2} = (4x^2 - 6x + 4)e^{x^2}.$$

When solving this ODE, we are in fact trying to find functions that would yield value  $x^{13}$  when substituted into this mapping  $L$ . In general, the problem of solving a linear ODE can be seen as a problem of finding a preimage of a certain function  $b(x)$  under the appropriate mapping  $L$ .

The key observation is that when our differential equation is linear, then so is the corresponding mapping. How would we prove it? Linear algebra tells us to investigate  $L[\alpha y_1 + y_2]$ . In the general case of a linear differential equation this means that we should investigate

$$\begin{aligned} L[\alpha y_1 + y_2] &= [\alpha y_1 + y_2]^{(n)}(x) + \sum_{i=0}^{n-1} a_i(x) [\alpha y_1 + y_2]^{(i)}(x) \\ &= \dots = \alpha \left[ y_1^{(n)}(x) + \sum_{i=0}^{n-1} a_i(x) y_1^{(i)}(x) \right] + \left[ y_2^{(n)}(x) + \sum_{i=0}^{n-1} a_i(x) y_2^{(i)}(x) \right] \\ &= \alpha L[y_1] + L[y_2]. \end{aligned}$$

This is essentially the same calculation as we did in the proof of theorem 15.2 above. Indeed, the fact that we can distribute derivatives to terms and rearrange sums is at the heart of this whole topic.

This suggests that perhaps we could do this calculation just once and then somehow refer to it whenever we need it again. And that is exactly what our mapping approach does. A more sophisticated approach to linear differential equations would be as follows: First we introduce this associated mapping  $L$  and prove that it is linear. Then we start with the theorems and many of their proofs can be done in a more efficient manner, because the major manual work is all hidden in the statement that  $L$  is linear. For instance, the calculation in the above proof of closedness of  $M$  would look like this:

$$L[\alpha y_1 + y_2] = \alpha L[y_1] + L[y_2] = \alpha \cdot 0 + 0 = 0.$$

It is a matter of taste, we preferred to start with a hands-on approach. However, now we also

understand the more abstract approach and we will use it to simplify calculations. If the reader prefers, it is always possible to rewrite proofs using the mapping  $L$  into hands-on proofs that manipulate the general form of a linear ODE.

By the way, a more efficient proof of the fact that  $M$  is a linear space is available.  $M$  is the set of all solutions on  $I$ , so it is a set of all functions  $y$  such that  $L[y] = 0$  on  $I$ . This set has a name in linear algebra, it is the kernel of  $L$ . We just argued that  $M = \text{Ker}(L)$  and it is known from linear algebra that the kernel of every linear mapping is a linear subspace. Thus we get this conclusion directly from linear algebra. Incidentally, this is one of the reasons why linear algebra is studied, it is an umbrella theory for many special fields and it can offer quite a few useful results when we realize that our specific situation can be viewed through the eyes of linear algebra. However, linear algebra does not help us with proving that the dimension of  $M$  is  $n$ , that has to be done in a specific way.

Since  $M$  is a subspace, it must include the zero vector, in this case the zero constant function  $y(x) = 0$ . This is no surprise: If we substitute this function into our homogeneous equation, then all the derivatives will be also zero and the equation is obviously satisfied. But it is good to take a note of this.

△

The practical outshot of the structural theorem is that we only need to find some basis of that space and we know it all. A basis in this case means  $n$  functions that are solutions of the given equation and that are linearly independent. What does it mean? The usual, it should not be possible for one function to be expressed as a linear combination of the others. For instance, the set  $\{x, x^2, x^3\}$  is linearly independent on  $\mathbb{R}$ , because no matter what constants  $\alpha, \beta$  we take, the expression  $\alpha x + \beta x^2$  will never be equal to  $x^3$  on  $\mathbb{R}$ , similarly we fail in attempts to express  $x$  or  $x^2$  as some combination of the other two.

On the other hand, the set  $\{13, \sin^2(2x), \cos(2x)\}$  is not linearly independent on  $\mathbb{R}$ , because  $\sin^2(x) = \frac{1}{26} \cdot 13 - \frac{1}{2} \cdot \cos(2x)$ . Formally,

$$\sin^2(x) + \frac{1}{2} \cdot \cos(2x) + \frac{1}{26} \cdot 13 = 0,$$

we have a non-trivial null linear combination.

**Example 15.a:** Consider the differential equation

$$y'' - \frac{x}{2(x-2)}y' + \frac{1}{2(x-2)}y = 0.$$

It is a linear equation of order 2. We see trouble at  $x = 2$ , so we will look for solutions on any interval  $I$  not containing  $x = 2$ . We claim that the set  $\{x, e^{x/2}\}$  is a basis of the space of solutions on any such interval  $I$ .

We start with the basic qualification of the given functions: They should solve the given equation. Indeed, we have

$$L = [x]'' - \frac{x}{2(x-2)}[x]' + \frac{1}{2(x-2)}[x] = 0 - \frac{x}{2(x-2)} \cdot 1 + \frac{1}{2(x-2)} \cdot x = 0$$

and

$$\begin{aligned} L &= [e^{x/2}]'' - \frac{x}{2(x-2)}[e^{x/2}]' + \frac{1}{2(x-2)}[e^{x/2}] = \frac{1}{4}e^{x/2} - \frac{1}{2} \frac{x}{2(x-2)}e^{x/2} + \frac{1}{2(x-2)}e^{x/2} \\ &= e^{x/2} \left[ \frac{1}{4} - \frac{x}{4(x-2)} + \frac{1}{2(x-2)} \right] = 0. \end{aligned}$$

Both calculations are valid for any  $x \neq 2$ .

Second, we need to show that this set is linearly independent. In this case it is actually fairly obvious, the function  $e^{x/2}$  cannot be expressed as  $\alpha x$  for any real number  $\alpha$ , but in other cases this may not be so clear. What is the general approach? We form a linear combination that yields

zero:

$$\alpha x + \beta e^{x/2} = 0$$

and inquire whether this is possible also in a non-trivial way, that is, when not all coefficients are zero. One fruitful approach is to note that this equality is true on some interval  $I$ . We can choose two distinct numbers from this set and substitute for  $x$ , obtaining two equations with two unknowns. For instance, if  $I = (-\infty, 2)$ , we could choose  $x = 0$  and  $x = 1$ , obtaining equations

$$\begin{aligned}\alpha \cdot 0 + \beta \cdot 1 &= 0 \\ \alpha \cdot 1 + \beta \cdot e^{1/2} &= 0.\end{aligned}$$

From the first we get  $\beta = 0$ , substituting into the second we get  $\alpha = 0$ . This means that the only way to get a zero linear combination is the trivial combination, and therefore the set in question is linearly independent.

The last property of a basis is that the two functions span the space of solutions. Since we do not know (yet) how to find those solutions, we would not know how to go about this. Fortunately, there is an alternative. We know that the dimension of the space of solutions is 2 and we have two functions in our linearly independent set, so it must be a basis.

We conclude that  $\{x, e^{x/2}\}$  is a basis for our space  $M$  of solutions.

Now the payoff. Any element of the linear space  $M$  can be obtained as a linear combination of the basis. This means that we get a universal formula for all solutions, and we can say that a general solution of our equation is

$$y(x) = ax + be^{x/2}, \quad x \neq 2.$$

Typically, we would choose between the intervals  $(-\infty, 2)$  and  $(2, \infty)$ .

Remark: How did I find this basis? I cheated; first I chose two independent functions and then I dreamed up a suitable equation for them. In fact, I do not know of any general procedure for solving linear equations, and in particular I do not know how I would go about solving the one in this example. As you will see below, we will have to restrict our attention to very nice linear equations to make any progress. Yes, differential equations are tough.

△

This example shows why we talk about bases. They are the key: Once we know a basis, we know all solutions. Due to their importance, ODE people even use a better-sounding name for bases.

**Definition 15.5.**

Consider a homogeneous linear ODE

$$y^{(n)} + a_{n-1}(x)y^{(n-1)} + \dots + a_1(x)y' + a_0(x)y = 0.$$

By a **fundamental system of solutions** of this equation on  $I$  we mean any basis of the space of all solutions of this equation on  $I$ .

Finding a basis generally means two things: Finding an appropriate number of solutions (more on that below) and proving that they are linearly independent. This is not always easy when done by definition. Fortunately, there is a practical tool. We will need to look at a certain object.

**Definition 15.6.**

Let  $y_1, y_2, \dots, y_n$  be  $(n - 1)$ -times differentiable functions. We define their **Wronskian** as

$$W(x) = \det \begin{vmatrix} y_1(x) & y_2(x) & \dots & y_n(x) \\ y_1'(x) & y_2'(x) & \dots & y_n'(x) \\ \vdots & \vdots & \dots & \vdots \\ y_1^{(n-1)}(x) & y_2^{(n-1)}(x) & \dots & y_n^{(n-1)}(x) \end{vmatrix}.$$

Note that we first create a matrix whose entries are functions, and then we find its determinant, so the Wronski matrix is in fact a function on  $I$  again. Now for the result. The traditional form is as follows.

**Theorem 15.7.**

Consider a homogeneous linear ODE

$$y^{(n)} + a_{n-1}(x)y^{(n-1)} + \dots + a_1(x)y' + a_0(x)y = 0,$$

where  $a_i$  are continuous on an open interval  $I$ . Let  $y_1, y_2, \dots, y_n$  be solutions of this equation on  $I$ , let  $W$  be their Wronskian.

These functions form a linearly independent set (and thus a fundamental system) if and only if  $W(x) \neq 0$  on  $I$ , which is if and only if  $W(x_0) \neq 0$  for some  $x_0 \in I$ .

Formally, the theorem says that the following properties are mutually equivalent:

- (a)  $\{y_1, \dots, y_n\}$  is LI on  $I$ ;
- (b)  $W(x) \neq 0$  on  $I$ ;
- (c)  $W(x_0) \neq 0$  for some  $x_0 \in I$ .

Note that one direction is obvious, namely (b)  $\implies$  (c). The other direction is unusual: If this function is not zero somewhere, then it is never zero. It is true only thanks to the assumption that those  $n$  functions  $y_i$  happen to solve the same homogeneous linear ODE.

**Proof:** To prove mutual equivalence of three statements, it is enough to prove three implications that tie them into a circle. It turns out that the two that are not obvious are best approached via indirect proofs, that is, by passing to contrapositives. So in fact we will prove mutual equivalence of the following three negations:

- (a') The set  $\{y_1, \dots, y_n\}$  is linearly dependent;
- (b')  $W(x_0) = 0$  for some  $x_0 \in I$ .
- (c')  $W(x) = 0$  on  $I$ ;

We will tie up the three statements into a circle.

1) First we will prove that (a')  $\implies$  (c'). We assume that functions  $y_1, \dots, y_n$  are linearly dependent as functions on  $I$ . From this it follows that there must be coefficients  $\alpha_i$  such that  $\sum \alpha_i y_i(x) = 0$  on  $I$  and at least one of these coefficients is not zero. Since the order of these functions is not crucial, we can always reorder them so that in fact  $\alpha_1 \neq 0$ . Dividing the equality by this non-zero number, and denoting  $\beta_i = \frac{\alpha_i}{\alpha_1}$  we find that  $y_1(x) + \sum_{i=2}^n \beta_i y_i(x) = 0$  on  $I$ . This equality can be repeatedly differentiated, obtaining

$$y_1'(x) + \sum_{i=2}^n \beta_i y_i'(x) = 0$$

⋮

$$y_1^{(n-1)}(x) + \sum_{i=2}^n \beta_i y_i^{(n-1)}(x) = 0$$

Now consider the Wronski matrix. To its first column we can add the second column multiplied by  $\beta_2$ , the third multiplied by  $\beta_3$  etc., in general the  $i$ th column multiplied by  $\beta_i$ . These are equivalent operations that do not influence the outcome of determinant. Therefore for every

$x \in I$  we have

$$\begin{aligned}
 W(x) &= \det \begin{vmatrix} y_1(x) & y_2(x) & \dots & y_n(x) \\ y_1'(x) & y_2'(x) & \dots & y_n'(x) \\ \vdots & \vdots & & \vdots \\ y_1^{(n-1)}(x) & y_2^{(n-1)}(x) & \dots & y_n^{(n-1)}(x) \end{vmatrix} \\
 &= \det \begin{vmatrix} y_1(x) + \sum_{i>1} \beta_i y_i(x) & y_2(x) & \dots & y_n(x) \\ y_1'(x) + \sum_{i>1} \beta_i y_i'(x) & y_2'(x) & \dots & y_n'(x) \\ \vdots & \vdots & & \vdots \\ y_1^{(n-1)}(x) + \sum_{i>1} \beta_i y_i^{(n-1)}(x) & y_2^{(n-1)}(x) & \dots & y_n^{(n-1)}(x) \end{vmatrix} \\
 &= \det \begin{vmatrix} 0 & y_2(x) & \dots & y_n(x) \\ 0 & y_2'(x) & \dots & y_n'(x) \\ \vdots & \vdots & & \vdots \\ 0 & y_2^{(n-1)}(x) & \dots & y_n^{(n-1)}(x) \end{vmatrix} = 0.
 \end{aligned}$$

This proves the claim.

2) Obviously, if  $W(x) = 0$  on  $I$ , then it must also be zero at some point of  $I$ . This proves (c')  $\implies$  (b').

3) To conclude the circle, we need to prove that if  $W(x_0) = 0$  for some  $x_0 \in I$ , then the functions are dependent. The assumption means that the Wronski matrix with  $x_0$  substituted in is a singular matrix, hence its columns are linearly dependent. Therefore there are numbers  $\alpha_i$ , not all of them being zero, such that

$$\sum_{i=1}^n \alpha_i \begin{bmatrix} y_i(x_0) \\ y_i'(x_0) \\ \vdots \\ y_i^{(n-1)}(x_0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

In other words,

$$\begin{bmatrix} \sum_{i=1}^n \alpha_i y_i(x_0) \\ \sum_{i=1}^n \alpha_i y_i'(x_0) \\ \vdots \\ \sum_{i=1}^n \alpha_i y_i^{(n-1)}(x_0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Consider the function  $y_0(x) = \sum_{i=1}^n \alpha_i y_i(x)$ . The above observation shows that  $y_0(x_0) = 0$ ,  $y_0'(x_0) = 0, \dots, y_0^{(n-1)}(x_0) = 0$ .

By our main assumption, the functions  $y_i$  solve the given homogeneous linear ODE ( $L$ ) on  $I$ , so every linear combination of them, including this  $y_0$ , solves this ODE on  $I$  as well.

Consider the following initial conditions:

$$y(x_0) = 0, y'(x_0) = 0, \dots, y^{(n-1)}(x_0) = 0 \quad (\text{IC}).$$

We see that the function  $y_0$  solves the initial value problem ( $L$ ),(IC) on  $I$ .

On the other hand, the constant function  $y_c(x) = 0$  also solves the same initial value problem on  $I$ . By the uniqueness theorem 15.2, this initial value problem can have only one solution on  $I$ , which means that  $y_0 = y_c$  on  $I$ . In other words,  $\sum_{i=1}^n \alpha_i y_i(x) = 0$  on  $I$ . We also know that at least

some of the coefficients  $\alpha_i$  are not zero, which means that we found a non-trivial linear combination of  $y_i$  that yields zero on  $I$ . Therefore the functions  $y_i$  form a linearly dependent set on  $I$ .  $\square$

It is useful to observe that in the proof of the implication (a')  $\implies$  (c') we did not use one of the assumptions, namely that the functions  $y_i$  all solve the given ODE. This means that the implication (a)  $\implies$  (c) from the statement of the theorem is valid for just any functions on  $I$ . By a remarkable coincidence, that is exactly the implication that we find most useful when solving differential equation. It is therefore worth stating formally.

**Corollary 15.8.**

Consider functions  $y_1, \dots, y_n$  that are  $n - 1$  times continuously differentiable on an open interval  $I$ . Let  $W$  be their Wronskian. If  $W(x_0) \neq 0$  for some  $x_0 \in I$ , then the functions  $y_1, \dots, y_n$  are linearly independent on  $I$ .

We actually do not use theorem 15.7 or this corollary much in practical calculations. Rather, it will be embedded in proofs of convenient theorems that will provide us with complete answers. But to appreciate how it works, here is an example that will come handy later.

**Example 15.b:** We are given solutions  $e^x$  and  $e^{2x}$  of the differential equation  $y'' - 3y' + 2y = 0$  on  $\mathbb{R}$ . Are they linearly independent?

First, we should not trust everything strangers tell us, so we check that these two functions indeed solve the given equation on  $\mathbb{R}$ . To get some practice, we will check on the first one in the straightforward way: We substitute into the left-hand side and make our way to the right-hand side of the equation (namely zero). For the second function we use the mapping point of view, and we will see that it is just a handy wrapper for natural calculations.

$$\begin{aligned} L[e^x] &= [e^x]'' - 3[e^x]' + 2e^x = e^x - 3e^x + 2e^x = 0, \\ L[e^{2x}] &= [e^{2x}]'' - 3[e^{2x}]' + 2e^{2x} = 4e^{2x} - 3 \cdot 2e^{2x} + 2e^{2x} = 0. \end{aligned}$$

Okay, so they were right this time, let's move to independence. First we form the matrix in question and find the Wronskian.

$$\begin{aligned} \begin{pmatrix} e^x & e^{2x} \\ [e^x]' & [e^{2x}]' \end{pmatrix} &= \begin{pmatrix} e^x & e^{2x} \\ e^x & 2e^{2x} \end{pmatrix} \\ \implies W(x) &= 2e^{3x} - e^{3x} = e^{3x}. \end{aligned}$$

We can easily find some  $x$  where  $W(x) \neq 0$ , for instance  $x = 13$ , so by the above theorem, the set  $\{e^x, e^{2x}\}$  is linearly independent. The numbers match (two functions for second degree equation), so  $\{e^x, e^{2x}\}$  is in fact a fundamental system of that equation.

Consequently,  $y(x) = a e^x + b e^{2x}$ ,  $x \in \mathbb{R}$  is a general solution of that equation.

$\triangle$

We are ready for the key question: Where do we actually get those solutions? Although linear equations tend to be the nicest of all, this is not enough in the world of differential equations. In order to be able to solve them, we need to restrict our attention to a special case where the coefficients in front of  $y^{(i)}$  do not depend on  $x$ .

**Definition 15.9.**

By a **linear ODE with constant coefficients** we mean any linear ODE for which  $a_0(x) = a_0$ ,  $a_1(x) = a_1$ ,  $\dots$ ,  $a_{n-1}(x) = a_{n-1}$  are constant functions.

Fortunately, many practical equations fall into this group, so this is not too restrictive. To solve these equations we first change them into polynomials.

**Definition 15.10.**

Consider a homogeneous linear ODE with constant coefficients

$$y^{(n)} + a_{n-1}y^{(n-1)} + \cdots + a_1y' + a_0y = 0.$$

We define its **characteristic polynomial** as

$$p(\lambda) = \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0.$$

We define its **characteristic equation** as  $p(\lambda) = 0$ . The solutions of this equation are called **characteristic numbers** or **eigenvalues** of the given ODE.

**Example 15.c:** Consider the differential equation  $y'' - 3y' + 2y = 0$ . We create its characteristic polynomial by turning derivatives into powers, recalling that  $y = y^{(0)}$ :

$$p(\lambda) = \lambda^2 - 3\lambda + 2 = \lambda^2 - 3\lambda + 2.$$

To find characteristic numbers of the given equation we find roots of  $p(\lambda)$ :

$$\lambda^2 - 3\lambda + 2 = 0 \implies (\lambda - 1)(\lambda - 2) = 0 \implies \lambda = 1, 2.$$

△

By a remarkable coincidence, in example 15.b we were supplied with solutions  $e^{1 \cdot x}$ ,  $e^{2 \cdot x}$  to this very equation. This is not an accident.

**Fact 15.11.**

Consider a homogeneous linear ODE with constant coefficients

$$y^{(n)} + a_{n-1}y^{(n-1)} + \cdots + a_1y' + a_0y = 0.$$

Let  $\lambda_0$  be its characteristic number. Then  $y(x) = e^{\lambda_0 x}$  is a solution of this equation on  $\mathbb{R}$ .

If  $\lambda_1, \dots, \lambda_N$  are distinct characteristic numbers of this equation, then  $\{e^{\lambda_1 x}, \dots, e^{\lambda_N x}\}$  is a linearly independent set of solutions.

**Proof:** Assume that  $\lambda_0$  is a characteristic number of the given equation. We check what happens when we substitute the corresponding exponential function  $e^{\lambda_0 x}$  into the left-hand side of that equation. For any  $x \in \mathbb{R}$  we have

$$\begin{aligned} L &= [e^{\lambda_0 x}]^{(n)} + a_{n-1}[e^{\lambda_0 x}]^{(n-1)} + \cdots + a_1[e^{\lambda_0 x}]' + a_0[e^{\lambda_0 x}] \\ &= \lambda_0^n e^{\lambda_0 x} + a_{n-1}\lambda_0^{n-1}e^{\lambda_0 x} + \cdots + a_1\lambda_0 e^{\lambda_0 x} + a_0 e^{\lambda_0 x} \\ &= e^{\lambda_0 x}[\lambda_0^n + a_{n-1}\lambda_0^{n-1} + \cdots + a_1\lambda_0 + a_0] = e^{\lambda_0 x}p(\lambda_0). \end{aligned}$$

In the last step we made a key observation that the polynomial in the brackets happens to be exactly the characteristic polynomial associated with the given equation. Now  $\lambda_0$  is a characteristic number of that equation, which by definition means that it is a root of  $p$ . Thus

$$L = e^{\lambda_0 x} \cdot 0 = 0.$$

To prove linear independence we use the corollary 15.8. We easily find the Wronski determinant.

$$\begin{aligned} W(x) &= \det \begin{vmatrix} e^{\lambda_1 x} & e^{\lambda_2 x} & \cdots & e^{\lambda_N x} \\ \lambda_1 e^{\lambda_1 x} & \lambda_2 e^{\lambda_2 x} & \cdots & \lambda_N e^{\lambda_N x} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1^{N-1} e^{\lambda_1 x} & \lambda_2^{N-1} e^{\lambda_2 x} & \cdots & \lambda_N^{N-1} e^{\lambda_N x} \end{vmatrix} \\ &= e^{\lambda_1 x} e^{\lambda_2 x} \cdots e^{\lambda_N x} \cdot \det \begin{vmatrix} 1 & 1 & \cdots & 1 \\ \lambda_1 & \lambda_2 & \cdots & \lambda_N \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1^{N-1} & \lambda_2^{N-1} & \cdots & \lambda_N^{N-1} \end{vmatrix}. \end{aligned}$$

Since the exponential functions are never zero, everything depends on the real-valued matrix. It is known in linear algebra under the name Vandermonde matrix, and it is also known that its determinant is not zero if the lambdas are distinct.

□

Now let's count. If we have a linear ODE with constant coefficients of order  $n$ , then its characteristic polynomial has degree  $n$  as well, therefore it has  $n$  roots. If these roots are distinct, then by the above Fact we obtain  $n$  linearly independent exponential functions that solve this equation, which just matches our requirements for a basis. In other words, this fact provides us with a direct way to determine fundamental systems.

Indeed, we observed that  $\lambda = 1, 2$  are characteristic numbers of the equation  $y'' - 3y' + 2y = 0$  and earlier we proved that the set  $\{e^{1 \cdot x}, e^{2 \cdot x}\}$  is a fundamental system. Everything fits.

**Example 15.d:** We will find a general solution of the equation

$$y''' + y'' - 6y' = 0.$$

To this end, we need to find some fundamental system. We start by writing the characteristic equation, we turn derivatives into powers:

$$\lambda^3 + \lambda^2 - 6\lambda = 0 \implies \lambda^3 + \lambda^2 - 6\lambda = 0.$$

When we write it as

$$0 = \lambda(\lambda^2 + \lambda - 6) = \lambda(\lambda + 3)(\lambda - 2),$$

we easily identify the roots  $\lambda = 0, -3, 2$ . These are the characteristic numbers. According to the fact above, the functions  $e^{0 \cdot x} = e^0 = 1$ ,  $e^{-3 \cdot x} = e^{-3x}$ ,  $e^{2 \cdot x} = e^{2x}$  are solutions of the given equation, and moreover, the set  $\{1, e^{-3x}, e^{2x}\}$  is linearly independent. Since the space of solutions is three-dimensional (see the order of the equation), this is in fact a basis, that is, a fundamental system. All solutions can therefore be captured with the formula

$$y(x) = a \cdot 1 + b \cdot e^{-3x} + c \cdot e^{2x} = a + b e^{-3x} + c e^{2x}.$$

It is automatically assumed that the coefficients  $a, b, c$  can be arbitrary real numbers. We also observe that no values of  $x$  are ruled out by the equation or the process, and we obtain our general solution

$$y(x) = a + b e^{-3x} + c e^{2x}, \quad x \in \mathbb{R}.$$

That was really easy. Still, students sometimes mess up, and the single most frequent mistake is this: Students get used to put decreasing powers of  $\lambda$ . They see  $y'' - 13y' + 23y = 0$ , they write  $\lambda^2 - 13\lambda + 23 = 0$ . So far so good. Then they encounter  $y'' - 13y = 0$  and sometimes write  $\lambda^2 - 13\lambda = 0$  automatically. Do you see the mistake? It should be  $\lambda^2 - 13 = 0$ , we skipped one derivative in the equation, hence we also have to skip one power in our characteristic polynomial. Now you have been warned, so try not to do it.

△

There are several things that can go wrong. First, it may easily happen that the roots are not distinct, that is, that some of them have higher multiplicity.

**Example 15.e:** Consider the equation  $y'' - 4y' + 4y = 0$ . We solve the characteristic equation:

$$0 = \lambda^2 - 4\lambda + 4 = (\lambda - 2)^2$$

to learn that there is one characteristic number  $\lambda = 2$  with multiplicity 2. By the Fact, the function  $e^{2x}$  is a solution to our equation, this we easily confirm. However, the space of solutions is two-dimensional, where do we get another vector for our basis?

Sometimes students would claim that  $\{e^{2x}, e^{2x}\}$  is a basis, or that  $y(x) = a e^{2x} + b e^{2x}$  is a general solution, but this does not work. We know from linear algebra that taking the same vector

twice does not create a two-dimensional space. After all, the formula above can be rewritten as  $y(x) = (a + b)e^{2x} = ce^{2x}$ , we do not have two independent degrees of freedom here.

Obviously, we need to learn something new. Perhaps another Fact would help.

△

**Fact 15.12.**

Consider a homogeneous linear ODE with constant coefficients

$$y^{(n)} + a_{n-1}y^{(n-1)} + \cdots + a_1y' + a_0y = 0.$$

Let  $\lambda_0$  be its characteristic number with multiplicity  $m$ .

Then  $e^{\lambda_0 x}, xe^{\lambda_0 x}, \dots, x^{m-1}e^{\lambda_0 x}$  are solutions of this equation and they form a linearly independent set.

Note that if we stop with the power  $x^{m-1}$ , we actually create  $m$  functions, which is exactly the number that we need to find for a characteristic number of multiplicity  $m$ . Things fit nicely again.

**Proof:** We will show the claim for a characteristic number of multiplicity 2. For higher multiplicities the idea of the proof is analogous, just the calculations are way more complicated.

Assume that  $\lambda_0$  is a characteristic number of a given homogeneous linear ODE with constant coefficients. This means that it is a root of its characteristic polynomial  $p(\lambda)$ . By the previous fact, the function  $e^{\lambda_0 x}$  solves the given ODE. We need to show that also the function  $y(x) = xe^{\lambda_0 x}$  is a solution.

To this end we first observe that

$$\begin{aligned} y'(x) &= e^{\lambda_0 x} + \lambda_0 x e^{\lambda_0 x}, \\ y''(x) &= \lambda_0 e^{\lambda_0 x} + \lambda_0 e^{\lambda_0 x} + \lambda_0^2 x e^{\lambda_0 x} = 2\lambda_0 e^{\lambda_0 x} + \lambda_0^2 x e^{\lambda_0 x}, \\ y'''(x) &= 2\lambda_0^2 e^{\lambda_0 x} + \lambda_0^2 e^{\lambda_0 x} + \lambda_0^3 x e^{\lambda_0 x} = 3\lambda_0^2 e^{\lambda_0 x} + \lambda_0^3 x e^{\lambda_0 x}, \dots, \\ y^{(i)}(x) &= i\lambda_0^{i-1} e^{\lambda_0 x} + \lambda_0^i x e^{\lambda_0 x}. \end{aligned}$$

We use this to work out the left-hand side of the given equation after substituting  $xe^{\lambda_0 x}$  into it:

$$\begin{aligned} L &= [xe^{\lambda_0 x}]^{(n)} + \sum_{i=0}^{n-1} a_i [xe^{\lambda_0 x}]^{(i)} \\ &= [n\lambda_0^{n-1} e^{\lambda_0 x} + \lambda_0^n x e^{\lambda_0 x}] + \sum_{i=0}^{n-1} a_i [i\lambda_0^{i-1} e^{\lambda_0 x} + \lambda_0^i x e^{\lambda_0 x}] \\ &= n\lambda_0^{n-1} e^{\lambda_0 x} + \lambda_0^n x e^{\lambda_0 x} + \sum_{i=0}^{n-1} a_i i\lambda_0^{i-1} e^{\lambda_0 x} + \sum_{i=0}^{n-1} a_i \lambda_0^i x e^{\lambda_0 x} \\ &= e^{\lambda_0 x} \left[ n\lambda_0^{n-1} + \sum_{i=0}^{n-1} a_i i\lambda_0^{i-1} \right] + x e^{\lambda_0 x} \left[ \lambda_0^n + \sum_{i=0}^{n-1} a_i \lambda_0^i \right] \end{aligned}$$

The second bracketed expression is the characteristic polynomial  $p(\lambda)$  with  $\lambda_0$  substituted into it, just like in the proof of the previous fact. Interestingly, the first bracketed expression happens to be the derivative of  $p$ , again with  $\lambda_0$  substituted in. The fact that  $\lambda_0$  is a root of  $p(\lambda)$  of higher multiplicity means that it is also the root of  $p'(\lambda)$ . Hence we obtain

$$L = e^{\lambda_0 x} p'(\lambda_0) + x e^{\lambda_0 x} p(\lambda_0) = e^{\lambda_0 x} \cdot 0 + x e^{\lambda_0 x} \cdot 0 = 0$$

as needed.

It remains to show that the set  $\{e^{\lambda_0 x}, xe^{\lambda_0 x}\}$  is linearly independent on  $\mathbb{R}$ . This is very simple using the Wronskian:

$$W(x) = \det \begin{vmatrix} e^{\lambda_0 x} & x e^{\lambda_0 x} \\ \lambda_0 e^{\lambda_0 x} & e^{\lambda_0 x} + \lambda_0 x e^{\lambda_0 x} \end{vmatrix} = e^{2\lambda_0 x} \neq 0.$$

The general case would use the fact that a root  $x_0$  of multiplicity  $m$  of polynomial  $p$  (or a function in general) satisfies  $p(x_0) = 0$ ,  $p'(x_0) = 0$ ,  $\dots$ ,  $p^{(m-1)}(x_0) = 0$ .

□

**Example 15.f:** Consider the equation  $y'' - 4y' + 4y = 0$  again. Looking for its characteristic numbers we find  $\lambda = 2$  with multiplicity 2. According to the new and improved Fact, also  $x e^{2x}$  is a solution and it is independent of  $e^{2x}$ . Thus  $\{e^{2x}, x e^{2x}\}$  is a fundamental system and we obtain a general solution of our equation in the form

$$y(x) = a e^{2x} + b x e^{2x}, \quad x \in \mathbb{R}.$$

△

**15.13 Remark on general solutions:** Note that we talk here of “a” general solution. The reason is simple, once there is one general solution, then there are infinitely many formulas that capture all solutions. For instance, returning to example 15.f, one can easily check that  $\{e^{2x}, (x - 13)e^{2x}\}$  is also a basis of the same space of solutions, and therefore the formula

$$y(x) = \tilde{a} e^{2x} + \tilde{b}(x - 13)e^{2x}, \quad x \in \mathbb{R}$$

is also a general solution.

These are two distinct expressions, but when we let the parameters run through all their possible values, we find that both expressions give rise to identical sets of solutions. In other words, every solution of the given equation can be reached by both formulas, we just have to use different values of coefficients to get there.

For instance,  $13e^{2x} + 2x e^{2x}$  is a solution of our equation, and we can see it as

$$\begin{aligned} 13e^{2x} + 2x e^{2x} &= 4 \cdot e^{2x} + 2 \cdot x e^{2x} & a = 13, \quad b = 2; & \quad \text{or} \\ 13e^{2x} + 2x e^{2x} &= 36 \cdot e^{2x} + 2 \cdot (x - 13)e^{2x} & \tilde{a} = 36, \quad \tilde{b} = 2; \end{aligned}$$

One can actually find a relationship between the two sets of parameters when they try to provide us with the same solution:

$$\tilde{a} e^{2x} + \tilde{b}(x - 13)e^{2x} = \tilde{a} e^{2x} + \tilde{b} x e^{2x} - 13\tilde{b} e^{2x} = (\tilde{a} - 13\tilde{b})e^{2x} + \tilde{b} x e^{2x}.$$

We can again turn to linear algebra for help when we are given two formulas with parameters and we wonder whether they describe the same space of solutions. Assuming that each formula is correct, that is, that each is a linear combination of linearly independent vectors (functions), then it is enough to check that dimensions fit (both formulas have the same number of parameters and then show that functions from one basis can be expressed by the other general formula.

△

In general, we can have several distinct characteristic numbers, each with a certain multiplicity. Thus for each distinct characteristic number we obtain a chain of functions based on the same exponential and the Fact guarantees that this whole chain is linearly independent. However, then we have to put all these chains into one set. Is it still linearly independent? The answer is in the positive, but we will not confirm it with a formal statement yet, we will wait until we have a complete picture.

There is still one case to address, namely when some characteristic numbers happen to be complex. Actually, the facts above also apply to complex characteristic numbers, so we could work with solutions of the form  $e^{(\alpha + \beta i)x}$ . However, when a problem is given in the real setting, we also expect answers to be real. Fortunately, this can be arranged.

When a polynomial over real numbers has a complex root  $\lambda = \alpha + \beta i$ , then also its conjugate number  $\bar{\lambda} = \alpha - \beta i$  is a root of the same polynomial and with the same multiplicity. Thus all complex roots come in pairs. This in particular means that when we have a complex characteristic number  $\lambda = \alpha + \beta i$ , then we have to account for a two-dimensional subspace of the space of

solutions that is connected with the pair  $\alpha \pm \beta i$ . We do it as follows. For the paired characteristic numbers we have two solutions:

$$\begin{aligned} f_1 &= e^{(\alpha+\beta i)x} = e^{\alpha x + \beta x i} = e^{\alpha x} [\cos(\beta x) + i \sin(\beta x)], \\ f_2 &= e^{(\alpha-\beta i)x} = e^{\alpha x - \beta x i} = e^{\alpha x} [\cos(-\beta x) + i \sin(-\beta x)] = e^{\alpha x} [\cos(\beta x) - i \sin(\beta x)]. \end{aligned}$$

In the last step we used the fact that the cosine is an even function, while the sine is odd.

All linear combinations of  $f_1, f_2$  are also solutions of the given equation, in particular the following two:

$$\begin{aligned} \frac{1}{2}(f_1 + f_2) &= e^{\alpha x} \cos(\beta x), \\ \frac{1}{2i}(f_1 - f_2) &= e^{\alpha x} \sin(\beta x). \end{aligned}$$

These two functions are linearly independent and span the same subspace as the original functions. This is actually something that is known from linear algebra, where for two linearly independent vectors we have  $\text{span}\{\vec{x}_1, \vec{x}_2\} = \text{span}\{\vec{x}_1 + \vec{x}_2, \vec{x}_1 - \vec{x}_2\}$  and the new pair is also linearly independent.

There is another way to derive these two real solutions. Note that we in fact have the following:

$$\begin{aligned} e^{\alpha x} \cos(\beta x) &= \text{Re}(e^{(\alpha+\beta i)x}), \\ e^{\alpha x} \sin(\beta x) &= \text{Im}(e^{(\alpha+\beta i)x}). \end{aligned}$$

There is a general rule that says that if  $y$  is a solution of a linear differential equation, then its real and imaginary parts also solve this equation. This point of view will come handy in later chapters. For now the important message is that when we see a complex eigenvalue  $\alpha + \beta i$ , we simply use its real part to create an exponential (which is something that we also do for real characteristic numbers, so nothing new here), and we use the imaginary part to turn set up a version with cosine and another with sine.

We obtain in this way a two-dimensional space, so we also accounted for the conjugate number  $\alpha - \beta i$ . Indeed, if we apply the same trick to this other number, we obtain functions  $e^{\alpha x} \cos(\beta x)$  and  $-e^{\alpha x} \sin(\beta x)$  that span the same subspace. Thus there is no point of actually working with  $\alpha - \beta i$ . The conclusion is that we always treat the pair  $\alpha \pm \beta i$  together, use one to create two functions and that is it. If this pair happens to have higher multiplicity as roots of the characteristic polynomial, we use the same trick that we learn before and we start multiplying the pair  $e^{\alpha x} \cos(\beta x), e^{\alpha x} \sin(\beta x)$  by  $x$ , then by  $x^2$  and so on until we have enough functions. If the numbers  $\alpha \pm \beta i$  has multiplicity 2, then we need to create  $2m$  functions for our basis, and thus the chain of functions ends with  $x^{m-1} e^{\alpha x} \cos(\beta x), x^{m-1} e^{\alpha x} \sin(\beta x)$ , which is exactly the same power  $x^{m-1}$  as we had when handling just one real characteristic number.

We can wrap it up. We first make a list of all distinct eigenvalues. For every real eigenvalue we take the corresponding exponential, for each pair of conjugated complex eigenvalues we take a pair of exponentials with cosine and sine. Next, for all characteristic numbers of higher multiplicity we take the function or functions associated with it, and add a suitable number of copies multiplied by increasing powers of  $x$ . Then we put it all into one big set and the numbers should match the order of our equation. Fortunately, the resulting set will automatically be linearly independent, and we have our fundamental system.

**Theorem 15.14.** (on fundamental system for LODE with constant coefficients)

Consider a homogeneous linear ODE with constant coefficients

$$y^{(n)} + a_{n-1}y^{(n-1)} + \cdots + a_1y' + a_0y = 0.$$

Let  $\lambda$  be its characteristic number of multiplicity  $m$ .

(1) If  $\lambda = \alpha \in \mathbb{R}$ , then  $e^{\alpha x}$ ,  $x e^{\alpha x}$ ,  $\dots$ ,  $x^{m-1}e^{\alpha x}$  are solutions of the associated homogeneous equation on  $\mathbb{R}$  and they are linearly independent.

(2) If  $\lambda = \alpha \pm \beta i \in \mathbb{C}$  with  $\beta \neq 0$ , then  $e^{\alpha x} \cos(\beta x)$ ,  $x e^{\alpha x} \cos(\beta x)$ ,  $\dots$ ,  $x^{m-1}e^{\alpha x} \cos(\beta x)$ ,  $e^{\alpha x} \sin(\beta x)$ ,  $x e^{\alpha x} \sin(\beta x)$ ,  $\dots$ ,  $x^{m-1}e^{\alpha x} \sin(\beta x)$  are solutions of the associated homogeneous equation on  $\mathbb{R}$  and they are linearly independent.

(3) The set of functions from (1) and (2) for all distinct characteristic numbers is linearly independent and it forms a fundamental system of the given equation on  $\mathbb{R}$ .

Now we have all cases covered. However, there is a third possible complication. If the differential equation is of order three or more, then we may have trouble finding the characteristic numbers. In fact, for polynomials of degree over four there is no general procedure for finding roots. What can we do then? We have to turn to numerical analysis and ask for an approximate solution to the characteristic equations. And we will do just that. There are chapters coming up on solving algebraic equations numerically, after we settle with linear differential equations.

Fortunately, in school we usually meet equations of lower degree (typically two, sometimes three), and what's more, surprisingly often their characteristic polynomials have very nice roots. Thus we should be able to solve all problems that we meet. In the next example we will actually show a problem that one would not expect on a test; it is obviously artificial, designed to exhibit the salient features of this chapter. Enjoy it, you will never see anything like this again.

### Example 15.g:

We will find a general solution of  $y^{(8)} - y^{(7)} - y^{(4)} + y^{(3)} = 0$ .

Of course, we normally write these derivatives as  $y^{(8)}$  and  $y^{(7)}$ , but when you have just one chance in life to do something crazy (and safe), you should go for it.

The characteristic polynomial is

$$p(\lambda) = \lambda^8 - \lambda^7 - \lambda^4 + \lambda^3 = \lambda^3(\lambda^5 - \lambda^4 - \lambda + 1).$$

We immediately see that  $\lambda = 0$  is a triple characteristic number. Now we need to solve the equation

$$\lambda^5 - \lambda^4 - \lambda + 1 = 0$$

and as we commented above, there is no formula for its roots. However, it is a school problem, so we use the usual approach and simply guess some nice root. A bit of experimentation shows that  $\lambda = 1$  works, so we can factor out the corresponding factor:

$$\lambda^5 - \lambda^4 - \lambda + 1 = (\lambda - 1)(\lambda^4 - 1).$$

One can use the long division for this or perhaps the Horner scheme from linear algebra. Now there seem to be two natural paths to follow. One possibility is to observe that  $\lambda = 1$  is also a root of the fourth degree polynomial, factor out the corresponding factor again and continue in this way. Or one can recall one useful identity and proceed as follows:

$$\lambda^4 - 1 = (\lambda^2 - 1)(\lambda^2 + 1) = (\lambda - 1)(\lambda + 1)(\lambda^2 + 1).$$

We thus have

$$p(\lambda) = \lambda^3(\lambda - 1)^2(\lambda + 1)(\lambda^2 + 1).$$

We can see the following roots:

$$\lambda = 0 \text{ (3}\times\text{)}, 1 \text{ (2}\times\text{)}, -1, \pm i.$$

The corresponding contributions to the basis are:

$$\begin{aligned} 0 : e^{0 \cdot x}, x e^{0 \cdot x}, x^2 e^{0 \cdot x} &\implies 1, x, x^2; \\ 1 : e^{1 \cdot x}, x e^{1 \cdot x} &\implies e^x, x e^x; \\ -1 : e^{-1 \cdot x} &= e^{-x}; \\ 0 \pm 1i : e^{0 \cdot x} \cos(1 \cdot x), e^{0 \cdot x} \sin(1 \cdot x) &\implies \cos(x), \sin(x). \end{aligned}$$

We thus obtain the fundamental system

$$\{1, x, x^2, e^x, x e^x, e^{-x}, \cos(x), \sin(x)\}.$$

An experienced solver would not need to write down this basis, passing directly from characteristic numbers to a general solution:

$$y(x) = a + bx + cx^2 + d e^x + f x e^x + g e^{-x} + h \cos(x) + k \sin(x), \quad x \in \mathbb{R}.$$

We preferred not to use  $e, i, j$  as coefficients to avoid confusion.

If we wanted just one solution, then it would be typically selected by specifying initial conditions at some  $x_0$ . In this case we would need eight of them, so we would need to specify values for  $y(x_0)$ ,  $y'(x_0)$ ,  $\dots, y^{(7)}(x_0)$ .

To find this solution, we would apply the general solution above to these conditions, and obtain an  $8 \times 8$  system of linear equations. I hope I will be forgiven for not showing an example.

△

Now we know how to solve homogeneous linear differential equations.

**15.15 Remark:** In order to get general results we had to restrict our attention to the case of constant coefficients. However, there is special case when we can handle also homogeneous linear equations with general coefficients. We have already discussed it in chapter 9. A general first order homogeneous differential equation has the form  $y' + a(x)y = 0$ , and as such it is a separable equation:

$$y' = -a(x)y \implies \int \frac{dy}{y} = \int -a(x) dx \implies \ln |y| = -A(x) + C,$$

where  $A$  is some antiderivative of  $a$ . Then using the traditional approach we obtain the general solution  $y(x) = D e^{-A(x)}$ .

△

## 16. Nonhomogeneous linear differential equations

Given a nonhomogeneous (or inhomogeneous) linear differential equation, we can always pretend that there is a zero on the right, obtaining more friendly equation. In fact, this is a popular move when it comes to linear equations.

### Definition 16.1.

Given a linear ODE  $y^{(n)} + a_{n-1}(x)y^{(n-1)} + \cdots + a_1(x)y' + a_0(x)y = b(x)$ , by its **associated homogeneous equation** we mean the equation  $y^{(n)} + a_{n-1}(x)y^{(n-1)} + \cdots + a_1(x)y' + a_0(x)y = 0$ .

Of course, we are then solving a different equation, not the one that we were given. How does it help? The following theorem should be familiar to everyone who ever looked at linear equations of any kind.

### Theorem 16.2. (on structure of solution set of linear ODE)

Let  $y_p$  be some particular solution of a given linear ODE on an open interval  $I$ . A function  $y_0$  is a solution of this equation on  $I$  if and only if  $y_0 = y_p + y_h$  for some solution  $y_h$  of the associated homogeneous equation on  $I$ .

As an equivalence, this statement actually includes two claims. Both start with some particular solution  $y_p$  of the given problem.

One statement claims that whenever we add to  $y_p$  some solution  $y_h$  of the associated homogeneous equation (we will just say “homogeneous solution” from now on), we get a solution of the original equation. So this is actually a procedure for creating new solutions out of the one that we know.

The second claim is equally important. The theorem says that the above procedure provides us with all solutions and we do not need to try anything else. More precisely, every solution  $y_0$  of the original problem must be reachable as  $y_p + y_h$  for some homogeneous solution  $y_h$ .

These two observations form the plan for our proof. To simplify calculations, we will work with the linear mapping (operator)  $L$  defined by the left-hand side of the equation, see remark 15.4.

**Proof:** Assume that  $y_p$  is a solution of the given equation, we call it (L), on  $I$ . Let us denote the action of the left-hand side of the equation as  $L$ , so the equation (L) reads  $L[y] = b(x)$  and the associated homogeneous equation reads  $L[y] = 0$ .

1) Let  $y_h$  be some homogeneous solution on  $I$ . This means that  $L[y_h] = 0$  on  $I$ , that is,  $L[y_h](x) = 0$  for all  $x \in I$ . We claim that  $y = y_p + y_h$  is a solution of (L) on  $I$ . We substitute  $y$  into  $L$ , and by linearity we get that for any  $x \in I$ ,

$$L[y_p + y_h](x) = L[y_p](x) + L[y_h](x) = b(x) + 0 = b(x).$$

It may be a good idea for the reader to try a hands-on proof by substituting  $y = p + y_h$  into the left-hand side of the equation written in full like we did in the proof of theorem 15.3, then multiply out etc.

2) Let  $y_0$  be any solution of (L) on  $I$ . We need to find a homogeneous solution  $y_h$  such that  $y_0 = y_p + y_h$ .

We get our inspiration from the formula that we want to be true and decide to work with the function  $y_h = y_0 - y_p$ . It is a function on  $I$ , and obviously satisfies one of the two requirements:

$$y_p + y_h = y_p + (y_0 - y_p) = y_0.$$

We need to show that it also solves the associated homogeneous equation. We substitute this function into  $L$  and see:

$$L[y_h](x) = L[y_0 - y_p](x) = L[y_0](x) - L[y_p](x) = b(x) - b(x) = 0 \text{ on } I.$$

The claim is proved. □

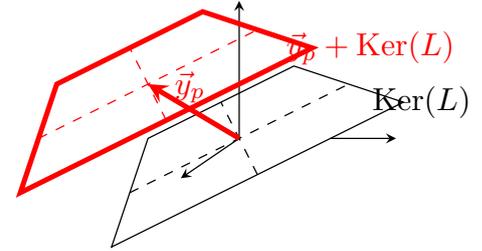
This theorem can be stated also in other ways. A more abstract point of view would tell us about the set of all solutions to the given linear ODE:

$$\begin{aligned} & \{y; y \text{ a solution of given ODE on } I\} \\ &= \{y_p + y_h; y_h \text{ a solution of the associated homogeneous ODE on } I\} \\ &= y_p + \{y_h; y_h \text{ a solution of the associated homogeneous ODE on } I\} \end{aligned}$$

In the last equality we used the convention that having a set of functions  $M$ , we can use the notation  $f + M$  to represent the set obtained by adding  $f$  to all functions in  $M$ . We can see it as a shift of the set  $M$ . Recalling the discussion about an associated mapping  $L$  from the previous chapter, we can write

$$\{y; y \text{ a solution of given ODE on } I\} = y_p + \text{Ker}(L).$$

The reader probably knows an analogous theorem from linear algebra about systems of linear equations. There we have a nice geometric interpretation. Solutions are actual geometric vectors (points), and the subspace  $\text{Ker}(L)$  of all homogeneous solutions is a line or plane or something going through the origin. When we want to solve a nonhomogeneous equation, we find one particular solution and use it to shift this set in space.



**16.3 Remark:** In section we introduced an interesting interpretation. In a differential equation  $L[y] = b(x)$ , the associated homogeneous equation  $L[y] = 0$  can be interpreted as describing the behavior of the studied system if left alone, while the full equation  $L[y] = b(x)$  describes its response when we interfere with it by some outer influence  $b(x)$ . It is a nice story and we cannot do much more about it in general.

However, for a linear functions we can say more. The theorem above tells us that the response of a linear system consists of two components. The part  $y_h$  shows how the system would behave if we left it alone, and this is still preserved as a factor in the solution; we learn how a modified system behaves by modifying the inherent behavior of the system with one particular response  $y_p$  to our input.

△

From a more abstract point of view we turn to a very practical one. One possible way to express the statement of the above theorem is as follows.

**Fact 16.4.**

Consider some linear differential equation. If  $y_h$  is a general solution of the associated homogeneous equation on  $I$ , then  $y_p + y_h$  is a general solution of the given equation on  $I$ .

This is a blueprint for finding general solutions that we will follow.

This shows that the purpose of the structural theorem is to pass the buck. Our aim is to obtain a complete information about solutions of the given equation. However, the this nonhomogeneous equation is too tough to handle, so instead we do the complete analysis for the associated homogeneous equation, and we can use our results from chapter 15 for that. True, we will also need some solution of the original equation, but just one is enough and we can always hope to get it somehow. The next two sections will show some approaches.

## 16a. Method of undetermined coefficients

Where do we find some solution of a given nonhomogeneous linear ODE? The most popular method uses educated guessing.

**Example 16a.a:** Consider the equation  $y'' - 3y' + 2y = 10e^{3x}$ .

The corresponding linear operator is  $L[y] = y'' - 3y' + 2y$  and we can see the given equation as a problem of hitting the target  $10e^{3x}$  using the mapping  $L$ . How does it behave? For instance like this.

$$\begin{aligned} L[e^{4x}] &= [e^{4x}]'' - 3[e^{4x}]' + 2 \cdot e^{4x} = 16e^{4x} - 3 \cdot 4e^{4x} + 2e^{4x} = 6e^{4x}; \\ L[e^{-x}] &= [e^{-x}]'' - 3[e^{-x}]' + 2 \cdot e^{-x} = e^{-x} - 3 \cdot (-e^{-x}) + 2e^{-x} = 6e^{-x}. \end{aligned}$$

It seems that when we feed  $L$  some exponential, then it will also appear on the output, perhaps with a multiplicative constant in front of it. We confirm it easily:

$$L[e^{\alpha x}] = (\alpha^2 - 3\alpha + 2)e^{\alpha x}.$$

Note that it will be true for all linear equations with constant coefficients. The corresponding operator  $L$  is then a linear combination of derivatives. Exponentials are untouched by derivatives, and a linear combination simply sums up those exponentials multiplied by numbers, resulting in some multiple of an exponential. So this is a very general observation and it can be used to guess how a solution looks like when we see an exponential on the right.

Returning back to our example, we want to reach the right-hand side  $b(x) = 10e^{3x}$ . By our above observations, when we want to see  $e^{3x}$  in the output, it makes sense to feed  $L$  with  $e^{3x}$  as the input.

$$L[e^{3x}] = [e^{3x}]'' - 3[e^{3x}]' + 2 \cdot e^{3x} = 9e^{3x} - 3 \cdot 3e^{3x} + 2e^{3x} = 2e^{3x}.$$

Almost, but not quite. However, the operator  $L$  is linear, so if we multiply the input by some constant, then the same multiplication appears in the output. This inspires us to try  $y(x) = 5e^{3x}$  and indeed,  $L[5e^{3x}] = 10e^{3x}$ . We just found a particular solution of the given equation.

This is the heart of the main method that we will introduce here. I like the name **guessing method**, as it is short and captures the spirit. However, it will not be just any guessing, but an educated and highly sophisticated guessing, so the official name is different.

We understand that our chances to hit  $10e^{3x}$  with the function  $e^{3x}$  are rather slim, because constants are bound to appear, so we expect that the actual solution will be some multiple of  $e^{3x}$ . Here is an idea: We will feed  $L$  with  $Ae^{3x}$ , see what comes out and use  $A$  to adjust our guess.

$$L[Ae^{3x}] = [Ae^{3x}]'' - 3[Ae^{3x}]' + 2 \cdot Ae^{3x} = 2Ae^{3x}.$$

We want to obtain  $10e^{3x}$ , which means that we need  $2A = 10$  to be true. We see that  $A = 5$  does the trick, and in this way we obtain the particular solution  $y_p(x) = 5e^{3x}$ .

What did we learn? When we see an exponential  $e^{\alpha x}$  on the right in our equation, we make a guess  $Ae^{\alpha x}$ . We put it into  $L$ , and compare the output with the desired right-hand side, obtaining the right value for  $A$  that will make our guess into a particular solution.

This is the preferred approach. We had to determine  $A$  to get our solution, so the official name for this method is the **method of undetermined coefficients**.

We are not ready yet to make some statement about it, we need to explore some more.

△

Note that there is a crucial restriction on the coefficients. Consider the differential equation  $y' + \ln(x)y = 0$ . It is linear, but when we consider the corresponding mapping  $L[y] = y' + \ln(x)y$ , then it no longer transforms exponentials into exponentials:

$$L[e^{13x}] = 13e^{13x} + \ln(x)e^{13x} = (13 + \ln(x))e^{13x}.$$

This observation and restriction on coefficients applies also to all the successive experimentd and deductions.

**Example 16a.b:** Consider the equation  $y'' - 3y' + 2y = 4x$ .

We see a polynomial on the right, and inspired by the previous example we ask what a linear

differential operator does to polynomials.

$$\begin{aligned} L[x^3 - x] &= [x^3 - x]'' - 3[x^3 - x]' + 2 \cdot (x^3 - x) = 6x - 3(3x^2 - 1) + 2 \cdot (x^3 - x) \\ &= 2x^3 - 9x^2 + 4x + 3. \end{aligned}$$

What do we think of this situation? A polynomial, when subjected to differentiation and linear combinations, turns into a polynomial again. It can never change into a polynomial of higher degree (we need constant coefficients in the equation and the mapping for this), and if there is the term  $y$  in the equation (which is usually true), then the degree is actually preserved.

It therefore makes sense to try a polynomial when we see a polynomial on the right. Since the degree cannot increase while passing through  $L$ , we have to use a polynomial of at least the same degree as we see on the right as our guess. And since we do not like working more than it is necessary, it makes sense to try a polynomial of the same degree.

We now apply this observation to our example with the right-hand side  $b(x) = 4x$ . For starters, we try  $y(x) = x$ :

$$L[x] = [x]'' - 3[x]' + 2 \cdot x = 2x - 3.$$

Focusing on the crucial linear term for now, we see that we should start with  $2x$  to get the constant right. Indeed,

$$L[2x] = [2x]'' - 3[2x]' + 2 \cdot 2x = 4x - 6.$$

There is still the absolute term to fix. We could try it intuitively, but let's try the efficient way. We feel that to obtain  $4x$ , we would need to start with some  $Ax$  on the input. But as we send it through  $L$ , the derivatives will turn this also into some constant terms. To get rid of them, we have to be able to adjust our guess, so we add a constant term. We therefore decide on the guess  $y(x) = Ax + B$  (we traditionally use upper-case letters for the undetermined constants). How does it work?

$$L[Ax + B] = [Ax + B]'' - 3[Ax + B]' + 2 \cdot (Ax + B) = 2Ax + (-3A + 2B).$$

We see the output, and we want this to be  $4x + 0$ . We readily deduce conditions  $2A = 4$  and  $-3A + 2B = 0$ , and we find that the right multiplicative constants are  $A = 2$  and  $B = 3$ . Indeed,

$$L[2x + 3] = [2x + 3]'' - 3[2x + 3]' + 2 \cdot (2x + 3) = 4x.$$

We found our solution.

What did we learn? When we see a polynomial on the right, it makes sense to take a general polynomial of the same degree as our guess. And we do have to take a full polynomial. We cannot skip any lower power, because a differential operator freely changes powers into lower ones, so we can never tell which ones we will need to adjust the output.

Note that we could also use the guess  $y(x) = Ax - B$ . Then

$$L[Ax - B] = 2Ax + (-3A - 2B)$$

and we obtain  $A = 2$ ,  $B = -3$ . We see that this new  $B$  adjusted to our choice of the sign, so we end up with the solution  $2x + 3$  anyway. Sometimes people prefer to put minus in their guesses (don't ask me why), and as we can see, it does not really make any difference, because the constants simply adjust.

△

Note how this fits nicely with the idea of the previous example. We see two types of expressions that are preserved by linear differential operators: exponentials and polynomials. In both cases multiplicative constants typically change when passing through  $L$ , with polynomials we see them as coefficients. We therefore base our guess on the shape of the expression on the right, we just replace concrete constants with unknown parameters.

It is actually possible to connect these two cases. Notice the following:

$$\begin{aligned} L[x e^{3x}] &= [x e^{3x}]'' - 3[x e^{3x}]' + 2 \cdot x e^{3x} = (6e^{3x} + 9x e^{3x}) - 3 \cdot (e^{3x} + 3x e^{3x}) + 2x e^{3x} \\ &= (2x + 3)e^{3x}. \end{aligned}$$

We see that the exponential survived, while the polynomial changed into another one. This can be confirmed in general, which brings another type that we can handle: When we see an exponential multiplied by a polynomial on the right in our equation, we create our guess by including the exponential, and changing the polynomial into a general one of the same degree.

Note that the first two examples can now be seen as just special cases of this general rule. An exponential  $e^{\alpha x}$  can be seen as  $1 \cdot e^{\alpha x}$ , where 1 is interpreted as a polynomial of degree zero. We know that passing  $e^{\alpha x}$  through some linear differential mapping  $L$  will, in general, result in some  $a \cdot e^{\alpha x}$ , and we can interpret this as a change in coefficients of the polynomial 1, which is exactly what we know happens with polynomials when passing through  $L$ .

Similarly, a polynomial  $p(x)$  can be interpreted as  $p(x) \cdot 1 = p(x) \cdot e^{0 \cdot x}$ . The general expectation is that a linear differential mapping preserves exponentials and changes coefficients of polynomials, so we expect to obtain some  $q(x) \cdot e^{0 \cdot x} = q(x)$ , that is, a polynomial. We could say that exponentials are preserved by  $L$ , and also the non-existence of exponentials is preserved by  $L$ .

Based on this we make a general guessing rule: If the right-hand side is a polynomial times an exponential, then preserve the exponential (if none, then preserve its non-presence) and generalize the polynomial.

This works very well, and it is natural to ask what other types of functions get preserved by linear differential operators. Unfortunately, there are none. However, this is not the end, we can actually apply the guessing procedure in one more case, but we need to revisit the polynomial example first.

Note that a polynomial is in fact just a linear combination of powers. So besides talking about one function of a certain type that gets preserved, we can also talk about a group of functions that gets preserved. An individual power  $x^k$  will most likely change into something else when acted upon by a linear differential mapping, but a linear combination of powers  $x^0 = 1, x, x^2, \dots, x^m$ , when substituted into such an operator  $L$ , produces another linear combination of the same group. Then we can use smart guessing to set up a general shape or form of a particular solution and work out proper values of parameters.

We thus arrive at a natural question: Is there another interesting group of functions that would be preserved by  $L$ ?

**Example 16a.c:** Consider the equation  $y'' - 3y' + 2y = 85 \sin(4x)$ .

What happens when we feed  $L$  with a sine of appropriate frequency  $\beta = 4$ ?

$$\begin{aligned} L[\sin(4x)] &= [\sin(4x)]'' - 3[\sin(4x)]' + 2 \cdot \sin(4x) = -16 \sin(4x) - 3 \cdot 3 \cos(4x) + 2 \sin(4x) \\ &= -14 \sin(4x) - 9 \cos(4x). \end{aligned}$$

We know that derivatives can turn a sine into a cosine (and vice versa), so we are not surprised to see more than just a multiple of  $\sin(4x)$  in the output. If we were interested just in functions that get preserved, then this would be a dead end. However, now we know that it makes sense to look at groups, and we readily recognize that any linear combination of a sine and cosine with the same frequency,

$$a \cos(\beta x) + b \sin(\beta x)$$

can change, through derivatives and linear combinations, only into another linear combination of the cosine and sine with the frequency  $\beta$ . We can therefore think of a cosine and a sine as a pair that always comes together, even if one of them may be invisible. For instance, the given equation can be seen as

$$y'' - 3y' + 2y = 85 \sin(4x) + 0 \cdot \cos(4x).$$

By our observation, we think that we should feed the mapping  $L$  some combination of  $\sin(4x)$  and  $\cos(4x)$  to get the right output. We do not know what combination, so we again form a general guess with undetermined coefficients:  $A \sin(4x) + B \cos(4x)$ . After all, we know that if we feed  $L$  just with the sine, there will most likely be some cosines with the same argument in the output, so we have to give ourselves some means to adjust. We easily find that

$$L[A \sin(4x) + B \cos(4x)] = (-14A + 12B) \sin(4x) + (-12A - 14B) \cos(4x).$$

Comparing this to the given right-hand side we see that we need to make sure that  $-14A + 12B = 85$  and  $-12A - 14B = 0$ . This yields  $A = -\frac{7}{2}$  and  $B = 3$ , and if you do not trust us, you can check for yourself that

$$y_p(x) = 3 \cos(4x) - \frac{7}{2} \sin(4x)$$

is a particular solution of the equation  $y'' - 3y' + 2y = 85 \sin(4x)$  on  $\mathbb{R}$ .

△

Now we have three basic types of expressions that we can handle. Exponentials, polynomials, and pairs involving a sine and a cosine with the same frequency. Unfortunately, there are no more distinct types like that, which limits the scope of our approach. The good news is that great many applied problems fit this pattern.

We also observed that we can actually combine a polynomial with an exponential (multiply them) and the resulting expression is also well-behaved, in fact one can form a general strategy for this expression and treat polynomials and exponentials as special cases.

We can similarly include sines and cosines in this way. If we take a linear combination of  $p(x)e^{\alpha x} \cos(\beta x)$  and  $q(x)e^{\alpha x} \sin(\beta x)$ , where  $p(x), q(x)$  are polynomials, then applying a linear differential mapping we obtain an expression of exactly the same type, with the exponential and cosine/sine preserved (if they are there in the first place), while the polynomials change coefficients. In fact, a linear combination of such terms can be always written as

$$p(x)e^{\alpha x} \cos(\beta x) + q(x)e^{\alpha x} \sin(\beta x),$$

so this is the basic form for us. When we understand how it behaves, we will also know how to handle special cases when some parts of this expression are not present.

However, the general case gets a bit more complicated. We noted before that when we feed  $L$  with just a polynomial, or a polynomial multiplied by an exponential, then the degree is preserved or decreases. However, this is no longer true once we allow sines and cosines into the game. Indeed, consider the following example:

$$[x \cos(x) + \sin(x)]' = 2 \cos(x) - x \sin(x).$$

We see that originally, the polynomial in front of cosine had degree one, while the (virtual) polynomial 1 that multiplies sine had degree zero. After we let it through a very simple differential mapping  $L[y] = y'$  the degrees of polynomials switched. This is caused by the fact that derivatives change cosines into sines or vice versa freely, so any adjacent polynomials can travel from one to another. The best we can say is the following fact: After we send some expression of the type

$$p(x)e^{\alpha x} \cos(\beta x) + q(x)e^{\alpha x} \sin(\beta x)$$

through some linear differential mapping  $L$  (that is, when we substitute it into a left-hand side of some linear equation with constant coefficients), then the new expression will have polynomials in front of  $e^{\alpha x} \cos(\beta x)$  and  $e^{\alpha x} \sin(\beta x)$ , and the degrees of these polynomials cannot be higher than the degrees present in the original linear combination. This observation allows us to make a good guess when looking for a particular solution.

Basic setup: We have a linear differential equation with constant coefficients, where the right-hand side fits the pattern

$$p(x)e^{\alpha x} \cos(\beta x) + q(x)e^{\alpha x} \sin(\beta x),$$

note that the sine and cosine have the same frequency  $\beta$ . Such expressions are sometimes called “quasipolynomials”, here we will also say that the right-hand side is “special”. Very often some parts are missing. In particular, when the trig functions are not present, then we deal with just one expression of the type  $p(x)e^{\alpha x}$ .

We create the basic form or the basic shape of a particular solution according to the following rules:

- If the expression involves a sine or a cosine, we always have to work with a complete pair

$$p(x)e^{\alpha x} \cos(\beta x) + q(x)e^{\alpha x} \sin(\beta x),$$

even if one of them may be absent in the given equation.

- For every group in the right-hand side we create a group in our guess, preserving the exponential and the cosine or sine, if present. The polynomials are generalized into full polynomials with unknown coefficients. There is always a polynomial there, even if it is just an unwritten constant polynomial 1 multiplying the given expression.

- If the group does not involve a cosine nor sine, then it is just one group and our situation is easier: We preserve the exponential (if present) and generalize the polynomial to one of equal degree.

- When we do have some trig function and deal with a pair of terms, then we determine the largest degree of the two polynomials and use this it when forming the general polynomials in our guess.

For instance, if we see  $13 \sin(3x) + x \cos(3x)$  on the right in our equation, then we have to put

$$(Ax + B) \cos(3x) + (Cx + D) \sin(3x)$$

in our guess. Note that we preserved exponentials (there were none).

After we write the proper form of a solution, we substitute it into the left-hand side  $L$ , work out the output, and compare it with the given right-hand side. If we guessed the right form for our particular solution, then the number of equations that we obtain should match the number of undetermined coefficients.

This approach will work most of the time, but we will not make any formal statement yet, first we need to look into that “most of the time”.

**Example 16a.d:** Consider the equation  $y'' - 3y' + 2y = 4e^{2x}$ . We see an exponential multiplied by a polynomial on the right, so we know how to create the right form of a solution: We copy the exponential and generalize the polynomial. We therefore think that there should be a particular solution of the form  $y(x) = Ae^{2x}$ . We substitute into  $L$ :

$$L[Ae^{2x}] = [Ae^{2x}]'' - 3[Ae^{2x}]' + 2 \cdot Ae^{2x} = 4Ae^{2x} - 3 \cdot 2Ae^{2x} + 2Ae^{2x} = 0.$$

This is bad, we were aiming for the target  $4e^{2x}$ . Where is the problem?

The associated homogeneous equation  $y'' - 3y' + 2y = 0$  has characteristic numbers  $\lambda = 1, 2$ , in particular we see that  $e^{2x}$  is actually a solution of the homogeneous equation. This explains why it was swallowed up by  $L$ . Now it is fairly obvious that the above guessing procedure fails if on the right we have an expression that also solves the associated homogeneous equation. We have to learn how to handle this.

The general approach is that we have to protect our guess, which is done by multiplying it by  $x$ . Indeed, if we try the guess  $y(x) = xAe^{2x}$ , we obtain

$$\begin{aligned} L[xAe^{2x}] &= [xAe^{2x}]'' - 3[xAe^{2x}]' + 2 \cdot xAe^{2x} \\ &= (4Ae^{2x} + 4Ax e^{2x}) - 3(Ae^{2x} + 2Ax e^{2x}) + 2Ax e^{2x} = Ae^{2x}. \end{aligned}$$

We see that  $L$  stripped away the extra term  $x$  and the exponential itself survived. Comparing to our target we see that  $A = 4$  works, so  $y_p(x) = 4x e^{2x}$  is a particular solution.

Sofar we were only looking for particular solutions, let's finish this problem properly. We know that a general solution can be obtained as  $y_p + y_h$ , where  $y_h$  is some general homogeneous solution.

We already found the characteristic numbers, so we can easily construct a general homogeneous solution and conclude that

$$y(x) = 4x e^{2x} + a e^x + b e^{2x}, \quad x \in \mathbb{R}$$

is a general solution of the given equation.

△

This example identified a serious problem: The expression on the right in our equation may be related to solutions of the associated homogeneous equation. We saw the basic remedy, but it actually deserves a closer look. Imagine that  $\lambda = 2$  actually were a characteristic number of multiplicity 2. Then also  $x e^{2x}$  would be a homogeneous solution, so  $L$  would turn it into zero. In that case we would use  $x^2$  for protection. And if  $\lambda = 2$  had multiplicity three, we would have to use  $x^3$  and so on. That is the general pattern.

A good question is how do we actually recognize such troublesome cases. An important point is that the polynomials in our special right-hand sides do not play any role here. The basic behaviour of a special expression is determined solely by presence of exponentials and cosines/sines in it. So we look at the expression on the right, strip any polynomial that is present, and check whether the resulting expression solves the associated homogeneous equation.

Typically, before we start looking for a particular solution  $y_p$ , we already solved the associated homogeneous equation. In the example above we would have

$$y_h(x) = a e^x + b e^{2x}.$$

The fundamental system is  $\{e^x, e^{2x}\}$ . Next we turn to finding  $y_p$ . We look at the right-hand side  $b(x)$ , disregard the polynomial 4 and ask whether the resulting expression  $e^{2x}$  matches one from the fundamental system. In this case the answer is in the positive, so we know that we will have to preserve our basic guess.

This intuitive approach is definitely possible, but it can get tricky, so most people prefer a different approach. It uses the relationship between numbers  $\lambda$  and expressions. We know this relationship well, because we use it to find general solutions of homogeneous equations. However, this relationship can be also used in the reverse direction, and assign numbers to expressions.

When processing some expression on the right-hand side, we ignore the polynomial in it and encode the presence of an exponential and trig functions using a special number  $\lambda$  in exactly the same way as in the previous chapter: The real part of  $\lambda = \alpha + \beta i$  encodes the presence of  $e^{\alpha x}$  in the expression (and we put  $\alpha = 0$  if there is no exponential there), while  $\beta$  is the frequency of the sine or cosine if present, again we put zero if there is none. This resulting special number can then be directly compared to characteristic numbers, and if we find a match, then we know that we have to protect.

Returning to the above example, when we see the right-hand side  $4e^{2x}$ , we could say that its special number is  $\lambda = 2$ . We see that there is also the same characteristic number, so we have to apply a correction to our basic guess  $A e^{2x}$ . Since the characteristic number  $\lambda = 2$  has multiplicity one, just one correction is enough, hence we use the guess  $x A e^{2x}$ . It all fits.

The construction of a general form of a particular solution therefore has two stages:

First, we create the basic shape as outlined above, by generalizing polynomials;

Second, we determine the special number of the right-hand side, compare it with characteristic numbers, and if there is a match (an overlap between the right-hand side and left-hand side as captured by those lambdas), then we protect our guess by a corrective factor  $x^m$ , with the power determined by the multiplicity of that match.

Obviously, it is important to have the relationship between  $\lambda$  and the expression mastered. Theoretically it is enough to know the general case

$$\alpha + \beta i \iff e^{\alpha x} \cos(\beta x), e^{\alpha x} \sin(\beta x)$$

However, it helps in practical calculations to be aware of special cases so that one can do them quickly:

$$\begin{aligned}\lambda = \alpha &\iff e^{\alpha x} \\ \lambda = \beta i &\iff \cos(\beta x), \sin(\beta x) \\ \lambda = 0 &\iff p(x)\end{aligned}$$

Especially the last one is sometimes tricky for beginners. My rule of thumb: I get  $\alpha$  from the exponential on the right, and if there is none, I put zero. Similarly,  $\beta$  indicates the presence of sines/cosines.

Finally, if you want, you can also use  $\alpha - \beta i$  for the special number, it makes no difference. We already commented that for a real polynomial, the numbers  $\alpha \pm \beta i$  come in pairs, and if they are roots (characteristic numbers), then they have to have the same multiplicity, therefore we get the same information about overlaps and degree of correction for both.

**16a.1 Remark:** Above we recalled the interpretation that  $L[y] = 0$  describes the behavior of some system itself and  $b(x)$  is an outer influence. Now we push this idea further. When we study some system and we are able to describe its inner workings by a linear differential equation with constant coefficients, then the characteristic numbers  $\lambda$  describe modes of behavior that are special for this system, one could say that these types of behavior are natural for it or that the system likes them.

When we influence this system from outside, as represented by the right-hand side  $b(x)$ , and this influence is special (a quasipolynomial), then the special number  $\lambda$  captures the substance of this outside influence. If this outer influence differs from those special modes that the system likes, then the response of the system to our input is proportional to what we do (the particular solution has the same form as the right-hand side) and it is added to the basic behavior, which we see in the formula  $y_p + y_h$ .

However, if the system likes acting in a certain way and then on the top of that we start influencing it in exactly the same way (the same lambdas), then the system reacts very happily and we observe a much more pronounced reaction of the system, which is mathematically reflected by us multiplying the basic form of particular solution by some power of  $x$ . One could say that the extra power of  $x$  is caused by a resonance between inner and outer influences.

Imagine a pendulum. If left alone, it simply swings (see section 17 for a mathematical treatment). If we act on this pendulum by a constant force (we blow at it from one side), then the swinging motion just shifts to the side and the pendulum keeps going. We add a constant displacement to the natural position of the pendulum, the formula  $y_p + y_h$  can be nicely interpreted here.

However, if we start alternately blowing and sucking, and we do it in such a way that we blow when the pendulum moves away from us and suck when it moves towards us, then the frequency of our influence matches the natural frequency of the swing (our lambda and the lambda of the pendulum agree), we get resonance and the outcome is obvious: The swings get wider and wider.

This is a serious consideration for people who design various structures. Every mechanical structure has some frequencies that it likes. If it happens that an outer force has the same frequency (think of a tall building swinging in a changing wind), then things can get very bad. There is a famous example of a suspended bridge in England and a unit of soldiers marching across it with exactly the right frequency, so the solution (the swings of the bridge) got multiplied by  $x$ . Fortunately no soldier died, but the bridge was not so lucky (Broughton bridge, 1831). Ever since, armies of the world order their soldiers to break step when crossing bridges, and engineers worry about natural frequencies of structures.

However, this is not an engineering book, so we will just be careful and protect our guesses when the special numbers characterizing the left-hand side and the right-hand side match.

△

Now we are ready to confirm that our guessing procedure is guaranteed to work, and we should be able to decipher the meaning of various parts of the following theorem. We will follow the custom and rewrite the expression

$$p(x)e^{\alpha x} \cos(\beta x) + q(x)e^{\alpha x} \sin(\beta x)$$

as one term.

**Theorem 16a.2.** (guessing a solution for special right hand-side)

Consider a linear ODE with constant coefficients

$$y^{(n)} + a_{n-1}y^{(n-1)} + \dots + a_1y' + a_0y = b(x).$$

Assume that  $b(x) = e^{\alpha x}[p(x) \cos(\beta x) + q(x) \sin(\beta x)]$  for some polynomials  $p, q$ , denote  $d = \max(\deg(p), \deg(q))$ .

We consider the special number  $\lambda = \alpha \pm \beta i$  and let  $m$  be the multiplicity of it as a characteristic number of the given equation (we put  $m = 0$  if it is not a characteristic number at all).

Then there are polynomials  $\tilde{p}, \tilde{q}$  of degree at most  $d$  such that

$$y(x) = x^m e^{\alpha x} [\tilde{p}(x) \cos(\beta x) + \tilde{q}(x) \sin(\beta x)]$$

is a solution of the given equation on  $\mathbb{R}$ .

This theorem guarantees that when we set up the right form of our guess with general polynomials in it, then there must be some coefficients that will turn our guess into a solution. One can set up the guess according to the theorem, or according to our observations that preceded them, the end result should be the same. I prefer to have the terms  $p(x)e^{\alpha x} \cos(\beta x)$  and  $q(x)e^{\alpha x} \cos(\beta x)$  separate, it makes it easier (I think) to deal with simpler cases where some parts are missing. In the following example we will show both approaches.

**Example 16a.e:** Consider the equation  $y' + 2y = e^{-x}[5x \cos(2x) + 9 \sin(2x)]$ . It is linear, and the right-hand side fits with our theorem. We will find its general solution.

a) We start with a homogeneous case. The equation  $y' + 2y = 0$  leads to the characteristic equation  $\lambda + 2 = 0$  with solution  $\lambda = -2$ . We have a general homogeneous solution

$$y_h(x) = a e^{2x}.$$

b) Now we find some particular solution using the method of undetermined coefficients as captured in the theorem.

Comparing  $b(x) = e^{-x}(5x \cos(2x) + 9 \sin(2x))$  with the general pattern we see that the parameters are  $\alpha = -1$ ,  $\beta = 2$ ,  $p(x) = 5x$ , and  $q(x) = 9$ , so the maximal degree is  $d = 1$ . The special number  $\lambda = -1 + 2i$  does not match any of the characteristic numbers (it is not  $-2$ ), hence  $m = 0$ . By the theorem, there must be a solution of the following form:

$$y(x) = x^0 e^{-x} (\tilde{p}(x) \cos(2x) + \tilde{q}(x) \sin(2x)),$$

where  $\tilde{p}, \tilde{q}$  are certain polynomials of degree  $d = 1$ . In other words, there must be constants  $A, B, C, D$  for which the function

$$y(x) = e^{-x} ((Ax + B) \cos(2x) + (Cx + D) \sin(2x)).$$

solves the given equation. To find them, we simply substitute our guess  $y$  into the equation and see. As usual, it pays to see first what the left-hand side  $L$  does to this function. It is a good idea to figure out the derivative first,

$$y' = e^{-x} [(2C - A)x + (A - B + 2D)] \cos(2x) + [(-2A - C)x + (-B - 2C + D)] \sin(2x),$$

and now we substitute:

$$\begin{aligned} L[y] &= e^{-x}[(2C - A)x + (A - B + 2D)] \cos(2x) + ((-2A - C)x + (-2B + C - D)) \sin(2x) \\ &\quad + 2e^{-x}((Ax + B) \cos(2x) + (Cx + D) \sin(2x)) \\ &= e^{-x}([A + 2C]x + [A + B + 2D]) \cos(2x) + ([C - 2A]x + [-2B + C + D]) \sin(2x). \end{aligned}$$

Note that we rearranged the output so that it fits with the pattern from the theorem, in particular we wrote the polynomials in front of the cosine and sine in a way that allows us to see the coefficients in them.

We want this output to be  $e^{-x}[(5x + 0) \cos(2x) + (0x + 9) \sin(2x)]$ . By comparing we obtain four equations:

$$\begin{aligned} A + 2C &= 5 \\ A + B + 2D &= 0 \\ -2A + C &= 0 \\ -2B + C + D &= 9 \end{aligned}$$

We can solve this system in any way we prefer. For instance, one can notice that the first and third equation form an independent subsystem, we obtain  $A = 1$  and  $C = 2$ . We put this into the second and fourth equation and obtain another system of two equations with two unknowns  $B + 2D = -1$ ,  $-2B + D = 7$ , which yields  $B = -3$ ,  $D = 1$ .

We have the particular solution  $y_p(x) = e^{-x}((x - 3) \cos(2x) + (2x + 1) \sin(2x))$ .

If we wanted to apply the approach developed from experiments, we would see the right-hand side as  $b(x) = 5xe^{-x} \cos(2x) + 9e^{-x} \sin(2x)$ . Disregarding the polynomials, we would notice that both terms have the same special number  $\lambda = -1 + 2i$  and thus conclude that they belong together. We may also remark for later that this  $\lambda$  does not match any characteristic number, so there will be no need for correction.

We now form the basis pattern for the solution: We preserve exponentials and trig functions and generalize polynomials, using the highest degree present, that is, degree 1:

$$y(x) = (Ax + B)e^{-x} \cos(2x) + (Cx + D)e^{-x} \sin(2x).$$

We already noticed that no correction is necessary, so this is our guess for the solution.

Comparing with the previous approach, we note that this guess is the same, but in this form is easier to digest, in particular because there is an extra level of bracketing in the first approach.

Since the guess is the same, the steps that follow (substitution etc.) would be analogous, just the expressions would be a bit more friendly again:

$$\begin{aligned} L[y] &= ([A + 2C]x + [A + B + 2D])e^{-x} \cos(2x) + ([C - 2A]x + [-2B + C + D])e^{-x} \sin(2x) \\ &\stackrel{?}{=} (5x + 0)e^{-x} \cos(2x) + (0x + 9)e^{-x} \sin(2x). \end{aligned}$$

Eventually we arrive at the particular solution  $y_p(x) = (x - 3)e^{-x} \cos(2x) + (2x + 1)e^{-x} \sin(2x)$ .

c) Using the structural result  $y = y_p + y_h$  we conclude that

$$y(x) = (x - 3)e^{-x} \cos(2x) + (2x + 1)e^{-x} \sin(2x) + ae^{2x}, \quad x \in \mathbb{R}$$

is a general solution.

△

This example has shown that when all three factors are present in the expression on the right, then the calculations can be longer, increasing the chance that we make a mistake. Usually we encounter simpler cases and we handled them in our intuitive exposition in a very simple way: We simply disregard parts that are not there. Does this fit with the procedure as described in the theorem?

Consider a right-hand side of the form  $p(x)e^{\alpha x}$ . This can be written as

$$\begin{aligned} p(x)e^{\alpha x} &= e^{\alpha x}p(x) \cdot 1 = e^{\alpha x}p(x) \cdot \cos(0) = e^{\alpha x}(p(x) \cdot \cos(0x) + \sin(0)) \\ &= e^{\alpha x}(p(x) \cdot \cos(0x) + 1 \cdot \sin(0x)). \end{aligned}$$

We have  $\alpha = 2$ ,  $\beta = 0$ ,  $q(x) = 1$ , so the maximal degree  $d$  is given by  $\deg(p)$ . We should therefore look for a solution in the form

$$\begin{aligned} y(x) &= x^m e^{\alpha x} (\tilde{p}(x) \cdot \cos(0x) + \tilde{q}(x) \cdot \sin(0x)) = x^m e^{\alpha x} (\tilde{p}(x) \cdot 1 + \tilde{q}(x) \cdot 0) \\ &= x^m \tilde{p}(x) e^{\alpha x}, \end{aligned}$$

where  $\tilde{p}$  is a polynomial of the same degree as  $p$ . So indeed, we can simply generalize the polynomial and we get the correct form. The special number is  $\lambda = \alpha$ , this also fits: no (co)sines, so no imaginary part.

Now consider a right-hand side without an exponential. Formally, we can write

$$p(x) \cos(\beta x) + q(x) \sin(\beta x) = e^{0 \cdot x} (p(x) \cos(\beta x) + q(x) \sin(\beta x)).$$

Thus we get  $\alpha = 0$  for this case, so the  $\alpha$  part of the special number indicates presence of an exponential, indeed. When we apply the theorem to form our guess, the exponential  $e^{0 \cdot x}$  in it will disappear again. Note that the guess then has the form

$$x^m \tilde{p}(x) \cos(\beta x) + x^m \tilde{q}(x) \sin(\beta x),$$

where these two polynomials share a common degree equal to the maximum of  $\deg(p)$  and  $\deg(q)$ . Again, this confirms our procedure and the fine points about both groups being necessary and the higher degree of polynomial forcing itself on the other one.

Finally, when we have just a polynomial, we can write it as

$$\begin{aligned} p(x) &= e^{0 \cdot x} p(x) \cdot 1 = e^{0 \cdot x} p(x) \cdot \cos(0) = e^{0 \cdot x} (p(x) \cdot \cos(0x) + \sin(0)) \\ &= e^{0 \cdot x} (p(x) \cdot \cos(0x) + 1 \cdot \sin(0x)). \end{aligned}$$

We see that  $\lambda = 0 + 0i = 0$  is indeed the right special number. When we form the right form of the particular solution according to the theorem, it will collapse again into the shape  $x^m \tilde{p}(x)$ .

We can sum up these observations as follows:

- $b(x) = p(x) \implies y(x) = x^m \tilde{p}(x)$ , where  $m$  is the multiplicity of  $\lambda = 0$ ;
- $b(x) = p(x)e^{\alpha x} \implies y(x) = x^m \tilde{p}(x)e^{\alpha x}$ , where  $m$  is the multiplicity of  $\lambda = \alpha$ ;
- $b(x) = p(x) \sin(\beta x) + q(x) \cos(\beta x) \implies y(x) = x^m [\tilde{p}(x) \sin(\beta x) + \tilde{q}(x) \cos(\beta x)]$ , where  $m$  is the multiplicity of  $\lambda = \beta i$ .

Now we know how to find a particular solution. In the previous chapter we also learned how to solve homogeneous linear equations with constant coefficients, so thanks to theorem we are now actually able to find a general solution for any linear differential equation with constant coefficients. This sets up a two-step procedure that we will always follow.

People usually start with the homogeneous solution, then find the particular one. This makes sense, as we need to know characteristic numbers when inquiring about corrections, and it also makes one feel better when after a few simple steps, half of the work (in some sense) is already done.

Sometimes we are also given initial conditions and asked for a corresponding solution. Then we need to find a general solutions first anyway, so the process of finding it becomes the first stage in the overall procedure.

### Example 16a.f:

Find the solution of  $y'' - y' - 2y = 20 \sin(2x)$  that satisfies the initial conditions  $y(0) = 14$ ,  $y'(0) = -19$ .

What is the plan? There are two major steps, and the first one splits into two sub-steps:

1. We find the general solution:

- a) we find a general  $y_h$ ,  
 b) we find some  $y_p$ ,  
 and  $y = y_p + y_h$  will be a general solution.

2. We handle the initial conditions.

It is a good idea to keep this general structure in mind, some students like to draw a flow chart. We note that the equation is linear and has constant coefficients, so our procedures apply. Here we go:

1a) The associated homogeneous equation reads  $y'' - y' - 2y = 0$ . We solve  $\lambda^2 - \lambda - 2 = 0$  to find  $\lambda = -1, 2$ , which leads to a general homogeneous solution

$$y_h(x) = a e^{-x} + b e^{2x}, \quad x \in \mathbb{R}.$$

1b) To find a particular solution to the equation with right-hand side  $20 \sin(2x)$  we will use the guessing method. There is no exponential there, but we see a sine, so we keep it and generalize the polynomial of degree zero:  $A \sin(2x)$ . Then we also create another term of analogous type with cosine, the basic shape of the solution should therefore be  $A \sin(2x) + B \cos(2x)$ .

The special number of this right-hand side is  $\lambda = 2i$  (no exponentials so  $0 + 2i$ ). This does not match any of the characteristic numbers, so no correction needed.

Thus we will look for a solution of the form

$$y_p(x) = A \cos(2x) + B \sin(2x).$$

We substitute it into the left-hand side of the equation:

$$\begin{aligned} L &= [A \cos(2x) + B \sin(2x)]'' - [A \cos(2x) + B \sin(2x)]' - 2 \cdot [A \cos(2x) + B \sin(2x)] \\ &= -4A \cos(3x) - 4B \sin(3x) + 2A \sin(3x) - 2B \cos(3x) - 2A \cos(3x) - 2B \sin(3x) \\ &= (-6A - 2B) \cos(3x) + (2A - 6B) \sin(3x). \end{aligned}$$

We want this to be  $0 \cdot \cos(2x) + 20 \sin(2x)$ , comparing we obtain a simple system of two equations:

$$\begin{aligned} -6A - 2B &= 0 \\ 2A - 6B &= 20 \end{aligned} \implies A = 1, \quad B = -3.$$

We deduced that the function  $y_p(x) = \cos(2x) - 3 \sin(2x)$  solves the given equation.

Consequently, we have a general solution of the given equation given by the formula

$$y(x) = \cos(2x) - 3 \sin(2x) + a e^{-x} + b e^{2x}, \quad x \in \mathbb{R}.$$

2. We need to determine  $a, b$  so that the resulting function satisfies the given conditions. First we prepare the derivative:

$$y'(x) = -2 \sin(2x) - 6 \cos(2x) - a e^{-x} + 2b e^{2x}.$$

Now, having formulas for  $y$  and  $y'$ , we can rewrite the given conditions:

$$\begin{aligned} \cos(2 \cdot 0) - 3 \sin(2 \cdot 0) + a e^{-0} + b e^{2 \cdot 0} &= 14 \\ -2 \sin(2 \cdot 0) - 6 \cos(2 \cdot 0) - a e^{-0} + 2b e^{2 \cdot 0} &= -19, \end{aligned}$$

that is,

$$\begin{aligned} a + b &= 13 \\ -a + 2b &= -13. \end{aligned}$$

We find that  $a = 13, b = 0$ . The solution of the given problem is

$$y(x) = \cos(2x) - 3 \sin(2x) + 13e^{-x}, \quad x \in \mathbb{R}.$$

△

Note that we process the initial conditions only after we build the complete general solution. It does not make much sense adjusting the homogeneous solution to satisfy  $y(0) = 14$  and the spoil it by adding  $y_p$  to it.

Note also the change of focus. When solving an equation  $L[y] = b(x)$ , we first look only at the

left-hand side and find the homogeneous solution. Then we look only at the right-hand side to find the basic shape of the solution. Only in the last stage we actually compare the behavior of the two sides and see whether a correction should be applied.

There is one more situation to work out. Sometimes the right-hand side does not fit the right pattern, but it is composed of parts that are special. If these parts are added or subtracted, we can use linearity to our advantage again.

**Theorem 16a.3.** (superposition principle)

Consider a linear ODE with left hand-side

$$L[y] = y^{(n)} + a_{n-1}(x)y^{(n-1)} + \dots + a_1(x)y' + a_0(x)y.$$

Let  $y_1$  be a solution of  $L[y] = b_1(x)$  on an open interval  $I$

and  $y_2$  be a solution of  $L[y] = b_2(x)$  on  $I$ .

Then  $y_1 + y_2$  is a solution of  $L[y] = b_1(x) + b_2(x)$  on  $I$ .

This is a mathematical way to see this statement, it makes for a simple proof.

**Proof:** By our assumption,  $L[y_1] = b_1$  and  $L[y_2] = b_2$ . By linearity then

$$L[y_1 + y_2] = L[y_1] + L[y_2] = b_1 + b_2.$$

The proof is complete. □

The practical view would read this theorem backwards. If we want to solve an equation of the form  $L[y] = b_1(x) + b_2(x)$ , then we just solve the simpler equations  $L[y] = b_1$  and  $L[y] = b_2$  and add these solutions. Traditionally we do not quite solve each equations as a complete separate problem. Since they share the same associated homogeneous equation, it is pointless to do the same work twice. Instead we apply this theorem to the phase when we look for a particular solution  $y_p$ . Some people would find two particular solutions and add them, I actually prefer to apply this theorem to the stage where we guess the shape of the solution.

**Example 16a.g:** We find the general solution of  $y'' - 2y' + y = 27e^{-2x} - 2e^x + 1$ .

There are two steps to be worked out.

a) Homogeneous solution: The equation  $y'' - 2y' + y = 0$  has the characteristic polynomial  $\lambda^2 - 2\lambda + 1 = (\lambda - 1)^2$ , so there is just one characteristic number  $\lambda = 1$ , but with multiplicity two. We have to recall the procedure for this situation and conclude that we have a general solution

$$y_h(x) = a e^x + b x e^x, \quad x \in \mathbb{R}.$$

It is a good idea to check that things make sense as you go through the procedure. As an equation of order two, the homogeneous solution must have two independent factors in it. If you forget about the special trick for characteristic values of higher multiplicity and use  $y_h(x) = a e^x$ , you should immediately become suspicious when doing the “does it make sense” check.

b) Now we should handle the right-hand side  $b(x) = 27e^{-2x} - 2e^x + 1$ .

We notice two distinct exponentials there, so it is obvious that we cannot put the three parts together into one expression of special type; we will have to handle each term separately.

- $27e^{-2x}$ : We create our guess by preserving the exponential and generalizing the polynomial  $p(x) = 27$  of degree zero. The basic guess is  $A e^{2x}$ . The special number  $\lambda = -2$  does not match any of the characteristic numbers, so no correction is needed.

- $2e^x$ : This is a term of the same type as the previous one, so we proceed as before and create the basic guess  $B e^x$ . Note that we did not use  $A$  here, because we will be adding the guesses and  $A$  already plays a role in the first part of our guess above.

The special number  $\lambda = 1$  matches one of the characteristic numbers, so we have to make a correction. Multiplicity of the characteristic number  $\lambda = 2$  is two, so there is a solution of the form  $x^2 e^x$ .

Note that we can see the overlap also on the level of the functions. The part  $2e^x$  has the exponential  $e^x$  as the main part, and exactly the same function happens to be in the fundamental system of the homogeneous equation. We also see  $x e^x$  there, so one correction is not enough.

Note also that we did not put a full second-degree polynomial here, just  $x^2$ . The rule about putting a complete polynomial applies only to the part where we replace specific polynomials from the right-hand side by general ones. Here  $x^2$  is a corrective factor (protection), a totally different story.

• 1: There are no exponentials or trig functions to preserve. We just see the polynomial  $p(x) = 1$  of degree zero, so we replace it with a general one and obtain the basic guess  $C$ . Again, we have to avoid the constants used before. The special number  $\lambda = 0$  does not match any of the characteristic numbers, so no correction needed.

These three parts are added in the given equation, so by the superposition theorem we obtain our solution by adding the three guesses:

$$y_p(x) = A e^{-2x} + B x^2 e^x + C.$$

One can also use  $y_p(x) = A e^{-2x} - B x^2 e^x + C$  to conform better with the original form. As we explained before, the constants in our guess adjust, so it does not make any difference at the end. I prefer addition to subtraction, so I will stick with my original form.

Now we substitute this into the left-hand side of the given equation:

$$\begin{aligned} L &= [A e^{-2x} + B x^2 e^x + C]'' - 2[A e^{-2x} + B x^2 e^x + C]' + [A e^{-2x} + B x^2 e^x + C] \\ &= 4A e^{-2x} + 2B e^x + 4B x e^x + B x^2 e^x - 2(-2A e^{-2x} + 2B x e^x + B x^2 e^x) \\ &\quad + A e^{-2x} + B x^2 e^x + C \\ &= 9A e^{-2x} + 2B e^x + C. \end{aligned}$$

Note that the terms  $x^2 e^x$  and  $x e^x$  disappeared, they fulfilled their role as protection of  $e^x$ . We want this output to be equal to  $27e^{-2x} - 2e^x + 1$ . We again compare the three parts separately and obtain  $A = 3$ ,  $B = -1$ , and  $C = 1$ . The particular solution is  $y_p(x) = 3e^{-2x} - x^2 e^x + 1$ .

Now we apply the  $y_p + y_h$  theorem to see that the given equation has a general solution

$$y(x) = 3e^{-2x} - x^2 e^x + 1 + a e^x + b x e^x, \quad x \in \mathbb{R}.$$

Note that the three distinct types stayed separated throughout the whole procedure. The derivatives and simplifications in  $L$  were all done in parallel, the three components never interfered with one another, say, the constant  $A$  from  $e^{-2x}$  never mixed up with the other two parts and vice versa. It always works like this. We in fact did three processes at once.

Some people prefer to do them separately. They would find the guess  $A e^{-2x}$  for the part  $27e^{-2x}$  and proceed to the “substitute and compare” stage:

$$L = \dots = 9A e^{-2x} \stackrel{?}{=} 27e^{-2x},$$

hence  $A = 3$  and we have the first part of the particular solution:  $y_{p1} = 3e^{-2x}$ . Similarly we can process the other two parts, obtaining  $y_{p2} = -x^2 e^x$  and  $y_{p3} = 1$ . We then connect them together, obtaining  $y_p = 3e^{-2x} - x^2 e^x + 1$ .

The overall amount of calculations is the same with both approaches. I prefer the first one, when we add already the guesses, because it cuts down on overhead. Some people prefer the individual treatment and put up with a bit more overhead, because they feel that it is safer to do the calculations in smaller bits.

△

**16a.4 Remark:** This was the last expository example. Before we outline the general procedure, it is worth noting two important points.

1. When we see a sum of terms on the right in our equation, we should not automatically handle each one separately. Some terms belong naturally together. For instance, if we see the expression  $5e^x \cos(13x) + e^x \sin(13x)$  on the right, we should recognize that they form a natural pair (the same exponential, the same frequency in trig functions) and we create just one pair for them in our guess. After all, it can be written as  $e^x(5 \cos(13x) + \sin(13x))$ . Similarly, it is not a good idea to treat  $x^2 - 1$  as two problems. It is just one polynomial.

What happens if we do not heed this advice? The proper way to handle  $x^2 - 1$  is to replace it with  $Ax^2 + Bx + C$  when forming our guess. There may be also some correction, but that is not the point now. If we mistakenly create separate guesses for  $x^2$  and 1, we end up with the guess  $Ax^2 + Bx + C + D$ . Note that there is one extra unknown parameter, so we do not get a unique solution when solving for them. This is not a fatal problem, we simply choose some solution, but it is a needless complication that increases the probability that we make some mistake; what is worse, it shows the examiner that we did not truly master the topic.

How do we recognize which terms belong together and which should be handled separately? By their special numbers. Terms with the same special number belong together.

2. When we form our guess and pass it through  $L$ , then the output must match the right-hand side in its form. In particular, comparing the output with the right-hand side should yield unique values for the unknown coefficients, and these values should be numbers. If we have more parameters than equations, then we overdid it somewhere in the guess. It can also happen that there may be a term missing in the output.

For instance, imagine that we would forget to put a double protection with  $e^x$ . Then the calculations would go as follows:

$$L[Ae^{-2x} + Bxe^x + C] = 9Ae^{-2x} + C \stackrel{?}{=} 27e^{-2x} - 2e^x + 1.$$

There is no  $e^x$  in the output, a clear sign that we did not do something right in our guess. Missing terms typically point towards forgotten correction, another typical cause is an error in algebra while wrestling with the expression for  $L$ .

Sometimes students try to fix this by magic. For instance, one could try

$$9Ae^{-2x} = 27e^{-2x} - 2e^x \implies 9Ae^{-2x} = (27 - 2e^{3x})e^{-2x} \implies A = 3 - \frac{2}{9}e^{3x}.$$

This does not work. The unknown parameters can only have real numbers as their values. If you feel tempted to write things like  $B = 2x$ , stop and check again that your guess was properly done and that your calculations were correct.

△

It is time to capture the whole process of solving linear differential equations.

### Algorithm 16a.5.

⟨solving linear differential equations with special right-hand side⟩

Given: a linear ODE  $y^{(n)} + a_{n-1}y^{(n-1)} + \dots + a_1y' + a_0y = b(x)$ ,

possibly also initial conditions  $y^{(k)}(x_0) = y_k$  for  $k = 0, 1, \dots, n-1$ .

Assumptions: The coefficients  $a_i$  are constant, and  $b(x)$  is a sum (or difference) of terms of the form  $p(x)e^{\alpha x} \cos(\beta x)$  or  $p(x)e^{\alpha x} \sin(\beta x)$  for some polynomials  $p(x)$ ; it is understood that the exponential component or the trig function component may be missing.

1. Find the general solution.

(a) Solve the associated homogeneous equation.

- Form its characteristic equation, solve it to obtain characteristic numbers  $\lambda_i$ .
- Create a fundamental system  $\{y_1, \dots, y_n\}$  according to rules from Theorem .
- Form the general “homogeneous solution”  $y_h(x) = \sum_{k=1}^n c_k y_k(x)$ .

(b) Find a particular solution.

- For each term  $p(x)e^{\alpha x} \cos(\beta x)$  or  $p(x)e^{\alpha x} \sin(\beta x)$  in the right-hand side, determine its special number  $\lambda = \alpha + \beta i$ . Terms with the same special numbers are treated as one group.
- For each term (or group of terms), set up the basic form of solution: Exponentials and trig functions are retained if present, and all polynomials (including an implied 1) are replaced by full general polynomials of the same degree.

If the term/group in question features trig functions, then the guess must include both a sine group and a cosine group, and both polynomials are of the largest degree present in the processed term/group.

After the basic form of a guess was created for the term/group, compare the special number  $\lambda$  of this term/group with characteristic numbers. If there is a match, multiply your guess by  $x^m$ , where  $m$  is the multiplicity of  $\lambda$  as a characteristic number.

Make sure that as you are creating guesses for each term/group, you use new letters for unknown parameters in polynomials; polynomials from different guesses cannot share names of parameters.

- Add all the guesses for terms/groups. This is the estimated form of the partial solution.
- Substitute this guess into the left-hand side of the equation and simplify to obtain an expression with structure identical to the right-hand side.

By comparing this output with the right-hand side, obtain a systems of equations featuring unknown coefficients, solve it.

- Using the now determined coefficients, write the particular solution  $y_p$ .

(c) The general solution is  $y = y_p + y_h$ . Do not forget to specify the region where this solution is valid (namely  $x \in \mathbb{R}$ ).

2. If initial conditions were given, substitute the known general solution for  $y$  in them and solve the resulting system of equations. This determines the values of coefficients  $c_i$ .

Write down the desired particular solution (including region of validity).

△

The single most frequent source of errors is the calculations after we substitute our guess into the left-hand side  $L$ . Sometimes people make a mistake in their derivatives, more often they miscopy a sign or a number when herding terms after substituting into  $L$ . Take your time.

The other frequent problem is not getting the guess right. It is actually quite simple, one just has to understand properly the underlying mechanism.

**Example 16a.h:** Consider the totally unrealistic equation

$$y' - 2y = 2e^{2x} \cos(x) + 13x e^{2x} + e^{-2x} - x e^{2x} \sin(x) - x^2 + \sin(2x) + 13.$$

We will show how to create the right shape of the particular solution. We start with the homogeneous equation:  $y' - 2y = 0$ . We see the characteristic number  $\lambda = 2$  of multiplicity 1.

Now we scan the right-hand side and assign each term its special number.

$2e^{2x} \cos(x)$	$13x e^{2x}$	$e^{-2x}$	$x e^{2x} \sin(x)$	$x^2$	$\sin(2x)$	13
$\lambda = 2 + i$	$\lambda = 2$	$\lambda = -2$	$\lambda = 2 + i$	$\lambda = 0$	$\lambda = 2i$	$\lambda = 0$

We see that some terms belong together. Now we process the terms and groups that we see.

$2e^{2x} \cos(x) - x e^{2x} \sin(x)$ : Basic guess is  $(Ax + B)e^{2x} \cos(x) + (Cx + D)e^{2x} \sin(x)$ . The special number  $\lambda = 2 + i$  is not equal to 2, hence no correction.

$13x e^{2x}$ : Basic guess is  $(Ex + F)e^{2x}$ . The special number  $\lambda = 2$  is equal to 2, hence correction to  $x(Ex + F)e^{2x}$ .

$e^{-2x}$ : Basic guess is  $G e^{-2x}$ . The special number  $\lambda = -2$  is not equal to 2, hence no correction.

$x^2 + 13$ : Basic guess is  $Hx^2 + Ix + J$ . The special number  $\lambda = 0$  is not equal to 2, hence no correction.

$\sin(2x)$ : Basic guess is  $K \cos(2x) + L \sin(2x)$ . The special number  $\lambda = 2i$  is not equal to 2, hence no correction.

According to theory, there should be a particular solution of the form

$$y_p(x) = (Ax + B)e^{2x} \cos(x) + (Cx + D)e^{2x} \sin(x) + (Ex^2 + Fx)e^{2x} + Ge^{-2x} + Hx^2 + Ix + J + K \cos(2x) + L \sin(2x).$$

If you feel like substituting this into the left-hand side and working out the constants, be my guest.  
 $\triangle$

To allow you to practice forming the guesses with more realistic examples we prepared a special exercise below. To explain how it works we show the last example of this section.

**Example 16a.i:** Consider the following table.

$y'' - 4y' + 3y =$	$y'' - 2y' =$	$L(y) = / = b(x)$
		$= e^{3x} - 13x$

This table represents differential equations, namely two of them. In columns we see the left-hand sides. The right-hand sides are marked in rows (just one here), which explains why the heading for the row is (unusually) on the right. For instance, the lower left cell stands for the equation

$$y'' - 4y' + 3y = e^{3x} - 13x.$$

Our task is to fill in the empty cells with guesses for particular solutions.

We start by processing the homogeneous equations. We find the characteristic numbers and we could easily write the homogeneous solutions for them, but we do not need them in this example. We fill in this information in our table. Note that we just use the information from column headings at this stage.

$y'' - 4y' + y =$ $\lambda = 1, 3$	$y'' - 2y' =$ $\lambda = 0, 2$	$L(y) = /$ $/ = b(x)$
		$= e^{3x} - 13x$ $3, 0$

Next we turn to the right-hand side. Now we do not care about the columns, and just based on the expression  $e^{3x} - 13x$  we conclude that we in fact have two terms with special numbers 3 and 0, and prepare the basic form of a particular solution:

$$Ae^{3x} + Bx + C.$$

Finally we compare information from rows and columns. For each term we check on the special number, and if there is a match, we apply correction.

$y'' - 4y' + y =$ $\lambda = 1, 3$	$y'' - 2y' =$ $\lambda = 0, 2$	$L(y) = /$ $/ = b(x)$
$xAe^{3x} + Bx + C$	$Ae^{3x} + x(Bx + C)$	$= e^{3x} - 13x$ $3, 0$

Does it make sense? There is more to come in exercises.

$\triangle$

We return to this method in the last section of this chapter.

## 16b. Solving linear ODEs by variation

In chapter 9 we used variation to solve nonhomogeneous linear equations of order 1. The substance can be captured in the following steps, applied to a given equation of the form  $y' + a(x)y = b(x)$ .

- Find a general solution  $y_h = c \cdot u(x)$  of the associated homogeneous equation.
- Decide to look for a particular solution of the form  $y_p(x) = c(x) \cdot u(x)$ . Substitute this guess into the original equation and obtain a condition for  $c'(x)$ . Thanks to a miracle of cancelling, it

has the convenient form of the equation

$$c'(x)u(x) = b(x).$$

Solve for  $c'(x)$ , integrate to obtain some antiderivative  $c(x)$ .

• Now we have a particular solution  $y_p(x) = c(x)u(x)$ , using the  $y_p + y_h$  formula we obtain a general solution of the given equation.

Variation also works for higher order, but then it gets more complicated.

**Example 16b.a:** Consider the equation  $y'' - 3y' + 2y = 10e^{3x}$ .

In example 16a.d we found its general homogeneous solution

$$y_h(x) = a e^x + b e^{2x}, \quad x \in \mathbb{R}$$

Following the hint, we will now try to find a particular solution of the form

$$y_p(x) = a(x) e^x + b(x) e^{2x}.$$

We want to substitute this into the given equation, and we start by preparing the derivatives:

$$y_p'(x) = a'(x) e^x + a(x) e^x + b'(x) e^{2x} + 2b(x) e^{2x},$$

$$y_p''(x) = a''(x) e^x + 2a'(x) e^x + a(x) e^x + b''(x) e^{2x} + 4b'(x) e^{2x} + 4b(x) e^{2x}.$$

Now we substitute:

$$\begin{aligned} L &= a''(x) e^x + 2a'(x) e^x + a(x) e^x + b''(x) e^{2x} + 4b'(x) e^{2x} + 4b(x) e^{2x} \\ &\quad - 3(a'(x) e^x + a(x) e^x + b'(x) e^{2x} + 2b(x) e^{2x}) + 2(a(x) e^x + b(x) e^{2x}) \\ &= a''(x) e^x - a'(x) e^x + b''(x) e^{2x} + b'(x) e^{2x}. \end{aligned}$$

Comparing this to the right-hand side we obtain an equation.

$$a''(x) e^x - a'(x) e^x + b''(x) e^{2x} + b'(x) e^{2x} = 10e^{3x}.$$

We see two major problems. The first is conceptual: We have just one equation, but two unknown functions  $a(x)$ ,  $b(x)$ . This is usually a good news, we simply choose one unknown as we wish and solve for the other. However, here it is not clear what choice of one unknown function would make our life easy.

The second problem is practical. When we used variation for equations of order one, the miracle of cancelling removed  $c(x)$  and we had one equation featuring only  $c'(x)$ . Now the miracle of cancelling removed  $a(x)$ ,  $b(x)$ , which is undoubtedly great, but we still have two different derivatives of the unknown functions in our equation, which is pretty bad.

People came up with an interesting idea. The root of the problem was traced to the preparatory calculation of derivatives, where we allowed higher derivatives of the unknown functions to appear. Since the substituting phase provides us with just one equation, we are free to impose another condition, and we use it to simplify the derivatives.

We start with

$$y_p'(x) = a(x) e^x + 2b(x) e^{2x} + a'(x) e^x + b'(x) e^{2x}.$$

Note that we did use the product rule where needed, but we grouped the resulting terms differently. Now we do not mind those on the left, but we are trying to prevent derivatives of  $a(x)$  and  $b(x)$  to appear, so we will now require that

$$a'(x) e^x + b'(x) e^{2x} = 0.$$

Then

$$y_p'(x) = a(x) e^x + 2b(x) e^{2x}$$

and therefore

$$y_p''(x) = a(x) e^x + 4b(x) e^{2x} + a'(x) e^x + 2b'(x) e^{2x}.$$

We would like to say that  $a'(x) e^x + 2b'(x) e^{2x} = 0$ , and we could do it if our equation was of higher order. However, we ran out of conditions that we are free to use, so we will accept it as it is and

now do the substituting phase:

$$\begin{aligned} L &= a(x) e^x + 4b(x) e^{2x} + a'(x) e^x + 2b'(x) e^{2x} \\ &\quad - 3(a(x) e^x + 2b(x) e^{2x}) + 2(a(x) e^x + b(x) e^{2x}) \\ &= a'(x) e^x + 2b'(x) e^{2x} \stackrel{?}{=} 10e^{3x}. \end{aligned}$$

Again, the miracle of cancelling removed  $a(x)$  and  $b(x)$ , but this time there are no higher derivatives, so this is a very nice equation. Together with the condition that we imposed it forms a consistent system of two equations with two unknowns  $a'(x)$ ,  $b'(x)$ :

$$\begin{aligned} a'(x) e^x + b'(x) e^{2x} &= 0, \\ a'(x) e^x + 2b'(x) e^{2x} &= 10e^{3x}. \end{aligned}$$

Before we move on, notice that we can capture the whole process as follows:

$$\begin{aligned} y_p(x) = a(x) e^x + b(x) e^{2x} &\implies \\ a'(x) e^x + b'(x) e^{2x} &= 0, \\ a'(x)[e^x]' + b'(x)[e^{2x}]' &= 10e^{3x}. \end{aligned}$$

This is a pattern that we will develop below.

Returning to our example, we have a 2 by 2 system to solve. Elimination is definitely possible, either intuitive (it seems that subtracting the first equation from the second is a very good idea) or formal by working with a matrix of the system. Interestingly, the Cramer rule that is not quite popular for systems of algebraic equations can be very viable when we deal with functions. Indeed,

$$\begin{aligned} D &= \det \begin{vmatrix} e^x & e^{2x} \\ e^x & 2e^{2x} \end{vmatrix} = e^{3x}, \\ D_a &= \det \begin{vmatrix} 0 & e^{2x} \\ 10e^{3x} & 2e^{2x} \end{vmatrix} = -10e^{5x}, \\ D_b &= \det \begin{vmatrix} e^x & 0 \\ e^x & 10e^{3x} \end{vmatrix} = 10e^{4x}. \end{aligned}$$

Therefore

$$\begin{aligned} a'(x) &= \frac{D_a}{D} = -10e^{2x}, \\ b'(x) &= \frac{D_b}{D} = 10e^x. \end{aligned}$$

Consequently, we easily find antiderivatives  $a(x) = -5e^{2x}$ ,  $b(x) = 10e^x$ , and substituting them into our guess we obtain the desired particular solution:

$$y_p(x) = -5e^{2x} \cdot e^x + 10e^x \cdot e^{2x} = 5e^{3x}.$$

We thus have a general solution

$$y(x) = 5e^{3x} + a e^x + b e^{2x}, \quad x \in \mathbb{R}.$$

Note that in example 16a.a we found the same particular solution using the guessing method. When we apply both the guessing method and the method of variation to the same problem, then surprisingly often the resulting particular solutions agree, but it is not a rule.

Before we leave this example, it is worth noting an interesting connection with theory. The matrix of the system

$$\begin{pmatrix} e^x & e^{2x} \\ e^x & 2e^{2x} \end{pmatrix}$$

comes from the two equations that we derived, and the coefficients in those equations come from the form of the homogeneous solution. In other words, in the first row we see the fundamental system of the equation, and in the second row we see the derivatives. This is in fact the Wronski

matrix for the fundamental system, and therefore  $D$  is the Wronskian  $W(x)$ . Since the fundamental system consists of linearly independent functions, the Wronskian can never be zero, that is,  $D \neq 0$  and the system that we obtain from variation is always solvable.

△

The example gave us inspiration for a general blueprint for variation. We take some linear ODE of order  $n$ , typically more than 1.

$$y^{(n)} + \sum_{k=0}^{n-1} a_k(x)y^{(k)} = b(x).$$

Since the solutions of the associated homogeneous equation form a linear space of dimension  $n$ , we can always find a general homogeneous solution of the form

$$y_h = \sum_{i=1}^n c_i u_i(x).$$

For higher order this requires that the coefficients of the equation are constant.

We start looking for a solution of the form  $y_p = \sum c_i(x)u_i(x)$ , where  $c_i(x)$  are  $n$  unknown functions to be determined. The general strategy is to substitute this guess into the given equation and derive an equation. Inspired by our observations above, we first prepare derivatives, and use the fact that we are free to demand that  $n - 1$  conditions be met to simplify those derivatives.

The first derivative of  $y_p$  is

$$y_p' = \sum c_i(x)u_i'(x) + \sum c_i'(x)u_i(x).$$

We will insist that

$$\sum c_i'(x)u_i(x) = 0, \tag{1}$$

then  $y_p' = \sum c_i(x)u_i'(x)$  and we easily find

$$y_p'' = \sum c_i(x)u_i''(x) + \sum c_i'(x)u_i'(x).$$

Now we insist that

$$\sum c_i'(x)u_i'(x) = 0, \tag{2}$$

then  $y_p'' = \sum c_i(x)u_i''(x)$  and we obtain

$$y_p''' = \sum c_i(x)u_i'''(x) + \sum c_i'(x)u_i''(x).$$

Note the pattern. When do we stop? We can pose  $n - 1$  conditions, so the last conditions that we can state is

$$\sum c_i'(x)u_i^{(n-2)}(x) = 0, \tag{n-1}$$

then  $y_p^{(n-1)} = \sum c_i(x)u_i^{(n-1)}(x)$  and we obtain

$$y_p^{(n)} = \sum c_i(x)u_i^{(n)}(x) + \sum c_i'(x)u_i^{(n-1)}(x).$$

This happens to be the highest derivative needed in our equation, so things fit nicely. When we substitute all these (simplified) derivatives into the given equation, then the fact that each  $u_i(x)$

solves the associated homogeneous equation makes the unknown functions disappear:

$$\begin{aligned}
 L &= \left[ \sum_{i=1}^n c_i(x) u_i^{(n)}(x) + \sum_{i=1}^n c'_i(x) u_i^{(n-1)}(x) \right] + \sum_{k=0}^{n-1} a_k(x) \sum_{i=1}^n c_i(x) u_i^{(k)}(x) \\
 &= \sum_{i=1}^n c'_i(x) u_i^{(n-1)}(x) + \sum_{i=1}^n \left[ c_i(x) u_i^{(n)}(x) + \sum_{k=0}^{n-1} a_k(x) c_i(x) u_i^{(k)}(x) \right] \\
 &= \sum_{i=1}^n c'_i(x) u_i^{(n-1)}(x) + \sum_{i=1}^n c_i(x) \left[ u_i^{(n)}(x) + \sum_{k=0}^{n-1} a_k(x) u_i^{(k)}(x) \right] \\
 &= \sum_{i=1}^n c'_i(x) u_i^{(n-1)}(x) + 0 \\
 &= \sum_{i=1}^n c'_i(x) u_i^{(n-1)}(x) \stackrel{?}{=} b(x).
 \end{aligned}$$

The miracle of cancelling is confirmed, and we obtain the main equation

$$\sum_{i=1}^n c'_i(x) u_i^{(n-1)}(x) = b(x).$$

Note that the variation process works even with general coefficients  $a_k(x)$ , we do not need constant coefficients for the cancelling to work.

This main equation has the same form as the conditions 1 through  $n - 1$  that we stated above, and we obtain the following system for the unknowns  $c'_1(x), \dots, c'_n(x)$ :

$$\begin{aligned}
 \sum_{i=1}^n c'_i(x) u_i(x) &= 0 \\
 \sum_{i=1}^n c'_i(x) u'_i(x) &= 0 \\
 \sum_{i=1}^n c'_i(x) u''_i(x) &= 0 \\
 &\vdots \\
 \sum_{i=1}^n c'_i(x) u_i^{(n-1)}(x) &= b(x).
 \end{aligned}$$

Given the computational demand of this process, people typically do not actually work out the miracle of cancelling, they simply remember that when we form a guess of the form  $\sum c_i(x) u_i(x)$  based on a basis  $\{u_i(x)\}$  of solutions for the associated homogeneous system, then we have to solve the above system. As noted above, the matrix of this system

$$\begin{pmatrix}
 u_1(x) & u_2(x) & \dots & u_n(x) \\
 u'_1(x) & u'_2(x) & \dots & u'_n(x) \\
 \vdots & \vdots & \ddots & \vdots \\
 u_1^{(n-1)}(x) & u_2^{(n-1)}(x) & \dots & u_n^{(n-1)}(x)
 \end{pmatrix}$$

is in fact the Wronski matrix for the fundamental system  $\{u_i(x)\}$ , and therefore it is never singular. We can always solve for the unknown derivatives  $c'_i(x)$ . The success of variation then depends on whether we can actually integrate those functions, and that is difficult to predict in general.

We have just proved that the idea of variation can be adapted also for equations of higher order and always leads to a solvable system. We are ready to outline the official procedure.

**Algorithm 16b.1.**

⟨variation of parameters for solving LODE⟩

Given: equation  $y^{(n)} + a_{n-1}y^{(n-1)} + \dots + a_1y' + a_0y = b(x)$ .

1. Using characteristic numbers, find a general solution  $y_h$  of the associated homogeneous equation. It has the form

$$y_h(x) = c_1 \cdot u_1(x) + \dots + c_n \cdot u_n(x).$$

2. Variation of parameter: Seek a solution of the form  $y(x) = c_1(x) \cdot u_1(x) + \dots + c_n(x) \cdot u_n(x)$ .

Unknown functions  $c_i(x)$  are found by solving the system of equations

$$c'_1(x)u_1(x) + \dots + c'_n(x)u_n(x) = 0$$

$$c'_1(x)u'_1(x) + \dots + c'_n(x)u'_n(x) = 0$$

$$\vdots$$

$$c'_1(x)u_1^{(n-2)}(x) + \dots + c'_n(x)u_n^{(n-2)}(x) = 0$$

$$c'_1(x)u_1^{(n-1)}(x) + \dots + c'_n(x)u_n^{(n-1)}(x) = b(x)$$

Solve for  $c'_i(x)$ , integrating them you get  $c_i(x)$ , substitute these into  $y(x) = \sum c_i(x)u_i(x)$ .

3. If you take for each  $c_i(x)$  one particular antiderivative, then you get one particular solution  $y_p(x)$ , the general solution is then  $y = y_p + y_h$ .

If you include “+ $C_i$ ” when deriving  $c_i(x)$ , then after substituting them into  $y(x) = \sum c_i(x)u_i(x)$  you get the general solution.

△

We justified those supplementary equations by practical considerations. There is also another, entirely different way to arrive at this procedure. It is related to the fact that in applications, linear equations of higher order are often transformed into systems of differential equations of order one. We will discuss this in the appropriate chapter 27.

## 16c. Summary

It is natural to compare variation and the guessing method. Both require linear equations, which is a key assumption. The guessing method also requires that coefficients be constant. The variation as such does not need it, but typically, when the order of our ODE is more than 1, then we need constant coefficients anyway to find the homogeneous solution  $y_h$ . Still, it is nice to know that if we can somehow find  $y_h$  for equations with non-constant coefficients (which we can always do for order 1), then variation can be applied.

The guessing method can be reliably applied: When the right-hand side is special, then it always succeeds. With variation we do not know until we arrive at the integration stage. Interestingly, for equations with constant coefficients and a special right-hand side variation typically leads to integrals that are known, like  $\int p(x)e^{\alpha x} dx$  or  $\int e^{\alpha x} \cos(\beta x) dx$ . In other words, we then have both methods as viable alternatives.

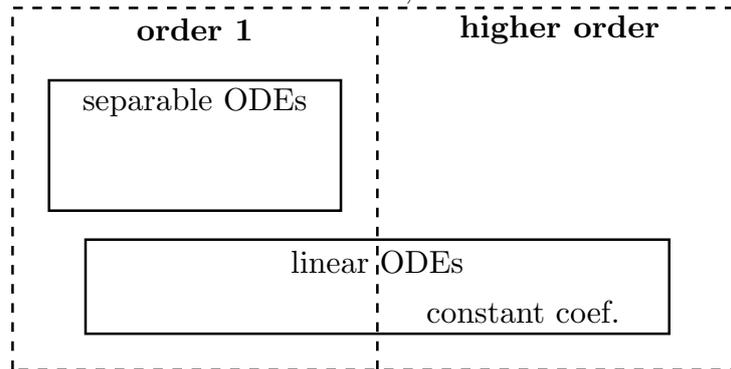
As students of calculus know all too well, such integrals are approached using integration by parts. In other words, given a choice, people prefer the guessing method, as the calculations are typically significantly more friendly. It may therefore come as a surprise that variation seems preferred in actual applications. This is caused by the fact that integrals (and equations) are no longer solved by hand, but by computer algebra systems. For a good computer algebra system, integration is routine, and variation scores high with its streamlined procedure. In particular, we know how the equations for the unknown coefficients  $c'_i(x)$  look like in general, so once the program finds a fundamental system  $\{u_i(x)\}$ , it can directly set up the relevant system and solve it easily.

In contrast, the guessing method requires that we set up the form of a particular solution using a procedure that is relatively complicated, special cases have to be checked on, and the system for unknown constants has to be derived for each equation individually by running the guess

through the differential mapping  $L$  (which the program can do routinely) and then comparing to the right-hand side, which for a computer is far more difficult than for a human.

In short, setting up a computer algorithm for the variation is much easier, and on the top of that this method is also more general. On the other hand, understanding the basic principle of the guessing method often allows us to guess how the solution will behave just by looking at an equation, which can be very useful. It also serves as a check, because computers sometimes supply an incorrect solution and checking answers supplied by calculators and computers against common sense is a good habit.

This concludes our exposition of basic methods for solving individual differential equations. It is good to have a general overview of what we can do, and how we decide which method to use.



Separation requires that our equation be separable, in particular it has to be of order 1. However, its success is not guaranteed, because we may have trouble with integration, or when trying to solve for  $y$ .

The (guessing) method of undetermined coefficients requires that the equation is linear and has constant coefficients; moreover, the right-hand side must be a sum of special terms. Then it is guaranteed to work, as long as we can find characteristic numbers for the left-hand side (which is a problem of being able to find roots of polynomials).

The method of variation requires that the equation is linear. It does not formally require constant coefficients, but for equations of higher order we are limited by our inability to solve homogeneous equations for non-constant cases. It is not guaranteed to work, again the decisive step is integration. For equations of order one, the homogeneous version can be always solved. The method of variation is then equivalent to the method of integrating factor.

There are other types of differential equations for which proper approaches are known, for instance homogeneous equations. However, these types are more specific, so the corresponding methods are not considered basic. Moreover, there is no neat structural theory for such types, which makes them less interesting from purely mathematical point of view.

## 17. Applications of linear ODEs

In this chapter we will look at some application of systems of differential equations.

### 17a. Oscillations

Hookev zakon:  $F = kx$ , kde  $x$  je vchylka z pirozen dlky pruiny a  $k$  je tuhost pruiny. Ale jde v opanm smru, proto  $a = -\frac{k}{m}x$ .

a) Verze bez ten a vnj sly:  $\ddot{x} + \frac{k}{m}x = 0$  neboli  $\ddot{x} + \omega^2x = 0$ , kde jsme oznaili  $\omega = \sqrt{\frac{k}{m}}$ . Mimochodem, rozmrov analza dv pro omegu sekundy.

een je  $a \cos(\omega t) + b \sin(\omega t)$ . Pokud bereme  $t_0 = 0$  (co je tradin), pak  $a = x(0)$ ,  $b = \frac{\dot{x}(0)}{\omega}$ .

Finta:  $A \sin(\omega t - \varphi) = A \sin(\omega t) \cos(\varphi) - A \cos(\omega t) \sin(\varphi)$ . Porovnm dostvme rovnice

$$A \sin(\varphi) = -a, \quad A \cos(\varphi) = b,$$

ze kterch (vydlme, umocnme a seteme) dostvme

$$\operatorname{tg}(\varphi) = -\frac{a}{b}, \quad A = \sqrt{a^2 + b^2}.$$

hel  $\varphi = -\arctg\left(\frac{a}{b}\right)$  je nkdy teba upravit o  $\pi$ , aby dval sprvn znamnka:  $A \sin(\varphi) = -a$  a  $A \cos(\varphi) = b$ .

Podobn je tak mon vyjdit een jako  $A \cos(\omega t - \psi)$ , ale sinus je tradin.

Zvr: Pi kmitn na pruin bez ten je poloha urena funkc  $A \sin(\omega t - \varphi)$ , kde  $A = \sqrt{x(0)^2 + \dot{x}(0)^2}$  a  $\operatorname{tg}(\varphi) = -\omega \frac{x(0)}{\dot{x}(0)}$ . Perioda (doba jednoho plnho kmity) je  $T = \frac{2\pi}{\omega} = 2\pi \sqrt{\frac{m}{k}}$ .

Velk  $k$  dv rychl kmity, to souhlas.

Omezen: Sla zvis na vchylce linern jen pro relativn mal  $x$ . Jakmile pekroim uritou mez, tak dojde k pokozen pruiny.

b) Verze se tenm: Fyzika k, e tec sla je dna jako  $dv(t)$ , kde  $v$  je okamit rychlost, a vede v opanm smru ne rychlost. Dostvme tedy po oznaen  $\omega_0 = \sqrt{\frac{k}{m}}$  rovnici

$$\ddot{x} + d\dot{x} + \omega_0^2x = 0.$$

Je to tzv. tlumen oscilace (damped oscillations), slo  $d$  je koeficient tlumen (damping coefficient) a  $\omega_0$  je zkladn frekvence (base frequency).

een jsou dna vlastnmi sly  $\lambda = -\frac{d}{2} \pm \sqrt{\frac{d^2}{4} - \omega_0^2}$ . Ppady:

•  $d < 2\omega_0$  znamen “underdamped” ppad, kdy je tlumen relativn mal. Vychzej komplexn koeny, po oznaen  $\omega = \sqrt{\frac{d^2}{4} - \omega_0^2}$  mme obecn een

$$x(t) = e^{-(d/2)t} [a \cos(\omega t) + b \sin(\omega t)].$$

Zase to lze pepsat jako  $x(t) = A e^{-(d/2)t} \sin(\omega t - \varphi)$ .

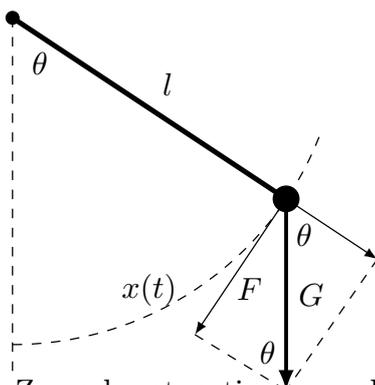
•  $d = 2\omega_0$  je kritick tlumen s jednm dvojnsobnm (a zpornm) koenem  $\lambda$ , vychz

$$x(t) = (at + b)e^{-\lambda t}.$$

•  $d > 2\omega_0$  znamen “overdamped” ppad, kdy je tlumen relativn velk. Vychzej zporn koeny  $\lambda_{1,2}$  a

$$x(t) = a e^{-\lambda_1 t} + b e^{-\lambda_2 t}.$$

### 17b. Matematické kyvadlo



Zva o hmotnosti  $m$  na nehmotnm a neprunm zvsu dlky  $l$  se pohybuje po krunici. Vzdlenost od doln ro

a) Verze bez ten a vnj sly:  $\ddot{\theta} + \frac{g}{l} \sin(\theta) = 0$ . Ozname  $\omega = \sqrt{\frac{g}{l}}$ , dostvme  $\ddot{\theta} + \omega^2 \sin(\theta) = 0$ .

Bohuel, tuto rovnici nelze analyticky vyeit. Vyaduje to tzv. eliptick integrly.

Speciln ppad: Pokud je  $\theta$  stle mal (prakticky vzato  $|\theta| < 0.5$ ), pak lze pout aproximaci  $\sin(\theta) \sim \theta$ .

Dostvme rovnici  $\ddot{\theta} + \omega^2 \theta = 0$  s eenm  $a \cos(\omega t) + b \sin(\omega t)$ . Pokud bereme  $t_0 = 0$  (co je tradin), pak  $a = x(0)$ ,  $b = \frac{\dot{x}(0)}{\omega}$ .

Zase funguje ten pepis  $x(t) = A \sin(\omega t - \varphi)$ , kde  $A = \sqrt{x(0)^2 + \dot{x}(0)^2}$  a  $\text{tg}(\varphi) = -\omega \frac{x(0)}{\dot{x}(0)}$ . Perioda je  $T = \frac{2\pi}{\omega} = 2\pi \sqrt{\frac{l}{g}}$ . m del kyvadlo, tm del perioda.

b) Verze se tenm: Viz pruina, analza obdobn, vznikaj tlumen kmity.

U pruiny i kyvadla je mon uvaovat nehomogenn rovnici v ppad tlumench kmit, pak mme nucen kmitn.

### 17c. Kyvadlo

U matematickho kyvadla se pedpokld zvs s nulovou hmotnost. Pokud uvaujeme zvs skuten, pak je teba brt v vahu jeho hmotnost a pracovat s momentem hybnosti msto zrychlenm.

Pokud m rovn stejnomrn ty o hmotnosti  $m_l$  a dlce  $l$  zvs ve vzdlenosti  $r$  od stedu, tak m moment hybnosti  $I = \frac{m_l l^2}{12} + m_l r^2$  a porovnn hybnost dv  $I\ddot{\theta} + rmg \sin(\theta) = 0$ .

Kdy seteme relnou ty uchycenou na konci a zva, dostaneme

$$(I + \frac{1}{2}ml^2)\ddot{\theta} + \frac{1}{2}mlg \sin(\theta) = 0.$$

Take stejn rovnice, jen jin konstanty.

### 17d. Pružné kyvadlo

Uvaujme zva o hmotnosti  $m$  zaven na pruin s tuhost  $k$  a klidovou dlkou  $l$ , jej hmotnost budeme ignorovat, stejn jako odpor vzduchu, odpor v zvsu a podobn. Zavedeme polrn souadnice  $(r, \theta)$  pro sted zva, potek je pirozen v zvsu. Jak rovnice dostaneme? Obvykle se vol pstup pes Lagrangin  $L = T - V$ , kde  $T$  je kinetick energie a  $V$  potencionln. U naeho kyvadla, kter se jednak kv a druhak kmit, bude teba kombinovat zdroje. U kinetick energie rozlome pohyb na sloku radiln a sloku tangenciln, ke kterm mme pm pstup ze souadnic. U potenciln energie bude u gravitan nutno spotat vku nad referenn rovinou a najt vzorec pro energii z pruiny.

$$T = \frac{1}{2}m\dot{r}^2 + \frac{1}{2}m(r\dot{\theta})^2,$$

$$V = -mgr \cos(\theta) + \frac{1}{2}k(r - l)^2,$$

$$\implies L = \frac{1}{2}m\dot{r}^2 + \frac{1}{2}m(r\dot{\theta})^2 + mgr \cos(\theta) - \frac{1}{2}k(r - l)^2.$$

Zkladn princip k, e proda vol takovou akci, kter minimalizuje celkovou spotebu energie. To vede na nulovost derivace odpovdajcho opertoru a rovnice  $\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{s}} \right) - \frac{\partial L}{\partial s} = 0$  vi vem souadnicm  $s$ , v

naem ppad  $r$  a  $\theta$ . Vylezou z toho rovnice

$$\begin{aligned}\ddot{r} + (\omega_r^2 - \dot{\theta}^2)r &= g \cos(\theta) + l\omega_r^2, \\ \ddot{\theta} + \frac{2}{r}\dot{r} \cdot \dot{\theta} + \frac{g}{r} \sin(\theta) &= 0,\end{aligned}$$

kde  $\omega_r = \sqrt{\frac{k}{m}}$ . Vidme, e prvni rovnice je netlumen, nucen kmitn s periodou zvisejc na okamit vchylce kyvu, zatmco druh rovnice popisuje tlumen nenucen kvn, kde tlumen a frekvence zvisej pro zmnu na okamit situaci kmitn.

Existuj dal pepisy, napklad je mon zavst efektivn dlku kyvadla  $l_g = l + \frac{mg}{k}$ , co je klidov dlka kyvadla se zavenm zvam, dle  $\omega_\theta = \sqrt{\frac{g}{l_g}}$ , omezit se na mal kmity, kdy  $\cos(\theta) \sim 1$  a  $\sin(\theta) \sim \theta$ , nae rovnice pejdou v

$$\begin{aligned}\ddot{r} + \omega_r^2 r &= l_g(\omega_r + \dot{\theta}^2 - \frac{1}{2}\omega_\theta^2\theta^2), \\ \ddot{\theta} + \frac{2}{l_g}\dot{r} \cdot \dot{\theta} + \omega_\theta^2\left(2 - \frac{r}{l_g}\right)\theta &= 0.\end{aligned}$$

K rezonanci dojde piblin v situaci, kdy  $\omega_r = 2\omega_\theta$ , neboli pro  $m \sim \frac{lk}{3g}$ . To mimochodem znamen, e  $l_g = \frac{4}{3}l$ , tedy experimentln prost vybereme takov zva, kter po zaven prodlou pruinu o tetinu.

Zajmav alternativa je pout kartzsk souadnice, kdy po oznaen  $\lambda = \frac{\omega_r^2 - \omega_\theta^2}{l_g}$  vychzej rovnice

$$\begin{aligned}\ddot{x} + (\lambda y + \omega_\theta^2)x &= 0, \\ \ddot{y} + \omega_r^2 y &= -\frac{1}{2}\lambda x^2.\end{aligned}$$

Cooling

Heat conduction

## 18. Numerical methods for higher order ODEs

When it comes to solving ordinary differential equations numerically, those of higher order differ from first order ODEs in one crucial aspect: They do not provide us with the slope of the solution, so the idea of going by segments cannot be used. We have to try something different.

### 18a. The Taylor method

Hookev zakon:  $F = kx$ , kde  $x$  je vchylka z pirozen dlky pruiny a  $k$  je tuhost pruiny. Ale jde v opanm smru, proto  $a = -\frac{k}{m}x$ .

a) Verze bez ten a vnj sly:  $\ddot{x} + \frac{k}{m}x = 0$  neboli  $\ddot{x} + \omega^2x = 0$ , kde jsme oznaili  $\omega = \sqrt{\frac{k}{m}}$ . Mimochoodem, rozmrov analiza dv pro omegu sekundy.

een je  $a \cos(\omega t) + b \sin(\omega t)$ . Pokud bereme  $t_0 = 0$  (co je tradin), pak  $a = x(0)$ ,  $b = \frac{\dot{x}(0)}{\omega}$ .

Finta:  $A \sin(\omega t - \varphi) = A \sin(\omega t) \cos(\varphi) - A \cos(\omega t) \sin(\varphi)$ . Porovnm dostvme rovnice

$$A \sin(\varphi) = -a, \quad A \cos(\varphi) = b,$$

ze kterch (vydlme, umocnme a seteme) dostvme

$$\operatorname{tg}(\varphi) = -\frac{a}{b}, \quad A = \sqrt{a^2 + b^2}.$$

hel  $\varphi = -\arctg\left(\frac{a}{b}\right)$  je nkdy teba upravit o  $\pi$ , aby dval sprvn znamnka:  $A \sin(\varphi) = -a$  a  $A \cos(\varphi) = b$ .

Podobn je tak mon vyjdit een jako  $A \cos(\omega t - \psi)$ , ale sinus je tradin.

Zvr: Pi kmitn na pruin bez ten je poloha urena funkc  $A \sin(\omega t - \varphi)$ , kde  $A = \sqrt{x(0)^2 + \dot{x}(0)^2}$  a  $\operatorname{tg}(\varphi) = -\omega \frac{x(0)}{\dot{x}(0)}$ . Perioda (doba jednoho plnho kmitu) je  $T = \frac{2\pi}{\omega} = 2\pi \sqrt{\frac{m}{k}}$ .

Velk  $k$  dv rychl kmity, to souhlas.

Omezen: Sla zvis na vchylce linern jen pro relativn mal  $x$ . Jakmile pekroim uritou mez, tak dojde k pokozen pruiny.

b) Verze se tenm: Fyzika k, e tec sla je dna jako  $dv(t)$ , kde  $v$  je okamit rychlost, a vede v opanm smru ne rychlost. Dostvme tedy po oznaen  $\omega_0 = \sqrt{\frac{k}{m}}$  rovnici

$$\ddot{x} + d\dot{x} + \omega_0^2x = 0.$$

Je to tzv. tlumen oscilace (damped oscillations), slo  $d$  je koeficient tlumen (damping coefficient) a  $\omega_0$  je zkladn frekvence (base frequency).

een jsou dna vlastnm sly  $\lambda = -\frac{d}{2} \pm \sqrt{\frac{d^2}{4} - \omega_0^2}$ . Ppady:

- $d < 2\omega_0$  znamen "underdamped" ppad, kdy je tlumen relativn mal. Vychzej komplexn koeny, po oznaen  $\omega = \sqrt{\frac{d^2}{4} - \omega_0^2}$  mme obecn een

$$x(t) = e^{-(d/2)t} [a \cos(\omega t) + b \sin(\omega t)].$$

Zase to lze pepsat jako  $x(t) = A e^{-(d/2)t} \sin(\omega t - \varphi)$ .

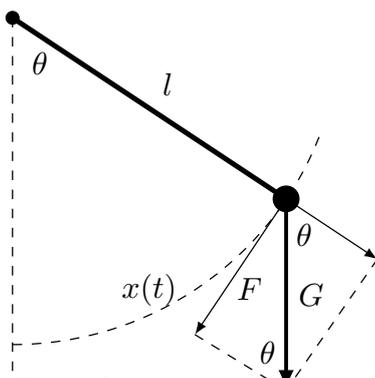
- $d = 2\omega_0$  je kritick tlumen s jednm dvojnsobnm (a zpornm) koenem  $\lambda$ , vychz

$$x(t) = (at + b)e^{-\lambda t}.$$

- $d > 2\omega_0$  znamen "overdamped" ppad, kdy je tlumen relativn velk. Vychzej zporn koeny  $\lambda_{1,2}$  a

$$x(t) = a e^{-\lambda_1 t} + b e^{-\lambda_2 t}.$$

### 18b. Matematické kyvadlo



Zva o hmotnosti  $m$  na nehmotnm a neprunm zvsu dlky  $l$  se pohybuje po krunici. Vzdlenost od doln ro

a) Verze bez ten a vnj sly:  $\ddot{\theta} + \frac{g}{l} \sin(\theta) = 0$ . Ozname  $\omega = \sqrt{\frac{g}{l}}$ , dostvme  $\ddot{\theta} + \omega^2 \sin(\theta) = 0$ .

Bohuel, tuto rovnici nelze analyticky vyeit. Vyaduje to tzv. eliptick integrly.

Speciln ppad: Pokud je  $\theta$  stle mal (prakticky vzato  $|\theta| < 0.5$ ), pak lze pout aproximaci  $\sin(\theta) \sim \theta$ .

Dostvme rovnici  $\ddot{\theta} + \omega^2 \theta = 0$  s eenm  $a \cos(\omega t) + b \sin(\omega t)$ . Pokud bereme  $t_0 = 0$  (co je tradin), pak  $a = x(0)$ ,  $b = \frac{\dot{x}(0)}{\omega}$ .

Zase funguje ten pepis  $x(t) = A \sin(\omega t - \varphi)$ , kde  $A = \sqrt{x(0)^2 + \dot{x}(0)^2}$  a  $\text{tg}(\varphi) = -\omega \frac{x(0)}{\dot{x}(0)}$ . Perioda je  $T = \frac{2\pi}{\omega} = 2\pi \sqrt{\frac{l}{g}}$ . m del kyvadlo, tm del perioda.

b) Verze se tenm: Viz pruina, analza obdobn, vznikaj tlumen kmity.

U pruiny i kyvadla je mon uvaovat nehomogenn rovnici v ppad tlumench kmit, pak mme nucen kmitn.

### 18c. Kyvadlo

U matematickho kyvadla se pedpokld zvs s nulovou hmotnost. Pokud uvaujeme zvs skuten, pak je teba brt v vahu jeho hmotnost a pracovat s momentem hybnosti msto zrychlenm.

Pokud m rovn stejnomrn ty o hmotnosti  $m_l$  a dlce  $l$  zvs ve vzdlenosti  $r$  od stedu, tak m moment hybnosti  $I = \frac{m_l l^2}{12} + m_l r^2$  a porovnn hybnost dv  $I\ddot{\theta} + rmg \sin(\theta) = 0$ .

Kdy seteme relnou ty uchycenou na konci a zva, dostaneme

$$(I + \frac{1}{2}ml^2)\ddot{\theta} + \frac{1}{2}mlg \sin(\theta) = 0.$$

Take stejn rovnice, jen jin konstanty.

### 18d. Pružné kyvadlo

Uvaujme zva o hmotnosti  $m$  zaven na pruin s tuhost  $k$  a klidovou dlkou  $l$ , jej hmotnost budeme ignorovat, stejn jako odpor vzduchu, odpor v zvsu a podobn. Zavedeme polrn souadnice  $(r, \theta)$  pro sted zva, potek je pirozen v zvsu. Jak rovnice dostaneme? Obvykle se vol pstup pes Lagrangin  $L = T - V$ , kde  $T$  je kinetick energie a  $V$  potencionln. U naeho kyvadla, kter se jednak kv a druhak kmit, bude teba kombinovat zdroje. U kinetick energie rozlome pohyb na sloku radiln a sloku tangenciln, ke kterm mme pm pstup ze souadnic. U potenciln energie bude u gravitan nutno spotat vku nad referenn rovinou a najt vzorec pro energii z pruiny.

$$T = \frac{1}{2}m\dot{r}^2 + \frac{1}{2}m(r\dot{\theta})^2,$$

$$V = -mgr \cos(\theta) + \frac{1}{2}k(r - l)^2,$$

$$\implies L = \frac{1}{2}m\dot{r}^2 + \frac{1}{2}m(r\dot{\theta})^2 + mgr \cos(\theta) - \frac{1}{2}k(r - l)^2.$$

Zkladn princip k, e proda vol takovou akci, kter minimalizuje celkovou spotebu energie. To vede na nulovost derivace odpovdajcho opertoru a rovnice  $\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{s}} \right) - \frac{\partial L}{\partial s} = 0$  vi vem souadnicm  $s$ , v

naem ppad  $r$  a  $\theta$ . Vylezou z toho rovnice

$$\begin{aligned}\ddot{r} + (\omega_r^2 - \dot{\theta}^2)r &= g \cos(\theta) + l\omega_r^2, \\ \ddot{\theta} + \frac{2}{r}\dot{r} \cdot \dot{\theta} + \frac{g}{r} \sin(\theta) &= 0,\end{aligned}$$

kde  $\omega_r = \sqrt{\frac{k}{m}}$ . Vidme, e prvni rovnice je netlumen, nucen kmitn s periodou zviseje na okamit vchylce kyvu, zatmco druh rovnice popisuje tlumen nenucen kvn, kde tlumen a frekvence zvisej pro zmnu na okamit situaci kmitn.

Existuj dal pepisy, napklad je mon zavst efektivn dlku kyvadla  $l_g = l + \frac{mg}{k}$ , co je klidov dlka kyvadla se zavenm zvam, dle  $\omega_\theta = \sqrt{\frac{g}{l_g}}$ , omezit se na mal kmity, kdy  $\cos(\theta) \sim 1$  a  $\sin(\theta) \sim \theta$ , nae rovnice pejdou v

$$\begin{aligned}\ddot{r} + \omega_r^2 r &= l_g(\omega_r + \dot{\theta}^2 - \frac{1}{2}\omega_\theta^2\theta^2), \\ \ddot{\theta} + \frac{2}{l_g}\dot{r} \cdot \dot{\theta} + \omega_\theta^2 \left(2 - \frac{r}{l_g}\right)\theta &= 0.\end{aligned}$$

K rezonanci dojde piblin v situaci, kdy  $\omega_r = 2\omega_\theta$ , neboli pro  $m \sim \frac{lk}{3g}$ . To mimochodem znamena, e  $l_g = \frac{4}{3}l$ , tedy experimentln prost vybereme takov zva, kter po zaven prodlou pruinu o tetinu.

Zajmav alternativa je pout kartzsk souadnice, kdy po oznaen  $\lambda = \frac{\omega_r^2 - \omega_\theta^2}{l_g}$  vychzej rovnice

$$\begin{aligned}\ddot{x} + (\lambda y + \omega_\theta^2)x &= 0, \\ \ddot{y} + \omega_r^2 y &= -\frac{1}{2}\lambda x^2.\end{aligned}$$

Cooling

Heat conduction

## 19. Finding roots numerically

Solving equations is a popular task in mathematics. The first type that one meets is algebraic equations. We start by learning how to solve linear equations of the form  $ax + b = c$ , and by the time we go to high school, we also master the art of solving quadratic equations. Remarkably, there is no more progress afterwards. We do learn some tricks for solving equations with powers  $a^k$  and logarithms, but the problems that we applied them to were carefully prepared, just a small change would put them beyond our means. Surprisingly, when it comes to reliably solving various types of equations, we did not progress beyond elementary school knowledge. Perhaps we should clarify that by solving an equation we mean writing its solution using an algebraic formula.

The reason for this lack of progress is simple: algebraic equations are tough. Speaking of polynomials, there are formulas for roots of polynomials of degree three and four, but they are rather unpleasant. Worse, mathematicians proved that it is impossible to have formulas for roots for polynomials of degree five and more. Once we move away from polynomials, things become even worse and even simple equations can defeat us.

Consider the equation  $\cos(x) = x$ . Just imagining the graphs of  $\cos(x)$  and  $y = x$  we readily conclude that they intersect, and this equation therefore has some solution. However, we do not have any tool how to get to it in an algebraic manner. In fact, it can be proved that this solution cannot be expressed by an algebraic formula. Consequently, no matter what algebraic tricks we try, we will never reach it. The same story is true for the equation  $e^{-x} = x$ , which also looks quite simple.

On the other hand, equations play an important role in applications, so we do have to be able to somehow identify their solutions. We therefore turn to numerical mathematics. In order to simplify our work, we start by unifying the setting. Every algebraic equation can be easily rearranged by putting all terms to the left-hand side of the equation, leaving just zero on the right. In other words, it is enough to know how to solve numerically equations of the form  $f(x) = 0$ . For solutions of such equations we have a special name.

### Definition 19.1.

By a **root** of a function  $f$  we mean any number  $r$  such that  $f(r) = 0$ .

Some people also say **zero of a function** instead.

How do we find roots of functions numerically? There are several problems one faces. To start with, we are not even able to recognize whether we found a root or not. When a computer tells us that  $f(x) = 0$  for some  $x$ , then it simply means that the value is smaller than the machine epsilon (see discussion in section 3c). This is useful to know, but it is not such a big problem in this context, because we do not expect to find the actual root anyway.

As is usual in numerical analysis, our aim is to provide some approximation of the root whose distance from the actual root is smaller than some prescribed tolerance  $\varepsilon > 0$ , which means that we want to control the absolute error  $E_r = r - \hat{r}$  of our approximation  $\hat{r}$  of  $r$ . In fact, one would expect that we focus on relative error, as it relates to reliability of our approximation, but it is customary to work with absolute error here. Typically we want  $|E_r| < \varepsilon$ , but asking for  $|E_r| \leq \varepsilon$  is also possible and there is just a formal difference between the two. This will be the basic setting of this and the following chapters.

It is useful to realize that the functions that we encounter in numerical mathematics need not be given by formulas like those that we meet at school. It is possible that the value has to be measured, in particular it may be an outcome of an experiment. So when we say “substitute a number  $a$  into a function  $f$ ”, it may also mean that we start an experiment with some setting set to  $a$  and then its outcome determines  $f(a)$ . If  $f(a)$  denotes the mass of tomatoes harvested from a hothouse when the temperature is set to  $a$ , then a simple evaluation can last a full season. This

is a rather extreme example, but it is good to remember that evaluating a function may not be so easy as we are used to.

We start with a simple question. Given some function  $f$ , how do we recognize that it actually does have some root? Even this question we cannot answer to our full satisfaction. There is no procedure that would reliably recognize the existence of a root. But there are some approaches that work reasonably well most of the time, which is a familiar situation in numerical analysis.

So what are the approaches? In the best case we have some information about the shape of the given function. For instance, we may be able to plot its graph. However, this is not reliable. Plotting a function essentially means that we sample it at specific points and then create a curve based on these values. However, imagine a function that is seemingly always positive, but has one really narrow spike reaching down below the  $x$ -axis. Unless we are lucky and sample this function in the right place, our plots will never reveal the existence of this spike and we will think that it has no root. Still, if we are able to plot our function, then this is the best start.

Sometimes the function is given by a formula that can be investigated using tools from calculus. Or the function may come from some real-life application and the story that accompanies it gives reason to believe in existence of such a root.

And when all fails, we can simply start sampling the function, that is, start substituting numbers into it (perhaps randomly) and see what happens. What we want to see is change in signs of the values as we go from one place to another. A layperson would then conclude that there must be a root somewhere inbetween, but those who took calculus know that in the world of mathematics there are also weird functions, and that we have to add a crucial assumption to be able to come to this conclusion.

**Theorem 19.2.**

Let  $f$  be a function continuous on some interval  $[a, b]$ . If the numbers  $f(a)$ ,  $f(b)$  have opposite signs, then there must be a root of  $f$  in  $(a, b)$ .

The assumption on signs can be efficiently expressed like this:  $f(a) \cdot f(b) < 0$ . The theorem itself is just an easy consequence of the classical Intermediate value theorem from calculus.

Unfortunately, this theorem is far from perfect. When it happens that the two signs are equal, then we simply cannot make any conclusion, as there may or may not be a root between  $a$  and  $b$ . Typically we would try to substitute more numbers, but unfortunately there are roots that simply cannot be identified in this way. Imagine the classical parabola, the graph of  $f(x) = x^2$ . It has a root at the origin, but no matter how hard we try, we never achieve opposite signs when sampling. This shape is typical of roots of even multiplicity. This is the first but not the last time that we see roots of higher multiplicity causing troubles.

However, in many cases this test works fine, and it is often the best tool that we have available.

**Example 19.a:** Consider the function  $f(x) = x^3 - x - 10$ . Does it have a root somewhere?

We try some numbers. My favorite is zero:

$$f(0) = -10.$$

We do not actually care about the exact value, just that it is negative. Now we would like to see some positive value. Since we know how the powers behave, we feel that some larger number should do. Say,

$$f(5) = 110.$$

Yes, there must be some root between 0 and 5.

Note that we could deduce the existence of the root also in other ways. For instance, we know that  $\lim_{x \rightarrow \infty} (f(x)) = \infty$  and  $\lim_{x \rightarrow -\infty} (f(x)) = -\infty$ , so as a continuous function it must have a root

somewhere.

△

When we identify a position of some root within a (finite) interval, we say that we **bracketed** it. However, this is not enough, we want to approximate it with a given precision.

The general approach is to try some initial guess, call it  $x_0$ . If it is not good enough, we try to make a better guess, call it  $x_1$ , using information gained from  $x_0$ . Of course, it is a sophisticated guessing. We continue in this way, constructing numbers  $x_k$ , until we find one whose absolute error as approximation for the desired root is within the specified tolerance.

Since different customers can have different tolerances, we are interested in procedures that are capable of producing arbitrarily good approximations, in other words, we should be able to let them run as long as needed, until they eventually succeed.

The general setting thus is that we will be interested in **iterative methods** that create (potentially infinite) sequences of numbers  $x_k$ . A good process (method) should be able to provide approximations for some root of arbitrary precision, which essentially means that the sequences such a method produces should converge (in the usual mathematical sense) to the desired roots.

There are many iterative methods for finding roots, and they can be grouped according to their nature. We will explore here two major approaches by introducing a representative method for each.

### 19a. The bisection method (bracketing)

We have the following situation: for a given (continuous) function  $f$  we found points  $a, b$  such that values  $f$  have opposite signs there. We know that this tells us that there must be some root in the interval  $(a, b)$ . Can we localize it closer?

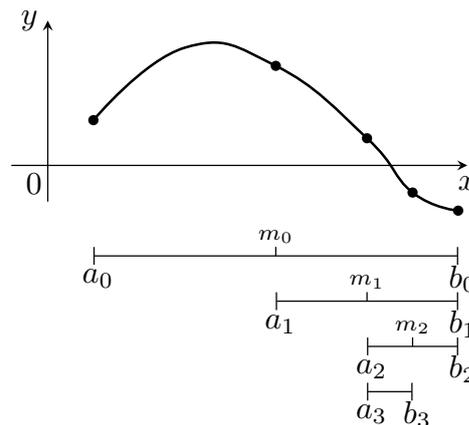
Here is a natural idea: We take the middle of this interval, call it  $m$ , and substitute into  $f$ . If the result is zero, then this  $m = \frac{1}{2}(a + b)$  is very likely a root (see the remark above about computer computing). However, the chances are low. It is more likely that  $f(m)$  is not zero, so it must be either positive or negative. Since the numbers  $f(a), f(b)$  have different signs,  $f(m)$  cannot obviously have the same sign as both of them. Consequently, either the pair  $f(a), f(m)$  has opposite signs, then there is a root in the interval  $(a, m)$ , or the pair  $f(m), f(b)$  has opposite signs and we have a root in  $(m, b)$ . Either way, we localized the root in an interval that is smaller than the original interval  $(a, b)$ , so we narrowed it down.

This looks like a good idea, so we continue, splitting the new interval again and choosing the half with opposite signs. To keep track, we introduce names. The starting interval will be called  $[a_0, b_0]$  with midpoint  $m_0$ , the next one  $[a_1, b_1]$  with midpoint  $m_1$ , and eventually we construct a (theoretically infinite) sequence of intervals  $[a_k, b_k]$  with midpoints  $m_k$ , each has half the length of the previous one, and we always have some root in every  $[a_k, b_k]$ , because we have opposite signs of  $f$  at the endpoints.

We wanted to focus on iterative procedures that produce sequences of approximations. To fit within this category, our procedure should produce some official root approximation at each stage  $[a_k, b_k]$ . We know that the root is somewhere within this interval, so the natural output for this stage would be the midpoint  $m_k$ .

When do we stop? If we know that there is a root in an interval  $[a_k, b_k]$  and we declare its midpoint  $m_k$  as the official output, then the largest possible error (absolute error) is  $\frac{1}{2}|b_k - a_k|$ . So we simply wait for this number to drop below the prescribed tolerance  $\varepsilon$ .

We just developed one of the basic root finding methods.



**Algorithm 19a.1.**

⟨bisection method for finding root of  $f$ ⟩

Given: a function  $f$  continuous on interval  $[a, b]$  and a tolerance  $\varepsilon$ .

Assumption:  $f(a)$  and  $f(b)$  have opposite signs.

0. Set  $a_0 = a$ ,  $b_0 = b$ . Let  $k = 0$ .

1. Assumption:  $f(a_k)$  and  $f(b_k)$  have opposite signs.

Let  $m_k = \frac{1}{2}(a_k + b_k)$ .

2. If  $f(m_k) = 0$  or  $\frac{1}{2}|b_k - a_k| < \varepsilon$  then the algorithm stops, output is  $m_k$ .

Otherwise:

If  $f(a_k)$  and  $f(m_k)$  have opposite signs, set  $a_{k+1} = a_k$ ,  $b_{k+1} = m_k$ ,  
increase  $k$  by one and go back to step 1.

If  $f(m_k)$  and  $f(b_k)$  have opposite signs, set  $a_{k+1} = m_k$ ,  $b_{k+1} = b_k$ ,  
increase  $k$  by one and go back to step 1.

△

**Example 19a.a:** Consider the function  $f(x) = x^3 - x - 10$ . Above we identified the interval  $[0, 5]$  where some root lies. We start the bisection method:

$a_0 = 0$ ,  $b_0 = 5$ , we recall that  $f(0) < 0$  and  $f(5) > 0$ .

We find the midpoint  $m_0 = 2.5$  and substitute into the function:  $f(2.5) = 3.125$ , this is positive.

We did not find the root, so we pass to a smaller interval. Since the sign of  $f$  changes between 0 and 2.5, the next iteration should be  $a_1 = 0$ ,  $b_1 = 2.5$  and the output is  $m_1 = 1.25$ .

We substitute the midpoint  $m$  into the function:  $f(1.25) = -9.296875$ , this is negative.

We did not find the root, so we pass to a smaller interval. Since the sign of  $f$  changes between 1.25 and 2.5, our next iteration will be  $a_2 = 1.25$ ,  $b_2 = 2.5$ . If we decided to stop here, the official output would be  $m_2 = 1.875$  and there is definitely some root within the distance  $0.625 = \frac{1}{2}(2.5 - 1.25)$ .

△

What are the main properties of this method?

- The bisection method is reliable.
- We have a good control over the error.
- The bisection method is slow.

We will now address these properties, and start with the second one.

Consider a certain stage  $k$  in the algorithm. There must be some root in  $[a_k, b_k]$ , we denote it  $r$ . We already observed that the official output  $m_k$  has its absolute error  $|E_k| = |r - m_k|$  at most  $\frac{1}{2}|b_k - a_k|$ . This is a relatively rare occasion in numerical analysis, it is a big bonus to have an upper bound for the error that uses data that we actually have available.

There is a group of root-finding methods that are called **bracketing methods**. They all work with nested intervals that keep the root inside, and the control over the position of the root is their major advantage.

Having established the control over the error, we move on. For simplicity, we denote  $e_k = \frac{1}{2}(b_k - a_k)$ , this is our upper estimate for  $|E_k|$ . From the way the intervals are constructed we easily see that  $e_k = \frac{1}{2}e_{k-1}$ , this recursive equality can be applied repeatedly and we obtain

$$|E_k| \leq e_k = \frac{1}{2}e_{k-1} = \frac{1}{2} \frac{1}{2}e_{k-2} = \cdots = \frac{1}{2^k}e_0 = \frac{1}{2^{k+1}}(b_0 - a_0).$$

This shows that  $E_k \rightarrow 0$ , so the bisection method is capable of producing root approximations of arbitrary precision. In other words, the sequence  $\{m_k\}$  produced by the bisection method always converges to a root.

This brings us to reliability. Can anything go wrong with this process? And the answer is in the negative. Given some tolerance  $\varepsilon > 0$  and an initial interval  $[a, b]$  on which the given function changes sign, the bisection method necessarily leads us to suitable approximation.

Could there be numerical troubles? There is only one calculation that is a part of the bisection method itself, namely the calculation of the midpoint. Surprisingly enough, we can run into trouble there. If the numbers  $a_k, b_k$  get really close, the formula  $\frac{1}{2}(a_k + b_k)$  can actually produce a number that is outside of the interval  $[a_k, b_k]$ !

For instance, consider the interval  $[5.3, 5.4]$  and imagine that we work with a two-digit precision. Then  $5.3 + 5.4 = 10.7$  gets rounded up to 11, so we get the midpoint  $m = 5.5$ . This would be fatal, which explains why in implementation the formula  $m_k = a_k + \frac{1}{2}(b_k - a_k)$  is preferred, because it is not susceptible to such problems.

Having fixed this, there is nothing else that can go wrong with the bisection method. Of course, we may run into numerical problems when evaluating  $f$ , but that is another story, and if that happens, then it will trouble us regardless of what method we use.

In conclusion, apart from possible problems with  $f$ , this method is absolutely reliable.

It remains to address its speed. Above we tried to approximate a root of  $x^3 - x - 10$ . How many iterations would it take before our error approximation drops below a prescribed tolerance? Our requirement is that  $|E_k| < \varepsilon$ , but we do not have access to the errors. However, we have an upper bound, so we ask for  $e_k < \varepsilon$  instead. The inequality  $|E_k| < e_k$  shows that we then have  $|E_k| < \varepsilon$  automatically.

When will  $e_k < \varepsilon$  become true? We can substitute for  $e_k$  the formula that we deduced above to obtain  $\frac{1}{2^k}e_0 < \varepsilon$ . We can solve this inequality for  $k$ :  $k > \log_2\left(\frac{e_0}{\varepsilon}\right)$ . Substituting for  $e_0$  and passing to a slightly large  $k$  to simplify the formula we see that  $N = \left\lceil \log_2\left(\frac{|b_0 - a_0|}{\varepsilon}\right) \right\rceil$  iterations should do.

This is another advantage of the bisection method: It is predictable. Note that the formula makes sense: The smaller the tolerance, the larger the number of steps. Which brings us to the question of speed: Is this  $N$  reasonable or is it too much?

We will shortly develop a general theory for comparing speeds of root-finding methods (section 19d), for now we will do with intuitive reasoning. The natural approach is to ask how the precision of our approximation improves with each step. In fact, we have an answer already, the formula  $e_{k+1} = \frac{1}{2}e_k$  tells us that the (upper estimate for) the error gets halved at each iteration. To get a feeling for this result we switch our point of view.

Intuitively, a common measure of a number's reliability is how many valid digits are determined right. We already commented before that the notion of "right digits" is somewhat fuzzy, but it is a useful aid for our intuition. We related this to the relative error, in particular we observed that, roughly speaking, improving accuracy by one digit is related to decreasing its relative error by a factor of ten. How much work does it take if we use the bisection method?

The relation  $e_{k+1} = \frac{1}{2}e_k$  seems to lead in the right direction. The decrease by 2 is not enough to gain a new valid digit, but more iterations may help:

$$\begin{aligned} e_{k+3} &= \frac{1}{2^3}e_k = \frac{1}{8}e_k, \\ e_{k+4} &= \frac{1}{2^4}e_k = \frac{1}{16}e_k. \end{aligned}$$

We can see that the desired improvement by the factor 10 is expected to happen somewhere between the third and the fourth iteration. One may argue that we worked with the absolute error here, but just dividing these inequalities by the actual error  $r$  yields exactly the same relationships for the relative errors, starting with the basic formula

$$\frac{e_{k+1}}{r} = \frac{1}{2} \frac{e_k}{r}.$$

This supports mathematically that when comparing errors of iterations, it is enough to focus on absolute errors and the conclusions will be valid also for relative errors.

We will now test this observation. Before we make a run of the bisection method, let us make a more practical (and less mathematical) observation. Once the absolute errors drop below 1, we can judge the precision of our approximation by the number of leading zeros in the absolute error. A new leading zero means a new valid digit gained. Again, this relationship is not completely precise, but as a rough guide it will do, and it allows us to judge how things go easily by looking at the outputs.

**Example 19a.b:** We return to  $f(x) = x^3 - x - 10$ , considered on the interval  $[0, 5]$ . We applied the bisection method, with tolerance  $\varepsilon = 0.0001 = 10^{-4}$ . In the following chart we list the intervals  $[a_k, b_k]$ , midpoints  $m_k$  and test values  $e_k$ :

$k$	$a_k$	$b_k$	$m_k$	$e_k$
0	0.00000	5.00000	2.50000	2.50000
1	0.00000	2.50000	1.25000	1.25000
2	1.25000	2.50000	1.87500	0.62500
3	1.87500	2.50000	2.18750	0.31250
4	2.18750	2.50000	2.34375	0.15625
5	2.18750	2.34375	2.26562	0.07812
6	2.26562	2.34375	2.30469	0.03906
7	2.30469	2.34375	2.32422	0.01953
8	2.30469	2.32422	2.31445	0.00977
9	2.30469	2.31445	2.30957	0.00488
10	2.30469	2.30957	2.30713	0.00244
11	2.30713	2.30957	2.30835	0.00122
12	2.30835	2.30957	2.30896	0.00061
13	2.30835	2.30896	2.30865	0.00031
14	2.30865	2.30896	2.30881	0.00015
15	2.30881	2.30896	2.30888	0.00008

The first three lines confirm that our previous attempts done by hand agree with the computer run. But now we focus on the last column, namely how many iterations it takes for another zero to appear after the decimal dot in  $e_k$ . We can see that we need 3, 4, and 3 iterations. Experiments with other tolerances and other functions and intervals would confirm that the pattern 4-3-3 is typical for the bisection method. If true, it would mean that we can expect 10 more iterations if we ask for improvement in precision by three digits.

As we see above, achieving the tolerance  $\varepsilon = 10^{-4}$  required  $N = 15$  iterations. When we change the tolerance to  $\varepsilon = 10^{-7}$ , the algorithm stops after  $N = 25$  iterations. This is a very good fit with our expectations.

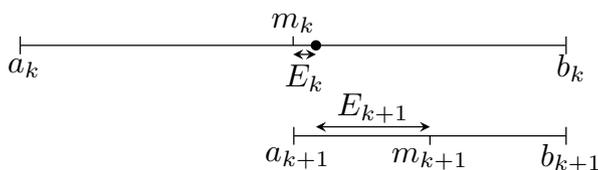
△

The last observation is interesting, What can we expect after ten iterations in theory?

$$e_{k+10} = \frac{1}{2^{10}} e_k = \frac{1}{1024} e_k \approx \frac{1}{1000} e_k.$$

Since 1024 and 1000 are close, we did not make a large error in the last step, so it indeed looks like ten iterations per three new digits is supported also by theory. Consequently, on average, three and one third of iteration are needed on average to gain a new valid digit.

Unfortunately, this conclusion would be true if we established those estimates for actual errors  $E_k$ , but we did it for the upper estimates  $e_k$ . The actual errors can behave in a different way, they may even increase from stage to stage. The following picture shows how this is possible. The actual root is marked with a dot.



Indeed, we can return to the last experiment above to observe this happening.

**Example 19a.c:** In example 19a.b we applied the bisection method to the function  $f(x) = x^3 - x - 10$  on the interval  $[0, 5]$ . We only listed the estimates  $e_k$  for the error, but we actually do know the actual root:

$$r = \frac{1}{3} \sqrt[3]{135 + \sqrt{2022}} + \frac{1}{\sqrt[3]{135 + \sqrt{2022}}}.$$

By the way, we commented at the beginning of this chapter that formulas for roots of polynomials of the third degree are not exactly friendly. Now you can see what we meant by this.

Anyway, knowing the actual root we can in fact look at the actual errors of our approximations:

$k$	$m_k$	$e_k$	$E_k$
0	2.50000	2.50000	0.19109
1	1.25000	1.25000	1.05890
2	1.87500	0.62500	0.43390
3	2.18750	0.31250	0.12140
4	2.34375	0.15625	0.03484
5	2.26562	0.07812	0.04328
6	2.30469	0.03906	0.00421
7	2.32422	0.01953	0.01531
8	2.31445	0.00977	0.00554
9	2.30957	0.00488	0.00066
10	2.30713	0.00244	0.00177
11	2.30835	0.00122	0.00055
12	2.30896	0.00061	0.00005
13	2.30865	0.00031	0.00025
14	2.30881	0.00015	0.00009
15	2.30888	0.00008	0.00002

As expected, the actual errors never exceed the upper estimates, so things are as they should. We also see repeatedly that the actual errors increased at some iterations. On the other hand, when we ignore small details and look at the overall pictures, we do see zeros appearing in the errors with roughly the same shape as in the estimates  $e_k$ , so the general trend seems to fit.

△

This is a general experience. The estimates  $e_k$  form an envelope under which the actual errors must stay, in particular they do have to get smaller at a rate that is, on average, about the same as the theoretical one. In other words, we can really expect about ten iterations for three new valid digits.

This may not sound bad, but in fact it is rather slow when it comes to problems involving functions that are not so easy to evaluate. We will see soon that much faster methods are available.

This concludes this section. We introduced a method whose strength is its simplicity, reliability and error control (a feature common to all bracketing methods), but we pay for it by low speed. One reason for this is the fact that we actually never use in our decision making any information about the actual shape of the function. We just care about signs at points that are determined by the initial interval  $[a_0, b_0]$ . If we want a faster algorithm, we have to look at what the function is doing.

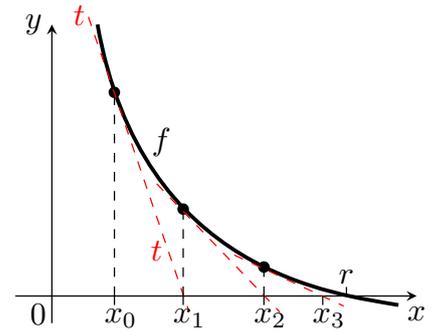
## 19b. The Newton method

We have a function  $f$  and we want to approximate its root that hopefully exists.

One possibility is to simply try some  $x_0$ . We substitute it into our function  $f$ . Unless we are extremely lucky (in which case we are most likely in Las Vegas instead of reading a book on ODEs),  $f(x_0)$  is not zero. What next? We would like to get a better approximation using our knowledge of the function.

If our function is differentiable, then we can find the tangent line to it at our point  $x_0$ . With a bit of luck this tangent line intersects the  $x$ -axis and the picture suggests that this intersection, call it  $x_1$ , could be a better approximation for the root. This looks like a good move, so we try it again:

We move by the tangent line at  $x_1$  to a still better approximation  $x_2$ . And again the same move, getting  $x_3$ . This seems to work, we are getting close.



How will this work mathematically? We will deduce a general formula. We start at a point  $x_k$ , and we know  $f(x_k)$  and  $f'(x_k)$ . First we construct the tangent line to the graph of  $f$  at  $x_k$ . It must pass through the point  $[x_k, f(x_k)]$  and it has the slope  $k = f'(x_k)$ , so we easily write its equation:

$$t: y = f(x_k) + f'(x_k) \cdot (x - x_k).$$

Where does it intersect the  $x$ -axis? We put 0 for  $y$  and solve the resulting equation for  $x$  to find out:

$$\begin{aligned} -f(x_k) &= f'(x_k) \cdot (x - x_k) \implies -\frac{f(x_k)}{f'(x_k)} = x - x_k \\ \implies x &= x_k - \frac{f(x_k)}{f'(x_k)}. \end{aligned}$$

This will be our new  $x_{k+1}$ . So we have a recursive formula for our procedure and we can write the algorithm. We also need to know when to stop, so we put a test in the algorithm, but we will not worry about it now. This is a very loaded question and we will return to it later, see section 19c.

### Algorithm 19b.1.

(Newton method for finding root of a function  $f$ )

Given: a function  $f$  continuous on  $\mathbb{R}$  and a tolerance  $\varepsilon$ .

0. Choose  $x_0$ . Let  $k = 0$ .

1. Let  $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$ .

If  $|x_{k+1} - x_k| < \varepsilon$  or  $|f(x_{k+1})| < \varepsilon$  then the algorithm stops, output is  $x_{k+1}$ .

Otherwise increase  $k$  by one and go back to step 1.

△

Some people call it the Newton-Raphson method, the natural name would be the tangent method.

**Example 19b.a:** Consider again our favorite test function  $f(x) = x^3 - x - 10$ . We will approximate its root using the Newton method, as the initial guess we take  $x_0 = 1$ .

First we find

$$f'(x) = 3x^2 - 1$$

and now we are ready.

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 1 - \frac{1^3 - 1 - 10}{3 \cdot 1^2 - 1} = 6.$$

The next step:

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 6 - \frac{6^3 - 6 - 10}{3 \cdot 6^2 - 6} = 6 - \frac{100}{51} = \frac{206}{51}.$$

Obviously, this is a job for a computer, but the idea should be clear now.

It is possible to set up a dedicated iterative formula when applying the Newton method to a specific problem. In our case it would read

$$x_{k+1} = x_k - \frac{x_k^3 - x_k - 10}{3x_k^2 - 1} = \frac{2x_k^3 + 10}{3x_k^2 - 1}.$$

Sometimes such a formula simplifies, making the actual calculations easier compared to the direct application of the Newton formula. One can even develop dedicated procedures for solving specific types of problems. We will return to this idea in section .

In example we asked how many iterations it took to find the desired root with certain tolerances, and we found  $N = 15$  for  $\varepsilon = 0.0001 = 10^{-4}$  and  $N = 25$  for  $\varepsilon = 10^{-7}$ .

We now start the procedure again, set tolerance to  $\varepsilon = 0.0001$ , but this time with the Newton method and the initial guess  $x_0 = 5$ . We obtain the following run.

$k$	$x_k$	$f(x_k)$	test
0	5.0000000	110.0000000	110.0000000
1	3.5135135	29.8600281	1.4864865
2	2.6848585	6.6688513	0.8286550
3	2.3615265	0.8082521	0.3233320
4	2.3101450	0.0185680	0.0513815
5	2.3089080	0.0000106	0.0012370
6	2.3089073	0.0000000	0.0000007

Now that was quite fast,  $N = 6$  iterations were enough. We will worry about the test column later, but the column with  $f(x_k)$  looks interesting, especially the last row. In fact, the computer thinks that we actually found the root at that step.

How many iterations are needed when we increase our demand by lowering the tolerance to  $10^{-7}$ ? Remarkably,  $N = 6$  is again the answer, because the algorithm thinks that  $x_6$  is also good for this new demand. Comparison with the bisection method (6 versus 25) is inevitable, although it is not completely fair as the two methods did not start from exactly the same starting block, one needing an interval, the other just an initial guess.

However, we do seem a rather clear message. Note that  $x_5$  was not judged good for the error 0.0001, while  $x_6$  is considered good even for 0.0000001. Now we are yet do discuss how the computer came to this conclusion, but if it is reasonably correct, then this would suggest that we gained four more valid digits in just one iteration, something that the bisection method can only dream of.

△

Recall that the main properties of the bisection method were as follows:

- The bisection method is reliable.
- We have a good control over the error.
- The bisection method is slow.

The main properties of the Newton method are as follows:

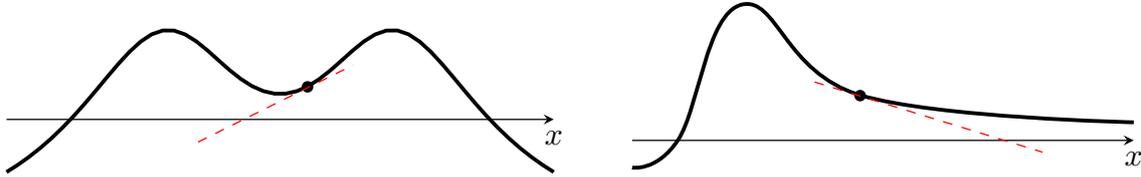
- The Newton method is unreliable.
- We have no control over the error.
- The Newton method is fast.

The third property sounds good, that is exactly what we wanted, but we paid for it dearly it seems. Each of the points deserves a thorough investigation.

The first is actually the simplest. For starters, just by looking at the formula we can see that we can get in trouble if  $f'(x_k) = 0$  for some  $x_k$ . Indeed, this would mean that our  $x_k$  happened to

catch our function at its local extreme, there the tangent line is horizontal and has no intersection with the  $x$ -axis. However, this is the least of our worries. For a typical function there is just a small number of local extrema, and we can easily avoid them by slight modification of our initial guess  $x_0$ , the resulting run will most likely miss the offending point.

A more serious problem lies in high sensitivity of the Newton method to the shape of the function. One can easily find shapes that totally throw this method off.



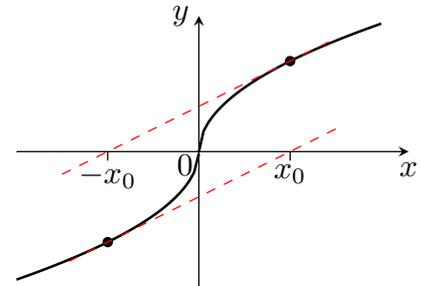
On the left we see one of my favorites, a camel. If our initial guess happens to land in the valley, then it can easily happen that tangent lines will be sending us left and right, never leaving the valley to reach the root on the side. The second shape is equally deadly. If our initial guess falls to the right of the hill, the tangent lines will chase us out to the right, all the way to infinity.

My personal favorite in this context is the function  $f(x) = \begin{cases} \sqrt{x}, & x \geq 0; \\ -\sqrt{|x|}, & x < 0. \end{cases}$  Simply put, this is an odd function based on the shape of the standard square root.

Now what happens if we start the Newton method at some  $x_0 > 0$ ? The formula says that

$$x_1 = x_0 - \frac{\sqrt{x_0}}{\frac{1}{2\sqrt{x_0}}} = -x_0.$$

So the tangent line sends us to a symmetric point. Since the graph is odd, that is, symmetric, the next step should send us back. And indeed, the calculation confirms that  $x_2 = x_0$ . The conclusion is obvious: Unless we start at the root itself, the iteration will keep jumping right and left.



Another problem with the Newton method is that it likes to wander. It may eventually get to the root, but first it can (and often does) take several detours and scenic rides.

**Example 19b.b:** We return to our favorite test function  $f(x) = x^3 - x - 10$ . When we used the Newton method to obtain an approximation of the root with the precision  $\varepsilon = 0.0001$  and the initial guess  $x_0 = 5$ , it took  $N = 6$  approximations and the run was straightforward.

However, when we change this initial guess to  $x_0 = 0$ , it will take fairly substantial  $N = 22$  iterations. We will not list them all, just the extremes.

$k$	$x_k$
0	0.0000000
1	10.0000000
2	-6.6555183
6	0.5299389
7	-65.3843761
17	4.1968304

Starting with  $x_{17}$ , the iterates became a decreasing sequence converging to the root.

Incidentally, while  $x_0 = 0.2$  leads to a run requiring  $N = 18$  iterations, the choice  $x_0 = 0.3$  can do with just  $N = 11$ .

△

Which brings us to another interesting feature of the Newton method: Extreme sensitivity to the initial guess.

**Example 19b.c:** This time we make an exception and consider the function

$$f(x) = x + \frac{1}{23}x^2 - \frac{1}{5}x^3 + \frac{31}{2 * (x - 5)^2 + 1}.$$

It has three roots:

$$r_1 = -1.938\dots$$

$$r_2 = -0.541\dots$$

$$r_3 = 5.370\dots$$

We will apply the Newton method with tolerance set to the traditional  $\varepsilon = 0.001$  with eleven closely spaced initial guesses. The chart shows to which root the resulting runs converge, and the number of iterations required before the algorithm terminated. In the last row we show some interesting values the run visited if the run was not straightforward.

$x_0$	4.0	4.1	4.2	4.3	4.4	4.5	4.6	4.7	4.8	4.9	5.0
$r$	$r_3$	$r_1$	$r_3$	$r_3$	$r_1$	$r_1$	$r_3$	$r_2$	$r_3$	$r_3$	$r_3$
$N$	11	19	9	9	9	17	16	8	7	7	5
	19.9	-49.8		15.5		-41.7				17.0	

The table speaks for itself. The run with  $x_0 = 4.6$  is especially intriguing. We did not list any extreme values of iterates, and yet we claim that it took 16 iterations. In fact, the iterates were oscillating between about 4 and about 5 for a while before settling on the right approximation.

△

The practical lesson here is that the Newton method is an algorithm that requires supervision. In particular, for all iterative methods it pays to implement some safety stops: The algorithm stops automatically after a certain number of iterations, and the algorithm stops when numbers involved get too large. When an algorithm stops prematurely due to conditions like these, the operator looks at the sequence produced by this algorithm and makes a judgement whether the algorithm is just slow, but a longer run would produce a satisfactory approximation, or something fishy is going on.

Still, at least pictures suggest that in some situations the Newton algorithm works reliably. Indeed, the right combination of monotonicity, concavity and initial guess can provide good convergence. Here is an example of a theorem that fairly popular in textbooks.

**Theorem 19b.2.**

Let  $f$  be a function on an interval  $[a, b]$  such that  $f(a) \cdot f(b) < 0$ . Assume that  $f$  is twice continuously differentiable on  $(a, b)$  and  $f' \neq 0$ ,  $f'' \neq 0$  on  $(a, b)$ .

If  $x_0 \in (a, b)$  is chosen so that  $f(x_0) \cdot f''(x_0) > 0$ , then the sequence  $\{x_n\}$  generated by the Newton method converges to a root  $r \in [a, b]$  of  $f$ .

Let's try to decipher it. We have a function that changes signs at endpoints, therefore there is a root in  $(a, b)$ . The fact that derivatives are not allowed to be zero means that both derivatives have to settle on one of the signs for the whole interval  $[a, b]$ . In other words, the function has to decide what kind of monotonicity and concavity it wants to have and stick with it on that interval. Note that this is not so restrictive. Functions do change their monotonicity and concavity, but usually only at a few places, and generally consist of large stretches where these assumptions are true. When we get really close to some root, then there is a good chance that the function behaves as needed, unless the root is actually a local extreme or an inflection point. In other words, roots of higher multiplicity cause trouble again.

Put in plain words, the Newton method may behave badly globally, but once it gets close to the root, there is a good chance that things will go well.

However, the second paragraph tells us that one has to be a bit lucky, too, as the initial guess has to fit well with the shape of the function. For instance, if it is concave up, then the initial guess should be positive.

This theorem is in fact not the most practical statement, in particular it may be faster to simply try a run of Newton's method than verifying all these assumptions. The main message is that it really makes most sense to use the Newton method when we are already close to a root, which is a strategy that we will return to later.

Now we address the other two properties. Both of them are in fact rather important, raise some general questions and lead to general notions. We therefore dedicate special sections to them.

### 19c. Stopping conditions

Here is the situation: We have a function  $f$ , we used the Newton method to estimate its root, and after a while we arrived at an approximation  $x_k$ . Should we stop or continue? We should stop when the distance between the approximation  $x_k$  and the root  $r$  is smaller than the given tolerance  $\varepsilon > 0$ , but unfortunately, this is exactly the kind of information that we do not have. From knowing  $f$  and  $f'$  at  $x_k$  (or the previous points) we cannot reliably estimate the distance from the root.

This is not just a problem of the Newton method. If we use a method that is not bracketing, then we only have information about the function  $f$  at points  $x_k, x_{k-1}, \dots$ , and there is no mathematical tool that would reliably derive the desired distance from the root based on this.

So how do people stop their algorithms? There are three popular **stopping conditions**. Assume that an iterative algorithm produces a sequence  $\{x_k\}$ . We can use one or more of the following to stop it:

- $|x_k - x_{k-1}| < \varepsilon$  (absolute difference)
- $\frac{|x_k - x_{k-1}|}{|x_k|} < \varepsilon$  (relative difference)
- $|f(x_k)| < \varepsilon$  (value difference)

The epsilon in the inequalities is typically the supplied tolerance  $\varepsilon$ , but it may be also another value, as we may want to modify the given precision to achieve a better run.

In some books, the first and the second condition are called the absolute error and the relative error. However, this is highly misleading, as the two conditions have nothing in common with the error of the approximation  $x_k$ . So why do we use them? Wishful thinking.

The absolute difference test is based on the idea that if the successive approximations almost do not change, then we are probably close to the root itself. Unfortunately, there is no good reason why this should be true, and it is easy to find examples of functions for which this fails spectacularly. Still, this is probably the most popular stopping condition, and we will see some indications that it can have something in common with the actual error.

For one such indication we return to the bisection method. We stopped the run based on the test  $\frac{1}{2}|b_k - a_k| < \varepsilon$ . However, note that this is exactly the distance between the official output  $m_k$  (the midpoint) and the endpoints of this interval. Now this interval came as a half of the previous one, so one of these endpoints must actually be the previous midpoint  $m_{k-1}$ . It follows that bisection stops when  $|m_k - m_{k-1}| < \varepsilon$ , that is, the absolute difference is the natural stopping condition for the bisection method.

When the root is a large number, then also the approximations become large and insisting on small absolute error can be counterproductive. After all, we are interested in relative error anyway. So when our iteration seems to be taking us to large numbers, it is a good idea to use the relative difference test. In fact, it is the natural error to consider, so perhaps we should be using it always. However, it is more difficult to calculate, and for most cases it is reasonably comparable to the absolute error. Moreover, it can behave very badly if used with very small numbers.

Now we pass to the third stopping test. It is based on the idea that if the function is really small somewhere, then the root should not be far. Again, one can easily imagine situations when this would fail, very flat functions being the culprits. However, if we can prevent our function from being flat, then we do have some control over the position of the error.

**Fact 19c.1.**

Let  $f$  be a function and  $r \in \mathbb{R}$  its root. Assume that there is a neighborhood  $U$  of  $r$  on which  $f$  is differentiable and  $|f'| \geq m_1$  on  $U$  for some  $m_1 > 0$ . Then for  $\hat{r} \in U$  we have the estimate  $|r - \hat{r}| \leq \frac{1}{m_1}|f(\hat{r})|$ .

**Proof:** Since both  $r$  and  $\hat{r}$  are in  $U$ , then the closed interval  $I$  with endpoints  $r, \hat{r}$  (in the right order) is a subset of  $U$ . Consequently, the function  $f$  is continuous on  $I$  and differentiable on its interior, which means that we can apply the Mean value theorem to it. We learn that

$$\frac{f(r) - f(\hat{r})}{r - \hat{r}} = f'(\xi)$$

for some  $\xi \in I$ . Applying our assumption we obtain

$$\frac{|f(r) - f(\hat{r})|}{|r - \hat{r}|} = |f'(\xi)| \geq m_1.$$

Now we realize that  $r$  is a root of  $f$ , so  $f(r) = 0$ , and rearrange the inequality to obtain the desired estimate. □

This theorem links the function value  $f(x_k)$  with the error  $E_k = r - x_k$ , which is very nice. The key idea is the control of the derivative. The inequality  $|f'| \geq m_1$  prevents  $f$  from being too flat around  $r$ , as it forces  $f$  to grow or decrease at least as fast as the rate  $m_1$ .

What are the chances if this assumption being true? Assuming that  $f'$  is continuous, then everything depends on  $f'(r)$ . Indeed, if  $f'(r) \neq 0$ , then by continuity there must be some neighborhood  $U$  of  $r$  on which the derivative is bounded away from zero. The only bad case is therefore the one when  $f'(r) = 0$ , that is, when  $r$  is a root of higher multiplicity. Here we go again.

We conclude that if a root  $r$  is simple, then there is a neighborhood where we have a link between the value of the function and the distance from the root. If we want to have an approximation with error bounded by  $\varepsilon$ , we simply wait until  $|f(x_k)| < \varepsilon \cdot m_1$ . Unfortunately, we rarely know this  $m_1$ , so in applications this is not as useful and we are back to hoping that our stopping conditions behave reasonably.

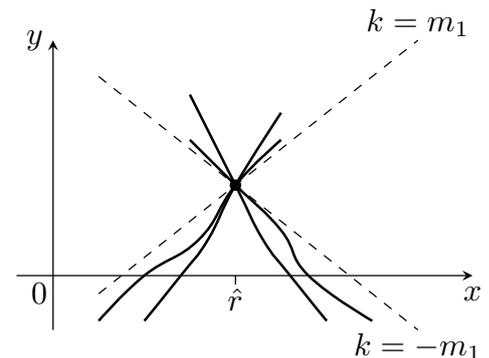
To summarize our exploration, we have three stopping conditions to choose from, but none works reliably, and even if there are some theoretical results in the right direction, we are often unable to use them due to lack of data. Sometimes people ask for more conditions to be true before they stop, typically they combine the absolute error test with the value test, but even that is no guarantee that we did not stop too soon while our approximation is still too far from the root.

Interestingly, there is a simple test that can guarantee that our approximation is good enough. One could call it a “three value test”.

**Algorithm 19c.2.**

⟨locating a root with given tolerance⟩

Given: A function  $f$ , tolerance  $\varepsilon > 0$  and a number  $\hat{r}$ .



1. Evaluate  $f(\hat{r} - \varepsilon)$ ,  $f(\hat{r})$ ,  $f(\hat{r} + \varepsilon)$ . If their signs are not the same, then there is a root  $r$  of  $f$  satisfying  $|r - \hat{r}| < \varepsilon$ .

△

Again, this test is not foolproof. What if the three numbers have the same sign? Then in general we do not know anything. It may be that there is no root, but it could also happen that there is a root but the function changes too quickly to show it. Then we have to investigate some more, for instance we can artificially lower the  $\varepsilon$  in our stopping condition to force the algorithm to run longer. We get a (hopefully) better approximation and test it again.

Of course, if it is a root of even multiplicity (think of the parabola), then this test will always fail, but when it comes to simple roots it usually performs rather well. In particular, if we have a reason to believe that the function we investigate is monotone around the root (for instance because we can analyze its derivative), then we can rely on this test entirely. Once all the signs are equal in a monotone function, then we can be sure that there is no root hiding in there.

Returning to the Newton method, we see that in algorithm 19b.1 we actually offer the absolute difference test and the value test for stopping the run.

**19c.3 Remark on residual:** When we present some solution obtained using tools of numerical mathematics, we can rarely determine its actual error. However, in many settings there is a parameter that we can evaluate. Often the thing that we are trying to find is supposed to serve some purpose. For instance, when we are trying to integrate some function  $f$ , then the result (an antiderivative  $F$ ) is supposed to serve the purpose that when we differentiate it, we get  $f$ . When we are trying to find  $\sqrt{A}$ , we are looking for a number that is supposed to give  $A$  when squared.

In such situations we can ask by how much the proposed solution missed its mark. For instance, if somebody claims to have an antiderivative  $F$  of a given function  $f$ , then  $f - F'$  tells us by how much it missed its purpose. This quantity is called a **residual**, or a **residuum**. People also talk of residual functions, residual numbers, residual vectors, depending on the setting.

The advantage is that it can be evaluated easily, unlike the error of the proposed solution. For instance, given some approximation  $x$  of  $\sqrt{13}$ , we would be hard pressed to find the exact error. On the other hand, we can easily calculate its residual  $13 - x^2$ . This is a useful concept that we will revisit later. In particular, in many situations a relationship may be found between a residual and the actual error, and a natural requirement for any iterative procedure is that it makes the residual converge to zero.

How does it apply to this chapter? The purpose of a root  $r$  of a function  $f$  is to make this function equal to zero. Therefore, given some root approximation  $\hat{r}$ , its residual would be  $0 - f(\hat{r})$ . In fact, above we proved a theorem that relates the size of this residual  $|f(\hat{r})|$  to the error. Returning to the topic of this section, the value stopping condition can be seen as a condition that works with the residual. This is another justification why it may be a good idea to use it.

△

## 19d. Order of method (order of error)

Intuitively we would judge the speed of a root-finding algorithm based on how quickly it produces valid digits. This is related to the question how often does the relative error get divided by ten, and since we always divide by the same root  $r$  in the relative error, we simply want to know how frequently does the absolute error get divided by ten during a typical algorithm run.

Mathematically, we are in some way to measure how fast sequences go to zero. The natural idea is to compare two successive terms. We already worked with such a relationship when investigating the error of bisection method. We actually had an equality there, but that is a rare situation, usually we are happy with an inequality. The message we take from the bisection method investigation is therefore this inequality:  $e_{k+1} \frac{1}{2} e_k$ . It obviously forces the sequence  $e_k$  to go to zero.

Generally, we can talk about sequences  $x_k$  that converge to zero and about comparing the speeds at which they do so. It seems natural to consider relationships of the form  $|x_{k+1}| \leq c|x_k|$  and focus on the influence of  $c$ . Obviously, the smaller the  $c$ , the faster the decrease to zero. However, it turns out that making  $c$  small is not powerful enough, and we should focus on making the “1” in the exponent bigger. Which “1”? The relationship above can be written as  $|x_{k+1}| \leq c|x_k|^1$ .

Generally, we can consider relationships of the form  $|x_{k+1}| \leq c|x_k|^q$ , and it turns out that the influence of  $q$  is incomparably stronger than the influence of  $c$ .

**Example 19d.a:** Here we will compare the trends of four sequences that tend to zero. They all start with  $x_1 = 0.0001 = 10^{-4}$ , but  $x_{k+1}$  depends on  $x_k$  in different ways.

$x_{k+1}$	$x_1$	$x_2$	$x_3$	$x_4$
$\frac{1}{2}x_k$	0.0001	0.00005	0.000025	0.0000125
$\frac{1}{5}x_k$	0.0001	0.00002	0.000004	0.0000008
$x_k^{1.5}$	0.0001	0.000001	0.000000001	0.0000000000000316...
$x_k^2$	0.0001	0.00000001	0.0000000000000001	0.00000000000000000000000000000001

Recall our interpretation that zeros after the decimal dot correspond to the number of digits in our approximation that are correct. The first row is as expected for the bisection method.

The second row can be imagined as coming from some better method that improved the constant  $c$  to  $\frac{1}{5}$ . For instance, we may be dividing intervals into five parts instead of two. There is some improvement in how quickly new digits appear, we get roughly a new digit at each iteration in our example, although it will not last,  $x_4 = 0.00000016$ .

So it is better than the previous relationship, but the next row shows that this improvement is weak compared to what can be achieved by increasing the power  $q$  in the comparison. This is where the real gains are made.

In the last row, the last number is  $10^{-32}$ . Of course we do not normally write such numbers in this way, but here it sends a clear message: The speed of convergence is spectacular. In fact, the number of zeros after the decimal dot at least doubles at each iteration. It would be really nice if errors behaved like that in our methods.

We will return to this comparison in example .

△

The moral of the story is that if we want to force a sequence to go to zero fast, we should focus on  $q$  in the relationship  $|x_{k+1}| \leq c|x_k|^q$ , the value of  $c$  is of secondary importance.

Once we know how to rate sequences convergent to zero, we can rank all convergent sequences: We simply look at the behavior of the “errors”. That is, if a sequence  $x_k$  converges to some  $x_\infty$ , then we look for relationship of the form  $|x_\infty - x_{k+1}| \leq c|x_\infty - x_k|^q$

**Definition 19d.1.**

Consider a sequence  $\{x_k\}$  that converges to some  $x_\infty$ .

We say that it converges with order (or rate) of convergence  $q > 0$  if there is  $C$  such that  $|x_\infty - x_{k+1}| \leq C|x_\infty - x_k|^q$  for all  $k$ .

There are other definitions of this notion around, but they are equivalent to this one. Note that when we say “for all  $k$ ”, we mean for all  $k$  used in indexing the given sequence. In fact, we do not care much, because the rate of convergence is decided at the tail of the sequence anyway, the first terms do not matter.

Note also that this notion is hierarchic. If a sequence converges for a certain rate  $q$ , then it also converges with all rates  $p$  satisfying  $p < q$ . Since more is better, we always try to determine the largest possible  $q$  and consider this to be the right rate of convergence for that sequence.

This notion is often called the  $Q$ -rate of convergence to distinguish it from another, similar notion. Why do we need two? recall that we plan on applying it to errors  $E_k$  of our approximations. However, in the case of bisection we were not able to establish any relationship; rather, we established an appropriate relationship for upper estimates of the errors. This can also happen in other examples, and a notion was created to take care of this situation.

**Definition 19d.2.**

Consider a sequence  $\{x_k\}$  that converges to some  $x_\infty$ .

We say that it converges with  $R$ -order (or  $R$ -rate) of convergence  $q > 0$  if there is a sequence  $\{e_k\}$  of upper estimates for  $|x_\infty - x_k|$ , that is,  $|x_\infty - x_k| \leq e_k$  for all  $k$ , such that it  $Q$ -converges to zero.

Obviously, if a sequence converges with a certain  $Q$ -rate  $q$ , then it also  $R$ -converges with the same rate, because we can simply take  $e_k = |x_\infty - x_k|$ . However, the converse is not true, so mathematically, these two notions are not equivalent.

However, in numerical mathematics we often do not worry about this distinction, simply because we usually do not have much choice. For some methods, we can establish a relationship between errors, and we naturally use the first definition. For methods like bisection we have no choice but to use the other. We will therefore simply talk about rate of convergence.

Now we apply this to iterative methods for finding roots. These produce sequences, but we know that these need not always converge. So when we want to assign some ranking to methods, we have to take into account only convergent runs. But even then things are not so simple, because we already noted that roots of higher multiplicity cause troubles. Moreover, usually we need some assumptions on functions  $f$  to be able to deduce anything. We will therefore also disregard cases that cause trouble when ranking methods.

The definition that follows is not completely rigorous. Some authors avoid this by simply not assigning any order to methods, and talk only of sequences generated by them. However, many authors do find it useful to talk about order of method, as it sends a direct message.

**Definition 19d.3.**

Consider a certain iterating method for finding roots of functions. We say that it is a **method of order**  $q$ , or that it has **error of order**  $q$ , where  $q > 0$ , if it satisfies the following condition:

Whenever this method produces a sequence  $\{x_k\}$  converging to a certain root  $r$  of a function  $f$ , this root is simple and the function sufficiently smooth, then  $\{x_k\}$  converges to  $r$  with rate  $q$ .

Note that it does not really make much sense to consider orders smaller than 1. Moreover, for methods of order 1 we actually have to focus on the constant  $c$  (note that each sequence is allowed its own  $c$ ), we have to insist that there is a common upper bound  $C < 1$  for all these  $c$ . In other words, all sequences  $\{x_k\}$  produced by a method of order one must satisfy  $|r - x_{k+1}| \leq C|r - x_k|$  with this common  $C < 1$ . We also say that these methods are of linear order.

Our analysis in section 19a has proved the following statement.

**Theorem 19d.4.**

The bisection method is of linear order (order 1).

Actually, with the bisection method there are no roots and functions to avoid, the rate of convergence is general and reliable (and small).

As we saw in the example above, we would prefer to have a rate of convergence greater than 1. How about the Newton method? Remarkably, while we cannot determine the errors, it is actually possible to establish a relationship between them. Formally, one gets a result about Q-rate of convergence.

**Theorem 19d.5.**

Let  $r$  be a root of a function  $f$  that is twice continuously differentiable on some neighborhood  $U$  of  $r$  and for which there are  $m_1, M_2 > 0$  such that  $|f'| \geq m_1$  and  $|f''| \leq M_2$  on  $U$ .

Consider a sequence  $\{x_k\}$  generated by the Newton method. Then for all  $x_k, x_{k+1} \in U$  we have estimates

$$|r - x_{k+1}| \leq \frac{M_2}{m_1} |r - x_k|^2 \quad \text{and} \quad |r - x_{k+1}| \leq \frac{M_2}{2m_1} |x_{k+1} - x_k|^2.$$

The technical assumptions are satisfied in cases when the function  $f$  is twice continuously differentiable at  $r$  and the root  $r$  is simple (then  $f'(r) \neq 0$ ). This perfectly fits with the restrictions described above and we get the following statement.

**Theorem 19d.6.**

The Newton method is of order 2.

For roots of higher multiplicity the Newton method generates sequences that converge linearly. This is the third time that we see roots of higher multiplicity being obtuse.

Note the second formula in the theorem. It actually offers a control over the error of approximation using the knowledge of the last two iterations, which is great. Unfortunately, it can be used only if we know those bounds  $m_1, M_1$ , which is often not the case.

Since this theorem is important, one would expect to see its proof. Since it is rather technical, we prefer to leave it to chapter 21 so that we can focus on crucial ideas here. Another reason is that the proof suggests an interesting modification of the Newton method that just fits there.

We now know that the Newton method is as fast as the best sequence in example . This is great. On the other hand, it sometimes does not converge at all. This is a tradeoff, and there are strategies that use the good properties while largely avoiding the bad ones. One particular strategy is to first narrow down the location of the root using a more reliable method, for instance the bisection method. Once we have the error markedly smaller than 1, there is a high probability that the Newton method will go directly to the root, and it does it with a lightning speed.

One particular advantage of the Newton method is its ability to create iterating schemes. We will show some in the section . On the other hand, one serious drawback is its reliance on the knowledge of derivative. When the function  $f$  is known only experimentally, we do not have this information available.

**19d.7 Remark:** In remark 19c.3 we introduced the notion of residual. Sometimes we can establish a relationship  $r_{k+1} \leq Cr_k^p$  for residuals corresponding to sequences generated by a certain method. This may be a useful information that sheds further light on how such a method performs. We will see such an example in section .

△

## 19e. The secant method

Therefore there is a big demand for so-called derivative-free methods. Of course, the bisection method is one, but we want something faster.

One interesting idea is to start with the fast Newton method, and ask whether we could do something with the derivative in the formula. Since we are in the world of numerical mathematics here, we naturally think back to chapter 4, where we approximated derivatives numerically. In particular, we had a simple method that only needed the knowledge of  $f$  at two points. Now when we run the Newton method, we do get to know  $f$  at many points  $x_k$ . So here is the idea: At stage  $k$ , we have  $x_k$ , but also  $x_{k-1}$  from the previous stage. We can use these two to approximate the derivative at  $x_k$ :

$$f'(x_k) \approx \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

and use it in the Newton formula. We get

$$\begin{aligned} x_{k+1} &= x_k - \frac{f(x_k)}{\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}} = x_k - \frac{f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})} \\ &= \frac{x_k(f(x_k) - f(x_{k-1})) - f(x_k)(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})} \\ &= \frac{f(x_k)x_{k-1} - f(x_{k-1})x_k}{f(x_k) - f(x_{k-1})}. \end{aligned}$$

This looks like an interesting iterative formula. However, note one fundamental difference. Our previous iterative methods only used the immediate present to produce the future iteration. Symbolically,  $x_k \mapsto x_{k+1}$ . Our new method uses the present and also goes one step into the past to produce a new iterate:  $x_{k-1}, x_k \mapsto x_{k+1}$ . This is not anything bad, but this tells us that in order to start, this procedure actually needs two initial guesses  $x_0, x_1$ .

There is actually another thing that we can try when we know a function  $f$  at two points. We may take the corresponding points on the graph, connect them with a line and check where it intersects the  $x$ -axis. If the picture is any guide, this new point should be a better estimate for the root of this function.

The secant line has the slope

$$k = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}},$$

and thus the equation

$$s : y = f(x_k) + k(x - x_k) = f(x_k) + \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}(x - x_k).$$

Where does it intersect the  $x$ -axis?

We set  $y$  equal to zero and solve for  $x$ , obtaining

$$x = \frac{f(x_k)x_{k-1} - f(x_{k-1})x_k}{f(x_k) - f(x_{k-1})}.$$

Remarkably, this is the same formula as above. Now we have two reasons to think that this could be a good idea, let's give it a name.

### Algorithm 19e.1.

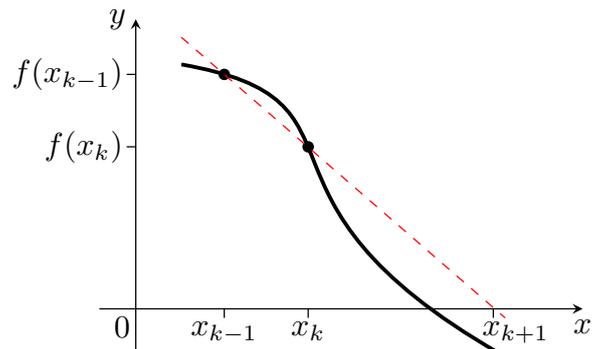
$\langle$ secant method for finding root of a function  $f$  $\rangle$

Given: a function  $f$  continuous on  $\mathbb{R}$  and a tolerance  $\varepsilon$ .

0. Choose  $x_0, x_1$ . Let  $k = 1$ .

1. Let  $x_{k+1} = \frac{x_{k-1}f(x_k) - x_kf(x_{k-1})}{f(x_k) - f(x_{k-1})}$ .

If  $|x_{k+1} - x_k| < \varepsilon$  or  $|f(x_{k+1})| < \varepsilon$  then algorithm stops, output is  $x_{k+1}$ .



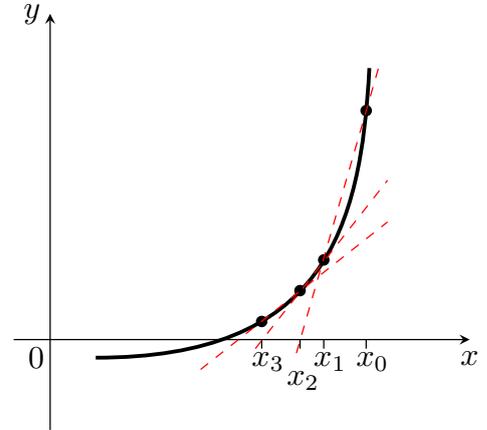
Otherwise increase  $k$  by one and go back to step 1.

△

What can we expect of this method?

- The secant method is unreliable.
- We have no control over the error.
- The secant method is quite fast.

Regarding the first observation, as an approximation of the Newton method, this new methods shares its bad traits. We may have trouble with division by zero (and we cure it by adjusting the initial guesses like with the Newton method), it can travel around before settling down, and some shapes are not good, namely the ones that we saw for the Newton method, like the camel one. When in a bad situation, the secant method may oscillate or run away. However, just like for the Newton method, also here we can identify some shapes that make the secant method work, like the one on the right. We see a nice monotone sequence being generated there, which is a nice bonus.



**Theorem 19e.2.**

Let  $f$  be a function continuous on an interval  $[a, b]$ . Assume that it is concave-up or concave-down on  $[a, b]$  and  $f(a)f(b) < 0$ . Then for arbitrary values  $x_0, x_1$  chosen so that  $f(x_0) > 0, f(x_1) > 0$  for the concave-up case, or  $f(x_0) < 0, f(x_1) < 0$  for the concave-down case, the sequence  $\{x_k\}$  generated by the secant method converges to a root  $r \in (a, b)$  of  $f$ .

Try to draw a picture capturing the meaning of this theorem, it is a good exercise.

Since the secant method is not a bracketing method, we again have no control over the error, so we are reduced to guessing when to stop this algorithm, and we arrive at the same stopping conditions as described above. Generally speaking, also this method requires supervision.

Regarding the speed, by approximating the derivative we actually lost a bit, but not that much.

**Theorem 19e.3.**

Let  $f$  be a function twice continuously differentiable on some neighborhood  $U$  of its root  $r$ . Let  $\{x_k\}$  be a sequence generated by the secant method such that  $x_k \rightarrow r$ . If the root  $r$  is simple, then there is a constant  $K > 0$  and  $N \in \mathbb{N}$  such that  $|r - x_{k+1}| \leq K|r - x_k|^\varphi$  for  $k \geq N$ , where  $\varphi = \frac{1+\sqrt{5}}{2}$ . If the root is of higher multiplicity, then the convergence is linear.

Note that  $\alpha \approx 1.618$ , so this convergence rate is definitely better than linear, but not as good as quadratic. Note that as usual, roots of higher multiplicity cause troubles.

**Corollary 19e.4.**

The secant method is of order  $\varphi = \frac{1+\sqrt{5}}{2}$ .

Now what kind of a number is this  $\varphi$ ? Actually, it is the famous golden ratio, famous for its role in arts, appearing in nature at unexpected places and perhaps, according to some, even capturing the substance of the universe. People of esoteric bend should definitely use the secant method to find their roots.

What can we expect of it in practice?

**Example 19e.a:** In example we explored sequences that all started with the same  $e_1$  and grew at different rates. One of them had rate of convergence 1.5, which is close to the rate of the secant method, so it gives us some idea of what to expect.

Here we will try it again, this time with the right order. To this end we will look at the sequence that starts with  $e_1 = 0.01$  and given by the recursive formula  $e_{k+1} = e_k^\varphi$ . However, this time we will not list the values  $e_k$ , but the number of zeros after the decimal dot for them. In this way we will get a fairly good idea of how fast the secant method improves its approximations and generates correct digits.

$k:$	2	3	4	5	6	7	8	9
zeros	3	5	8	13	22	35	58	93

Especially the first four numbers should look familiar. It is the Fibonacci sequence. The next number is different, but you can observe that most of the time, numbers of zeros are the sum of the previous two, so the number of zeros does behave almost like the Fibonacci sequence. Actually, this sequence is closely related to the golden ratio, so once there is one, the other is not far. Still, it is a nice observation.

△

Just like the Newton method, the secant method combines advantages with disadvantages. However, in the following section we will see that it may be even better than it seems.

## 19f. Practical considerations

What did we learn? We have the bisection method: the lumbering giant that takes on average three and a third iterations to gain one correct digit, but is bulletproof reliable and yields a guaranteed error estimate.

On the other hand we have the lightning fast Newton method that doubles the number of correct digits with each iteration, but that only if it happens to converge and we happen to get close to the root to start with. Now take your choice.

What people usually do is to combine them. First they fish around for the root, trying numbers until they hit on opposite signs, then they narrow the root down to a small interval using bisection, it is enough to get an interval shorter than 1, which can be done reasonably quickly. And then they switch to the Newton method to quickly obtain very good approximation. Which they cannot confirm, but readers of this book know our test 19c.2 can do the trick. Note that this approach works only if we have the derivative available to us. If we don't, then we can use the secant method as the finisher.

There are several topics that were not addressed yet. We leave some of them to the next chapter, but the notion of order deserves a closer look before we leave this chapter.

There are several problems with the definition of order. We start with the following idea. Imagine a new, lightning-fast method, there each step looks like this:

1. Given  $x_k$ , we look at the tangent line to the graph of  $f$  at  $x_k$  and call its intersection with the  $x$ -axis  $y_t$  for “temporary”.
2. We construct a tangent line to the graph of  $f$  at  $y_t$  and define  $x_{k_1}$  as its intersection with the  $x$ -axis.

You can see the trick: We take two steps of the Newton method and pretend that it is just one step. How fast is this new method? Consider some simple root  $r$  and a run that converges to it. Let  $E_k$  be the error of  $x_k$ . Since  $y_t$  was actually constructed using the Newton method, the error of  $y_t$  should satisfy  $|E_t| \leq c|E_k|^2$ . Similarly, the error of  $x_{k_1}$  should satisfy  $|E_{k_1}| \leq c|E_t|^2$ . Putting these two inequalities together we obtain  $|E_{k_1}| \leq c^3|E_k|^4$ . Voila, we have a method of order 4.

Now mathematicians would not do anything like this, they are interested in knowledge, not in upmanship, but this shows that the mathematical notion of order is not as solid as it could be. Is it possible to actually find some reliable measure of speed?

There is another concern. Some methods do not proceed uniformly, but mix “faster” and “slower” steps. How do we judge their speed then? It is possible to develop a notion of an “average rate of convergence”.

Consider some sequence  $\{x_k\}$  that converges to zero with known order of convergence  $q$ . We will check how it behaves after several steps:

$$\begin{aligned} |x_{k+1}| &\leq c|x_k|^q, \\ |x_{k+2}| &\leq c|x_{k+1}|^q \leq c(c|x_k|^q)^q = c^{q+1}|x_k|^{q^2}, \\ |x_{k+3}| &\leq c|x_{k+2}|^q \leq c(c^{q+1}|x_k|^{q^2})^q = C|x_k|^{q^3}, \dots \end{aligned}$$

In general,  $|x_{k+n}| \leq c|x_k|^{q^n}$ . This can be used to determine average rates of convergence over several iterations. If we can find a constant  $p$  so that  $|x_{k+n}| \leq c|x_k|^p$ , then the average order is  $q = \sqrt[n]{p}$ .

Now it is time to put things together. When we run an algorithm, its speed does not depend on number of iterations, because “iterations” are just ways of writing ideas. The speed depends on the number of calculations and other things that the computer must do (comparisons, checking on tests and such). Typically, the most work is spent on evaluating the function  $f$  and perhaps its derivatives. In some cases this can be extreme, for instance when the value has to be determined by an experiment that takes considerable time.

It therefore makes sense to measure performance of algorithms based on the number of evaluations that are needed. This does not have an official name, but it is of extreme importance to people who actually calculate things. We can call it the practical order. It is very useful and we will make good use of it in the next chapter. How do our three basic methods fare?

**The bisection method:** In every iteration we need just one function evaluation, namely  $f(m_k)$ . Then we are comparing this with signs at the endpoints of  $[a_k, b_k]$ , but this information is already calculated in previous iterations and we keep it stored somewhere. We conclude that each “mathematical” iteration requires one evaluation, and therefore the mathematical order coincides with the practical one.

**The secant method:** In every iteration we need just one function evaluation, namely  $f(x_k)$ . For  $x_{k+1}$  we also need  $f(x_{k-1})$ , but we keep it from the previous iteration for sure. Again, the mathematical order coincides with the practical one.

**The Newton method:** For every iteration we need to find  $f(x_k)$  and  $f'(x_k)$ , that is, two function evaluations. This means that while the improvement  $|E_{k+1}| \leq c|E_k|^2$  happens in one mathematical iteration, from practical point of view this actually represents two steps, and the practical order is therefore lower. If we denote this practical order  $q$ , we have  $|E_{k+1}| \leq c|E_k|^{q^2}$  by our above observation. Therefore  $q^2 = 2$  and the practical order of the Newton method is  $p = \sqrt{2} = 1.41\dots$ , which is even lower than the secant method.

This is interesting, and it can be compared in another way. Consider a certain approximation  $x_k$  of a root with error  $E_k$ . If we apply the Newton method, this error improves to  $c|E_k|^2$  (approximately). However, it will take two function evaluations, which means that the secant method can make two iterations during the same time. It can therefore improve the error to  $C|E_k|^{\varphi^2}$ , and  $\varphi^2 \approx 2.6$ . This is significantly better, even if we take into account that the constant  $C$  can be larger than the constant  $c$  for the Newton method.

We thus see an interesting picture. While on paper the Newton method is faster, in practice the secant method runs faster, and it also does not need any derivative as a bonus. Thus it actually makes sense to use the secant method rather than the Newton method.

So why is the Newton method so popular? First, when the function we investigate is simple to evaluate, then the service of iterations starts being important, and the Newton method does run faster. This is especially true if we do not substitute to the Newton formula as given, but first

create a dedicated iterating scheme for the given function. Determining  $x_{k+1}$  then means just a substitution of  $x_k$  into an expression that can be significantly simpler than the ratio  $\frac{f(x_k)}{f'(x_k)}$ .

In fact, the ability of the Newton method to prepare iteration schemes of high practical utility, as we will see in the next section, is the second reason why it is very useful.

Finally, the Newton method is very versatile. It can be readily generalized to more dimensional settings, in fact once we are in a situation when some kind of differentiation makes sense, then the Newton method has a good chance of working, which can include even infinitely-dimensional situations.

Now the reader probably wonders: Where is the obligatory section on numerical stability? The beauty of iterative methods is that we do not have to worry much about them. The only significant source of errors is the evaluation of  $f$  itself. Usually we worry about propagation of such errors in our calculations. However, here all methods are supposed to improve the approximations that they produce. This means that even if some approximation  $x - k$  is determined with a small numerical error, this will be remedied when  $x_{k+1}$  is created. In a way, iterative methods are self-correcting.

Thus we can expect that numerical errors in  $x_k$  should not differ much from the base roundoff error of the system. This is typically negligible compared to the desired precision  $\varepsilon$ , because in general it does not make sense to ask for a precision that is close to the base precision of the system.

## 20. Finding fixed points numerically

We return to the problem of solving algebraic equations numerically, this time from another angle. Every algebraic equation can be written in the form  $\varphi(x) = x$ . For instance, the quadratic equation  $x^2 - 3x + 2 = 0$  can be written as  $x = \frac{1}{3}(x^2 + 2)$ , now  $\varphi(x) = \frac{1}{3}(x^2 + 2)$ . We had a name for numbers  $r$  satisfying  $f(r) = 0$ . There is also a name for numbers solving the new kind of equation.

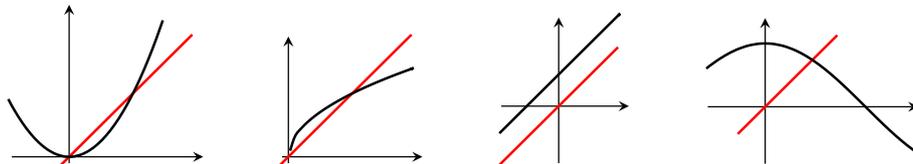
**Definition 20.1.**

Let  $\varphi$  be a function.

By a **fixed point** of  $\varphi$  we mean any number  $x_f$  satisfying  $\varphi(x_f) = x_f$ .

In the previous chapter we used the equation  $\cos(x) = x$  as an inspirational example of a simple equation that cannot be solved analytically. We had to rewrite it as  $\cos(x) - x = 0$  so that we can use tools that we learned there. However, now we can treat it as it came, as an equation asking for a fixed point of the function  $\varphi(x) = \cos(x)$ .

Most people associate functions with their graphs. In this setting, the fixed-point equation  $\varphi(x) = x$  can be interpreted as a problem of finding the intersection of two graphs, the graph of  $y = \varphi(x)$  and the graph of  $y = x$ .



Just looking at the pictures we make an educated guess that the functions  $\varphi(x) = x^2$  and  $\varphi(x) = \sqrt{x}$  have fixed points  $x_f = 0$  and  $x_f = 1$ , while the function  $\varphi(x) = x + 0.5$  has no fixed point at all. Finally, we see that  $\varphi(x) = \cos(x)$  has a fixed point.

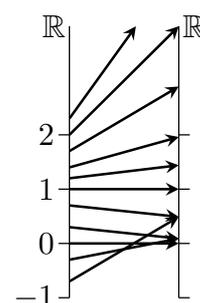
This point of view has its uses and we will return to it. However, the substance of the notion of a fixed point is better appreciated when we see functions as mappings that send numbers from one set to another.

**Example 20.a:** Here is an example for  $\varphi(x) = x^2$ :

The picture suggests that regions above  $x = 1$  get shifted up and stretched by this mapping, while regions between  $-1$  and  $1$  get somehow folded to appear between  $0$  and  $1$  and pulled towards  $0$ .

It is quite obvious that there are only two numbers that are preserved under this transformation, namely  $x_f = 0$  and  $x_f = 1$ . That is the substance of being a fixed point.

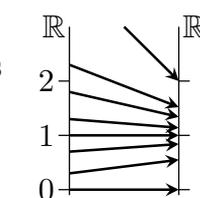
Similarly, when we view  $x^2$  as a number-sending process, then we see exactly these two numbers that get sent to themselves.



Here we see a similar picture for the function  $\varphi(x) = \sqrt{x}$ .

We know that this function makes numbers larger than  $1$  smaller, while numbers between  $0$  and  $1$  get larger.

We see the same fixed points as in the previous example,  $x_f = 0$  and  $x_f = 1$ .



△

This point of view allows us to appreciate one of the great advantages of the notion of a fixed point: It is a very general idea that can be applied to virtually any mapping that maps a set into itself.

**Example 20.b:** Consider the set  $M$  of all invertible (regular, nonsingular) square matrices of all dimensions. This is just a set, it does not have any algebraic structure, because square matrices of different dimensions cannot be added or multiplied.

We consider the mapping  $\varphi: M \mapsto M$  defined as  $\varphi(A) = A^{-1}$ . Does it have any fixed points?

We are looking for matrices such that  $A^{-1} = A$ . There is definitely one, namely the identity matrix  $E_n$  for any  $n$ . However, there are more. In linear algebra these are called involutory matrices and quite a lot is known about them, in particular they include orthogonal matrices, which is a pretty important group.

Simple involutory matrices can be obtained from identity matrices by changing signs of entries or by permuting rows.

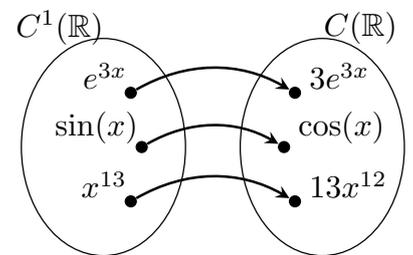
△

**Example 20.c:**

Consider the mapping  $D$  that sends functions to their derivatives. Formally,  $D[f] = f'$ . For instance,  $D[\sin(x)] = \cos(x)$ .

On the other hand, we cannot apply this mapping to the function  $|x|$ , unless we decide to work on, say  $(13, \infty)$ . Obviously we need a more precise specification of what we mean.

To make ourselves clear, we choose as our starting set (domain) the set  $C^1(\mathbb{R})$  of all real functions on  $\mathbb{R}$  that are differentiable and their derivatives are continuous. Then this mapping  $D$  can accept elements of this set as arguments. The outputs no longer need to have a derivative, but they are continuous, so the target set (co-domain) could be the set  $C(\mathbb{R})$  of all continuous real functions on  $\mathbb{R}$ . We therefore have this setup:



$$D : C^1(\mathbb{R}) \mapsto C(\mathbb{R}); \quad D : f \mapsto f'.$$

Now definitely the function  $|x|$  does not belong to the domain of  $D$ .

Every student who passed introductory calculus knows that this mapping  $D$  has two fixed points (which means functions now), namely  $e^x$  and  $0$  (a constant function). There are no other functions like this.

As we will see shortly, we prefer the mapping to have its domain and co-domain identical when talking about fixed points, or at least the co-domain should be a subset of the domain. This is not the case here, and it is caused by the fact that by differentiating a function we in a way deprive it of its properties - it loses one order of derivative, it may even lose continuity. Practically speaking, if we try to be more restrictive with the co-domain so that it becomes a part of the domain, we are in turn forced to be more restrictive in the domain. For instance, if we wanted to make sure that the outputs are differentiable, we would have to start with twice-differentiable functions, so the co-domain would not be a part of the domain again.

There is a way to get out of this vicious circle, we can consider the set  $C^\infty(\mathbb{R})$  of functions that have derivatives of all orders on  $\mathbb{R}$ . Derivatives of such functions still keep this property, so we get the following nice picture:

$$D : C^\infty(\mathbb{R}) \mapsto C^\infty(\mathbb{R}); \quad D : f \mapsto f'.$$

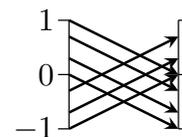
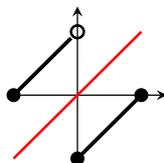
The space  $C^\infty(\mathbb{R})$  is quite rich, for instance it includes polynomials, the exponential function, and sine and cosine. On the other hand, the function  $x^{4/3}$  does not belong there. It is defined on  $\mathbb{R}$  and continuous derivative  $\frac{4}{3}x^{1/3}$  there, but its second derivative (namely  $\frac{4}{9} \frac{1}{x^{2/3}}$ ) exists only when  $x \neq 0$ .

△

Can we identify situations when a function must have a fixed point? One way to prevent a function from moving things around too much is to require that it maps some interval into itself. But this is not enough. For instance, the function  $x^2$  maps  $(0, 1)$  onto  $(0, 1)$ , but does not have

a fixed point there. This inspires us to focus just on closed intervals. However, that need not be enough.

Consider the function  $\varphi(x)$  defined as  $\varphi(x) = x - 1$  for  $x \geq 0$ , while  $\varphi(x) = x + 1$  for  $x < 0$ . Then  $\varphi$  maps the interval  $[-1, 1]$  into itself, but it has no fixed point there.



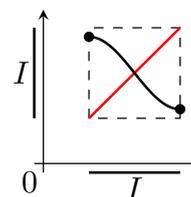
However, the action of this function is disjointed at places. When we restrict our attention only to reasonably smooth acting functions, then it is not possible to send an interval into itself without leaving some point in its place.

**Theorem 20.2.**

Let  $\varphi$  be a function on some closed bounded interval  $I$ .

If  $\varphi[I] \subseteq I$  and  $\varphi$  is continuous on  $I$ , then  $\varphi$  has a fixed point  $x_f \in I$ .

Geometrically, this statements seems obvious. The condition  $\varphi[I] \subseteq I$  means that the graph must be in the square  $I \times I$ . Since  $\varphi$  is defined on the whole closed interval  $I$ , then its graph must start somewhere on the left edge of that square and end somewhere on the right edge. And since the graph is uninterrupted by continuity, it must cross the diagonal somewhere, even if it were at one of the endpoints.



**Proof:** As a closed interval,  $I$  can be written as  $[a, b]$  for some  $a < b$ . Consider the auxiliary function  $h(x) = \varphi(x) - x$ . Since both  $\varphi(x)$  and  $x$  are continuous functions on  $I$ , so is the function  $h$ . Because  $\varphi$  maps  $I$  into  $I$ , we must have  $\varphi(a) \in I$  and  $\varphi(b) \in I$ , in particular  $\varphi(a) \geq a$  and  $\varphi(b) \leq b$ . Thus

$$h(a) = \varphi(a) - a \geq a - a = 0,$$

$$h(b) = \varphi(b) - b \leq b - b \leq 0.$$

This means that  $h(b) \leq 0 \leq h(a)$  and  $h$  is continuous, hence by the Intermediate value theorem the value 0 must be attained at some point  $c$  in  $I$ . But  $h(c) = 0$  means  $\varphi(c) = c$  as needed.  $\square$

**Example 20.d:** One of the motivational examples in the previous chapter was the simple yet difficult equation  $\cos(x) = x$ . This time we do not have to rewrite it into another form and treat it as it is: a fixed-point type of problem.

The function  $\cos$  definitely maps the interval  $[-1, 1]$  into itself, so it must have some fixed point there by the above fact.

$\triangle$

## 20a. (Plain) iteration

So much for the theory, but how do we find this fixed point? The default approach is to use an iterative procedure, that is, we will produce a potentially infinite sequence of approximations  $x_k$  that will hopefully converge to some fixed point  $x_f$ . The way in which these approximations are produced may seem weird at the first sight, but it can be remarkably efficient. Note that the coming algorithm has the same problems with error control as the Newton method, the second method and many others, so we have to resort to the usual stopping conditions, with their advantages and disadvantages.

**Algorithm 20a.1.**

⟨iteration for fixed points⟩

Given: A function  $\varphi$  and tolerance  $\varepsilon > 0$ .

**0.** Choose a number  $x_0$ . Let  $k = 1$ .

**1.** Let  $x_{k+1} = \varphi(x_k)$ .

If  $|x_{k+1} - x_k| < \varepsilon$  or  $|f(x_{k+1})| < \varepsilon$  then algorithm stops, output is  $x_{k+1}$ .

Otherwise increase  $k$  by one and go back to step **1**.

△

People also call this a plain iteration or a simple iteration, because there will be more sophisticated versions coming up. We offered the usual suspects for stopping the algorithm, and as usual, for larger  $x_k$  people sometimes use the relative stopping condition. How does the algorithm work?

**Example 20a.a:** Consider the problem  $\cos(x) = x$ . To find the fixed point we decide to use the plain iteration, and we start at, say,  $x_0 = 0$ .

Then  $x_1 = \cos(x_0) = \cos(0) = 1$ . So far so good.

Next,  $x_2 = \cos(x_1) = \cos(1)$ . Fine.

Next,  $x_3 = \cos(x_2) = \cos(\cos(1))$ .

Next,  $x_4 = \cos(x_3) = \cos(\cos(\cos(1)))$ .

Perhaps we could ask a calculator for help. In fact it is simple, we just put in 1, make sure that it is switched to radians and then just keep pressing the cos button and read out the answers (unless you have a modern calculator where you have to type in an expression, good for you):

0., 1., 0.540..., 0.857..., 0.654..., 0.793..., 0.701..., 0.764...

and after a while

0.7392..., 0.7390..., 0.7391...

Remarkably, this seems to converge to the solution we found in the previous chapter.

△

**Example 20a.b:** How about the two powers?

We start with  $\varphi(x) = \sqrt{x}$  and the initial guess 0.5. Pressing the  $\sqrt{\quad}$  button repeatedly yields the following numbers:

0.5, 0.71..., 0.84..., 0.92..., 0.96..., 0.978..., 0.989..., 0.994...

and even a pessimist would start thinking that this looks like a sequence that converges to 1, which is a fixed point as we observed above.

Try the same procedure with  $x_0 = 2$  to convince yourself that the resulting sequence again converges to 1.

Now we look at  $\varphi(x) = x^2$ . We try again  $x_0 = 0.5$ . Hitting the  $x^2$  button repeatedly produced numbers

0.5, 0.25, 0.0625, 0.0039..., 0.000015...

that seems to go to zero.

After all, if  $x_0 = \frac{1}{2}$ , then  $x_1 = x_0^2 = \left(\frac{1}{2}\right)^2 = \frac{1}{4}$ ,  $x_2 = x_1^2 = \left(\frac{1}{4}\right)^2 = \frac{1}{16}$ ,  $x_3 = x_2^2 = \left(\frac{1}{16}\right)^2 = \frac{1}{256}$ , this really looks like this sequence should converge to zero, and quite fast.

On the other hand, if we start with  $x_0 = 2$ , then the resulting sequence goes  $x_1 = 2^2 = 4$ ,  $x_2 = 4^2 = 16$ ,  $x_3 = 16^2 = 256$ ,  $x_4 = 256^2 = 65536$ , this looks like a divergent sequence.

So it seems that the plain iteration can work, but not always (actually, when I saw this for the first time I found the former more surprising).

△

**Example 20a.c:** We commented that the notion of fixed point can be very general. Recall the differential operator  $D$  that we now consider to map  $C^\infty$  to  $C^\infty$ . We will apply the plain iteration to this mapping, with  $x_0 = x^3 + 13x$ .

Then  $x_1 = D[x_0] = [x^3 + 13x]' = 3x^2 + 13$ ,  $x_2 = D[x_1] = [3x^2 + 13]' = 6x$ ,  $x_3 = D[x_2] = [6x]' = 6$ ,  $x_4 = [6]' = 0$ ,  $x_5 = [0]' = 0, \dots$

Even without knowing how convergence is done for functions, this looks like a sequence of functions that converges to the fixed point of  $D$ , the zero constant function.

What if we start with  $x_0 = e^{x/3}$ ? Then  $x_1 = [e^{x/3}]' = \frac{1}{3}e^{x/3}$ ,  $x_2 = [\frac{1}{3}e^{x/3}]' = \frac{1}{9}e^{x/3}$ ,  $x_3 = [\frac{1}{9}e^{x/3}]' = \frac{1}{27}e^{x/3}$ , in general  $x_k = \frac{1}{3^k}e^{x/3}$ .

There are several possible ways to define convergence of functions, but the most popular ones (pointwise convergence, uniform convergence on bounded intervals) would agree that the sequence of functions  $\{\frac{1}{3^k}e^{x/3}\}$  actually converges to the zero constant function.

So things seem to work quite well even in this abstract setting.

On the other hand, taking  $x_0 = \sin(x)$  does not get us far, obviously:

$$x_1 = \cos(x), x_2 = -\sin(x), x_3 = -\cos(x), x_4 = \sin(x), \dots$$

We briefly return to the example with matrices. Starting with a matrix  $x_0 = A$ , the plain iteration produces the sequence  $x_1 = A^{-1}$ ,  $x_2 = (A^{-1})^{-1} = A$ ,  $x_3 = A^{-1}$ ,  $\dots$  that does not get us anywhere. So in this example the plain iteration does not help much.

△

The experiments suggest that the (plain) iteration can work, but only sometimes. We need to investigate this closer, and start by confirming that if such an iteration converges, then it already must yield the right object.

**Theorem 20a.2.**

Let  $\varphi$  be a function. For some  $x_0 \in \mathbb{R}$ , consider the sequence defined recursively by  $x_{k+1} = \varphi(x_k)$  for  $k \in \mathbb{N}_0$ .

If  $x_k \rightarrow x_f$  and  $\varphi$  is continuous at  $x_f$ , then  $\varphi(x_f) = x_f$ .

**Proof:** The recursive equality can be seen as an equality between two sequences,  $\{x_{k+1}\}$  and  $\{\varphi(x_k)\}$ . If their terms are equal, then the limit must also be the same:

$$\lim_{k \rightarrow \infty} (x_{k+1}) = \lim_{k \rightarrow \infty} (\varphi(x_k)).$$

If  $\{x_k\}$  converges to  $x_f$ , then also the shifted sequence  $\{x_{k+1}\}$  must converge there, which settles the left-hand side. On the right we recall one theorem from calculus: If  $\varphi(x)$  is continuous at the limit point  $x_f$  of  $\{x_k\}$ , then we can move the limit operation inside the function. We obtain

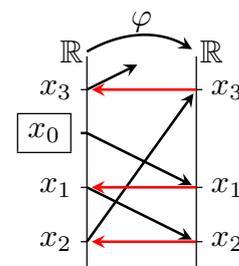
$$x_f = \varphi\left(\lim_{k \rightarrow \infty} (x_k)\right) \implies x_f = \varphi(x_f).$$

We confirmed that the limit point is also a fixed point of  $\varphi$ . □

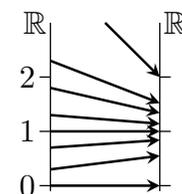
In words, we proved that if this procedure outputs some number, then this number is what we want. In the theory of algorithms, this type of result is called partial correctness. It leaves open the question of actually making this process produce some result. We would like to identify situations when the iteration converges. We start by interpreting our iteration as seen in the arrow-type picture.

We start with some  $x_0$  in the domain and it get sent to  $x_1$  in the co-domain on the right. However, to obtain  $x_2$  we need to substitute  $x_1$  into  $\varphi$ , so we have to move  $x_1$  identically back into the domain. Only then we can send it again, this time to  $x_2$ , and the whole process repeats itself.

Now that we understand how iteration works, we will revisit the examples that we started with.

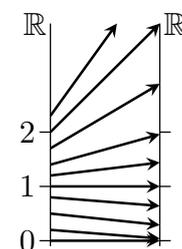


Here we see the action of  $\varphi(x) = \sqrt{x}$ . Imagine that we start with some  $x_0$  between 0 and 1. We move to the right and the square root shifts us up, then we jump back left on the same level, then we go right and a bit up again, jump left, and as we keep doing this, we imagine that we follow something like a spiral winding up towards 1. Similarly, if we start such a spiral from  $x_0 > 1$ , then we will spiral down towards 1.



We would guess that if we start our iteration with any positive  $x_0$ , then this iteration takes us to the fixed point 1. If we start with  $x_0 = 0$ , we obtain the constant sequence  $x_k = 0$  that clearly converges to the fixed point 0.

The picture for  $\varphi(x) = x^2$  suggests that if we start iterating with  $x_0$  between 0 and 1, then the resulting sequence should go to 0. On the other hand, if we start with  $x_0 > 1$ , then the iteration starts spiralling up beyond control.



It would seem that the only way to reach the fixed point 1 is the constant iteration  $x_k = 1$ .

We are ready to formulate some intuitive conclusions. If we want the iteration to take us successfully to a certain fixed point  $x_f$ , then we need for the function arrows to somehow point toward  $x_f$ .

We would like to express this idea mathematically. The fact that arrows somehow gravitate towards each other means that the distance between the outcomes should be smaller than the distance between starting points. Mathematically,

$$|\varphi(x) - \varphi(y)| < |x - y|.$$

This is not a bad idea, but one has to be careful. We surely saw in analysis that a limit process can turn sharp inequalities into equality. In this particular case, if we just asked for this property, it could happen that depending where we look, the quantities on the left and right can get arbitrarily close to each other, which would cause troubles. We therefore need to be more strict and insist that the decrease in mutual distances when being sent by a mapping has some minimal size.

**Definition 20a.3.**  
 Let  $\varphi$  be a function on an interval  $I$ . We say that it is **contractive** there, or that it is a **contraction**, if there exists  $q < 1$  such that for all  $x, y \in I$  we have

$$|\varphi(x) - \varphi(y)| \leq q \cdot |x - y|.$$

The constant  $q$  serves as a separation. We can see this better if we rewrite this condition in the following way:

$$\frac{|\varphi(x) - \varphi(y)|}{|x - y|} \leq q.$$

If we just asked for  $\frac{|\varphi(x) - \varphi(y)|}{|x - y|} < 1$ , then it could happen that the ratio approaches 1 arbitrarily close, which would not do. We use  $q < 1$  to keep this ratio away from 1 by at least some positive distance.

It turns out that this condition is just what was needed.

**Theorem 20a.4.** (Banach's fixed-point theorem)

Let  $\varphi$  be a contractive function with coefficient  $q$  on a closed bounded interval  $I$  such that  $\varphi[I] \subseteq I$ .

Then there exists exactly one solution  $x_f$  of the equation  $\varphi(x) = x$  in  $I$ . Moreover, for all choices of  $x_0 \in I$  the sequence given by  $x_{k+1} = \varphi(x_k)$  converges to  $x_f$  and satisfies

$$|x_f - x_{k+1}| \leq q|x_f - x_k| \quad \text{and} \quad |x_f - x_{k+1}| \leq \frac{q}{1-q}|x_{k+1} - x_k|.$$

This theorem supplies us with everything that we may ask for. We get convergence for any starting point  $x_0$ , and also an interesting information. The first inequality actually compares errors of approximations  $x_k$ . It reads  $E_{k+1} \leq c|E_k|$ , so the method of fixed point iteration is of order 1. This does not look too good, but as we will see below, this can be significantly improved.

The second inequality shows that  $|E_{k+1}| \leq c|x_{k+1} - x_k|$ , so we have a connection between the absolute difference (a popular stopping condition) and the error of approximation. As usual, we do not always have the information needed for a practical application of these formulas, but it is a nice thing to have anyway.

The Banach fixed point theorem is actually a very strong result, because it works not just for functions, but in general for mappings between metric spaces. We commented before that the notion of fixed point is very general, and this theorem shows that this idea can be taken to its full fruition.

**Proof:** We already proved that for any continuous function  $\varphi: I \mapsto I$  there must be some fixed point  $x_f$  in  $I$ .

Could there be more? Assume that we get fixed points  $x_f$  and  $y_f$  in  $I$ . Since they are fixed points, we can replace  $x_f$  with  $\varphi(x_f)$  and similarly with  $y_f$ , then we use the fact that  $\varphi$  is a contraction. We obtain

$$|x_f - y_f| = |\varphi(x_f) - \varphi(y_f)| \leq q|x_f - y_f|.$$

Rewriting this inequality we get

$$0 \leq q|x_f - y_f| - |x_f - y_f| = (q - 1)|x_f - y_f|.$$

The contraction assumption means that  $q < 1$ , so the number  $q - 1$  is negative. We divide the inequality:

$$0 \geq |x_f - y_f|.$$

However, as an absolute value, the expression on the right can never be negative, so  $|x_f - y_f| = 0$ , that is,  $x_f = y_f$ . There can be only one fixed point.

Now take some  $x_0 \in I$  and consider the sequence  $x_k$  generated by the iteration. Using the replacement  $x_f = \varphi(x_f)$  and the definition of contraction we get

$$|x_f - x_{k+1}| = |\varphi(x_f) - \varphi(x_k)| \leq q|x_f - x_k|, \quad (*)$$

which proves the first inequality in the statement.

It is also a recursive formula that can be applied repeatedly.

$$\begin{aligned} |x_f - x_k| &\leq q|x_f - x_{k-1}| \leq q \cdot q|x_f - x_{k-2}| \leq \dots \\ &\dots \leq q^k|x_f - x_0|. \end{aligned}$$

Since  $|q| < 1$ , the geometric sequence  $q^k$  tends to 0, therefore  $|x_f - x_k| \rightarrow 0$ , that is,  $x_k \rightarrow x_f$ .

Finally, we return to the inequality (\*):

$$\begin{aligned} |x_f - x_{k+1}| &\leq q|x_f - x_k| = q|x_f - x_{k+1} + x_{k+1} - x_k| \\ &\leq q|x_f - x_{k+1}| + q|x_{k+1} - x_k|. \end{aligned}$$

We subtract  $q|x_f - x_{k+1}|$  from both sides, rewrite the inequality and obtain

$$(1 - q)|x_f - x_{k+1}| \leq q|x_{k+1} - x_k|$$

$$\implies |x_f - x_{k+1}| \leq \frac{q}{1 - q}|x_{k+1} - x_k|$$

as claimed. □

It would be nice if we could recognize contractions easily. There is such a tool.

**Theorem 20a.5.**

Assume that function  $\varphi$  defined on an interval  $I$  has a continuous derivative on the interior  $I^O$  of  $I$ .

If there is  $q < 1$  such that  $|\varphi'(t)| \leq q$  on  $I^O$ , then  $\varphi$  is a contraction on  $I$  with coefficient  $q$ .

**Proof:** Take any  $x, y \in I$ , for simplicity we may assume that  $x < y$ . Then  $[x, y] \subseteq I$  and  $(x, y) \subseteq I^O$ , so  $\varphi$  is continuous on the former interval and differentiable on the latter. In other words, we can apply the Mean Value theorem on  $[x, y]$  to learn that

$$\frac{\varphi(x) - \varphi(y)}{x - y} = \varphi'(\xi).$$

for some  $\xi \in (x, y)$ . Applying our assumption we obtain

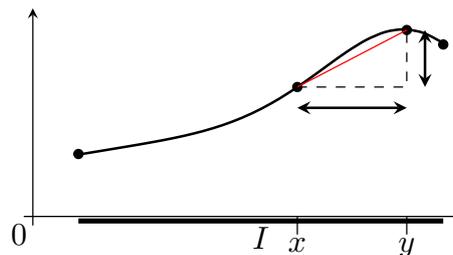
$$\frac{|\varphi(x) - \varphi(y)|}{|x - y|} = |\varphi'(\xi)| \leq q$$

and the proof is complete. □

This brings us back to the second point of view that we tried. How do contraction look in the usual graph point of view of real functions? We need to control the ratios

$$\frac{|\varphi(x) - \varphi(y)|}{|x - y|}$$

that we know well:



A function  $\varphi$  is a contraction on an interval  $I$  if all the secant lines as in the picture have slopes smaller (by some definite separation) than 1. In short, the function should be rather flat. As we know well, this can be controlled by the derivative, so the above result makes sense.

Returning to our examples, the graph of  $\sqrt{x}$  is flat around 1, so we are not surprised to see contraction working there. On the other hand, the graph of  $x^2$  grows fast around 1, that explains a lot.

However, note that the Banach contraction theorem is just an implication. The success or failure of iteration owns a lot to geometric configuration of the graph. To appreciate this we now look closer how the iteration actually appears in a graph of a function.

**Example 20a.d:** Consider the equation  $\cos(x) = x$ . The natural interval  $I$  is  $[-1, 1]$ , then  $\varphi(x) = \cos(x)$  indeed maps  $I$  into  $I$ . Is it a contraction there?

We check on the derivative:  $|\varphi(x)| = |\sin(x)|$ . Since the sine function is increasing on  $[-1, 1]$ , we get an upper bound  $|\varphi(x)| \leq \sin(1) = q$ . Since this  $q$  is smaller than 1, the mapping  $\cos(x)$  is a contraction on  $[-1, 1]$ . Therefore there is some fixed point  $x_f$  in  $[-1, 1]$  and for every  $x_0 \in [-1, 1]$ , the plain iteration produces a sequence that converges to  $x_f$ .

Note that if we just choose any  $x_0 \in \mathbb{R}$ , then  $x_1 = \cos(x_0) \in [-1, 1]$ , and we can think of the process that follows as if it started there. Thus the resulting sequence also necessarily converges to  $x_f$ . We conclude that the plain iteration also works if we consider  $\cos(x): \mathbb{R} \mapsto \mathbb{R}$ , although it is not a contraction there. Indeed, if we take  $x, y$  closer and closer to  $\frac{\pi}{2}$ , then the ratio  $\frac{|\varphi(x) - \varphi(y)|}{|x - y|}$  approaches 1 arbitrarily close.

△

Similarly, we know that the derivative  $[\sqrt{x}]'$  approaches infinity as  $x \rightarrow 0^+$ , so the function  $\sqrt{x}$  is definitely not a contraction on intervals  $[a, b]$  for  $b > 1$  and  $a$  close to zero, but as we observed, the plain iteration produces sequences that tend to the fixed point 1 on such sets.

We see that the contraction test is not perfect, but it is the best tool that we have if we want to guess whether iteration has a good chance of converging. We check on the derivative, and if it is small, we know that things look good. But we also know that even if the derivative may be large, the iteration is still worth trying, because we may get lucky.

We conclude this introductory section by revisiting the problem of order. The inequalities that we obtained suggest that the flatter the function (the smaller the derivative), the better behavior we can expect from iteration. The smallest derivative is zero, then the function is constant, which is very flat indeed.

**Theorem 20a.6.**

Let  $\varphi$  be a function that is at least twice differentiable on some neighborhood of its fixed point  $x_f$ . If  $\varphi'(x_f) = 0$ , then for every sequence  $\{x_f\}$  generated by (plain) iteration that converges to  $x_f$  there is  $c > 0$  such that

$$|x_f - x_{k+1}| \leq c|x_f - x_k|^2.$$

This means that the iteration then behaves as if it were of order 2, which is very nice, indeed. We can also see from the proof that if, moreover,  $\varphi''(x_f) = 0$ , then the iteration behaves as a method of order three. However, while the former can be often achieved using the relaxation approach described below, with the second derivative we can only hope for a lucky coincidence.

## 20b. Relaxation

Relaxation is a certain procedure that can be used to (try to) improve convergence of iteration. The motivation is based on our observations above. Given an equation of the form  $\varphi(x) = x$ , we would appreciate if the function  $\varphi$  was flat. We can flatten it by multiplying that equation using some (small but) non-zero number traditionally denoted  $\lambda$ :

$$\lambda\varphi(x) = \lambda x.$$

That looks good, but unfortunately this is no longer a problem of the right type. We need to see  $x$  on the right. Dividing the equation by  $\lambda$  would fix this, but that would get us back to square one. Instead we will try another way to fix problems. There is a certain quantity missing on the right, so we will add it:

$$\begin{aligned} \lambda\varphi(x) + (1 - \lambda)x &= \lambda x + (1 - \lambda)x \\ \implies \lambda\varphi(x) + (1 - \lambda)x &= x. \end{aligned}$$

We obtained a new problem of the right type. Since all the operations were reversible (we never take  $\lambda = 0$ ), this new problem has exactly the same solutions as before. We are in fact asking for

fixed points of a new function

$$\varphi_\lambda(x) = \lambda\varphi(x) + (1 - \lambda)x,$$

and by our observation, it has the same fixed points as the original  $\varphi(x)$ .

This approach is called relaxation and it is an idea that can be applied also in other settings. For us the important thing is that we can apply our usual iterative procedure to this new problem, and iterate with  $\varphi_\lambda(x)$ .

What are our expectations here? We expected the function  $\lambda\varphi(x)$  to be flatter, but now we add another function  $(1 - \lambda)x$  to it, which changes the situation. There are two interesting interpretations. Both start with the idea of a weighted average.

Given two quantities  $A, B$ , we find the average of them using the formula  $\frac{1}{2}A + \frac{1}{2}B$ . It combines the two influences in such a way that both have the same impact. Sometimes we may want to consider one of the two influences more important, but to balance things out, we then have to downrate the other. Mathematically, we consider expressions of the form  $\alpha A + \beta B$ , where we require that  $\alpha + \beta = 1$ . This is called a weighted average. We can use the restriction to get rid of one parameter and write such a weighted average as  $\alpha A + (1 - \alpha)B$ . We can imagine that  $\alpha$  is a slider that moves our attention from one to another in varying degrees. Of course, we could also use the formula  $(1 - \beta)A + \beta B$ .

Now we are ready to return to our relaxed iteration. It uses the following formula:

$$x_{k+1} = \lambda\varphi(x_k) + (1 - \lambda)x_k.$$

We can see this as a weighted average of two iterations and  $\lambda$  is an indicator of our confidence in the original iteration. If we are happy with it, we can simply put  $\lambda = 1$  and then  $\varphi_\lambda(x) = \varphi(x)$ , that is, we iterate  $x_{k+1} = \varphi(x_k)$ .

The other iteration in the mix is  $x_{k+1} = x_k$  and it always produces constant and hence convergent sequence. So if we are not quite happy with the convergence of the original iteration, we can put some weight (or more weight) on this perfectly convergent iteration. However, we should not overdo it, because for  $\lambda = 0$  we get the convergent procedure  $x_{k+1} = x_k$  that has, unfortunately, nothing in common with our original problem.

**Example 20b.a:** We return to our equation  $\cos(x) = x$ . We will try to find an approximation of its solution with precision  $\varepsilon = 0.0001$ . To have a fair comparison, we will always use  $x_0 = 1$  and the absolute difference stopping condition with this given  $\varepsilon$ .

To get some benchmark, we rewrite our equation as  $\cos(x) - x = 0$  and apply the Newton method (of order two). It found its result after  $N = 4$  iterations.

Now we try the plain fixed point iteration  $x_{k+1} = \cos(x_k)$ , and the procedure needed  $N = 24$  iteration for comparable result. This was to be expected from a method of general order one.

Relaxed iteration uses the formula  $x_{k+1} = \lambda \cos(x_k) + (1 - \lambda)x_k$ . We try several values for  $\lambda$ :

$\lambda =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3
$N =$	14	9	6	4	7	10	14

This is a fairly typical picture. We see that the rate of convergence of iteration can significantly improve with the right relaxation, and when the relaxation parameter  $\lambda$  is about 0.6, iteration becomes comparable to the Newton method. Can it get even better? Trying values like 0.61 we find that not really, the four iteration benchmark is the best. This is also fairly typical, although there are cases when the relaxed iteration can even beat the Newton method.

△

### Algorithm 20b.1.

⟨relaxation for fixed point iteration⟩

Given: Function  $\varphi$ .

0. Define  $\varphi_\lambda(x) = \lambda\varphi(x) + (1 - \lambda)x$ , where  $\lambda$  is the relaxation parameter.

1. Choose some value for  $\lambda$ , and proceed with the iteration method applied to this  $\varphi_\lambda$ .
2. If you are happy with the way iterations go, complete the iteration and obtain  $x_k$ , an approximation of fixed point of  $\varphi$ .

Otherwise repeat **1.** with different  $\lambda$ .

△

Now why would anyone want to do that, given that after the first try we already have an approximation of our root? There is a situation where relaxation might help. Imagine that we repeatedly look for roots of functions that are very similar. Then it makes sense to expect (or hope) that if some relaxation parameter  $\lambda$  works for one, it will also work for others. So every time we look for a root, we try a different value of  $\lambda$ , and after a while we find what relaxation works well for the problems of similar type that will come afterwards.

If the function  $\varphi(x)$  is given by a formula, we can also try to determine a suitable relaxation parameter  $\lambda$  before we actually start iterating. Recall that the general aim is to make the function

$$\varphi_\lambda(x) = \lambda\varphi(x) + (1 - \lambda)x$$

very flat, to make its derivative as small as possible. The smallest derivative is zero, which leads to a natural requirement that  $\varphi'_\lambda(x)$  should be zero. That is, we want

$$\lambda\varphi'(x) + (1 - \lambda) = 0.$$

But we have two unknowns there,  $\lambda$  and  $x$ , which brings us to the natural question: Where do we want our  $\varphi_\lambda$  to be flat? If we want this to be true around some point  $x_c$ , then we get

$$\lambda\varphi'(x_c) + (1 - \lambda) = 0 \implies \lambda = \frac{1}{1 - \varphi'(x_c)}.$$

The natural location  $x_c$  is the root itself, because then we will get, according to the theorem above, a quadratic convergence. However, we do not know the root.

We can often at least estimate the location of the root, then we can optimize  $\lambda$  for that location. Another interesting idea is to optimize  $\lambda$  at each step of our iteration; that is, for every  $x_k$  we use the  $\lambda_k$  that would make the next iterative step best possible, by making  $\varphi_\lambda$  flat right where we need it most. On the other hand, such dynamic adjustment of  $\lambda$  requires extra calculations, which could slow down the algorithm and thus offset gains.

**Example 20b.b:** We return to the problem  $\cos(x) = x$ . We can guess by plotting the graph that this root is not far from 0.7, so we ask for  $\varphi'_\lambda(0.7) = 0$ . Since

$$\varphi'_\lambda(x) = -\lambda \sin(x) + (1 - \lambda),$$

we deduce as above that the best relaxation parameter is

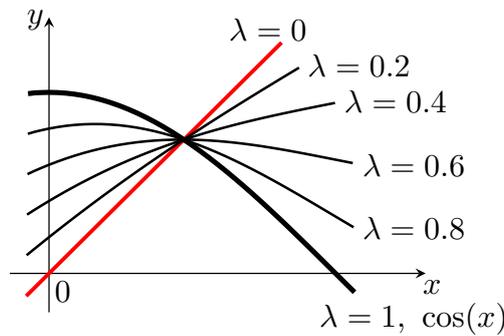
$$\lambda = \frac{1}{\sin(0.7) + 1} \approx 0.61.$$

This matches rather well the results of our experiments above.

△

We talked about the relaxed function  $\varphi_\lambda$  creating a weighted average of two iterations, but we can also view the function itself as a weighted average of two functions, and to take this one step further, we can imagine that the graph of  $\varphi_\lambda$  is an average of graphs of  $y = \varphi(x)$  and  $y = x$ . We can imagine that we have a slider, with  $\lambda = 1$  we see the graph of the function  $\varphi(x)$ , and as we move the slider to 0, the graph morphs gradually into the graph of  $f(x) = x$ .

In the following picture we show how this works with the graph of  $\varphi(x) = \cos(x)$ .

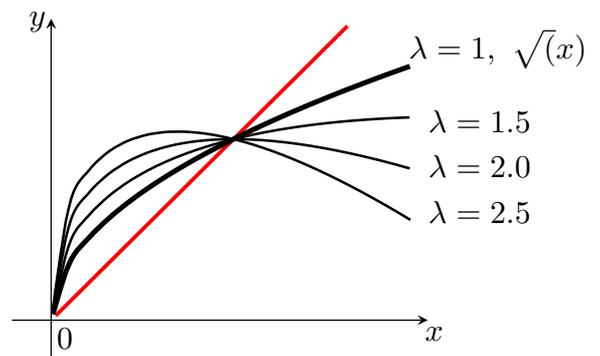


Note that for  $\lambda = 0.6$  we get a graph that is seemingly horizontal at the point of intersection, that is, at the fixed point. This, graphically, explains why this choice of parameter led to such a fast convergence.

When we ponder this example, we may get an insight that when a function  $\varphi$  is decreasing, then the best  $\lambda$  will be between 0 and 1. If the function  $\varphi$  is increasing but not as fast as  $y = x$ , then we would like to bend the graph somewhat downwards, which does not happen when we take intermediate versions of the compromise. We have to move in the direction of the graph of  $\varphi$  and then beyond, that is, the value of  $\lambda$  should be greater than 1, which is actually allowed.

**Example 20b.c:** Consider the function  $\varphi(x) = \sqrt{x}$ .

We observed earlier that iteration with a positive starting value should lead us to the fixed point  $x_f = 1$ . This pictures suggest that for  $\lambda$  around 2, the corresponding  $\varphi_\lambda$  looks rather flat around this fixed point. We will now make a series of experiments with common initial guess  $x_0 = 2$ , common tolerance and stopping condition, and different values of  $\lambda$  to test this. The following chart shows the number of iterations necessary for the algorithm to stop. We include also the case  $\lambda = 1$ , that is, the plain iteration.



$\lambda =$	0.8	1.0	1.2	1.4	1.6	1.8	2.0	2.2
$N =$	17	13	10	8	6	5	4	5

The results match our expectations remarkably well. As expected, trying  $\lambda < 1$  made actually the situation worse, as we actually bent the graph up more, so it got steeper.

Incidentally, the Newton method under the same conditions stops after  $N = 4$  iterations.

△

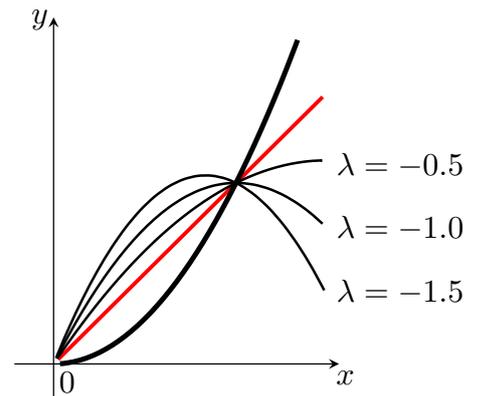
When a function grows faster than  $y = x$ , then we need to pull it towards  $y = x$  and then even beyond, suggesting that negative values of  $\lambda$  may be of help.

**Example 20b.d:** Consider the function  $\varphi(x) = x^2$ .

We observed earlier that iteration with a starting value greater than 1 leads to a divergent iteration. This pictures suggest that for  $\lambda$  around -1, the corresponding  $\varphi_\lambda$  looks rather flat around the fixed point  $x_f$ . We will again try to see how experiments fit with this expectation, we will use the initial guess  $x_0 = 1.5$ .

$\lambda =$	0.2	0.0	-0.2	-0.4	-0.6	-0.8	-1.0	-1.2
$N =$	$N$	$X$	30	15	9	7	5	7

For  $\lambda > 0$  we have divergence as expected. For  $\lambda = 0$  the iteration does not make sense, as then it has nothing to do with our problem, we simply work with the formula  $x_{k+1} = x_k$ .



It is the negative lambdas that are interesting, and again we see the behavior that we expected. And again, we matched the performance of the Newton method.

It should be noted that the function  $\varphi_\lambda$  is curving rather sharply down as we move to the right, and definitely does not look like a contraction there. Indeed, if we try the starting point  $x_0 = 2$ , then we get divergent iterations for  $\lambda < -1$ , and the best performance is  $N = 10$  for  $\lambda \approx -0.98$ , while the Newton method can still do with five iterations with this starting point. So this time we did not match this performance, but we did achieve success by saving iteration that originally diverged.

Another interesting feature of  $x_0 = 2$  is that when we set  $\lambda = -1$ , the iteration immediately jumps to  $x_f = 0$ , the other fixed point, and naturally stays there, never getting to  $x_f = 1$ .

△

Now we should have a good idea of what relaxation does for us. Before we move on, an important remark should be made. When we try to make  $\varphi_\lambda$  small somewhere, it only affects its shape around the point where we do it. If the iteration moves us elsewhere, we may be in trouble. This is related to the fact that making  $\varphi_\lambda$  into a contraction locally does not allow us to appeal to the conclusions of the Banach contraction theorem.

For that we would have to identify an interval  $I$  that is mapped by  $\varphi_\lambda$  into itself, which may be already quite a task given that the range most likely also depends on  $\lambda$ . Then we would have to show that  $\varphi_\lambda$  has a small derivative globally, on the whole interval  $I$ . This is a pretty serious work.

We will therefore use the local flatness as an indication of what to expect, but we should be aware that this is nothing more than that, and that the actual iteration may behave differently.

## 20c. Roots, fixed points and relaxation

Obviously, the root approach and the fixed-point approach are closely related. Indeed, every equation of the form  $f(x) = 0$  can be rewritten as  $f(x) + x = x$ , so the problem of finding a root of  $f$  can be replaced by the problem of finding a fixed point of  $\varphi(x) = f(x) + x$ . Conversely,  $\varphi(x) = x$  can be written as  $\varphi(x) - x = 0$ , so finding a fixed point of  $\varphi$  can be done by finding a root of  $f(x) = \varphi(x) - x$ . To sum it up, finding a root and finding a fixed point is just two facets of the same problem.

Thus, given some algebraic equation, we can choose in which way we can approach it. The root approach offers high performance methods, on the other hand, the fixed-point approach seems somewhat more flexible.

This flexibility can be observed already when rewriting our equation. In particular, consider a root problem  $f(x) = 0$ . We changed it above into a fixed-point problem by adding  $x$  to each side. The corresponding iteration function  $\varphi(x) = f(x) + x$  will be considered the **standard transformation** in this book.

Some people actually prefer a slight modification. before adding  $x$  they multiply the original equation by  $-1$ , arriving at the iteration function  $\varphi(x) = x - f(x)$ . There isn't much to choose from between these two approaches, and statistically they are about equally efficient. I chose the former as the standard in order to complete a nice circle. Given an equation  $f(x) = 0$ , we create the fixed-point version as  $f(x) + x = x$ , and the natural transformation into a root problem mentioned above (subtracting  $x$ ) returns us back to where we started:  $f(x) = 0$ . But the version  $x - f(x)$  also has its charm.

Whichever way we decide to play it, in many cases this does not make much difference as we prefer to do something else than the standard anyway.

**Example 20c.a:** Consider our traditional test problem  $x^3 - x - 10 = 0$ . We will try to find the solution using the fixed point approach, with initial guess  $x_0 = 3$ .

a) Standard transformation: Adding  $x$  to each side we obtain

$$x^3 - 10 = x \implies \varphi(x) = x^3 - 10.$$

What can we expect from this iteration? We have  $\varphi'(x) = 3x^2$ , where should we investigate it? We find by bracketing that the root is between 2 and 3, perhaps a bit closer to 2 judging from function values, so we will look at  $\varphi$  around 2 to have nice numbers. We have  $\varphi'(2) = 12$ . This indicates a very steep function, definitely very far from being a contraction. We thus feel that the standard iteration is perhaps not the best approach here.

If we did not know the rough location of the root, it would make sense to at least look at the situation at the place where the iteration starts, in our case at  $x_0 = 3$ . We get  $\varphi'(3) = 27$ , this is even worse.

Numerical experiment confirms that the iteration diverges, and quite badly at that.

Can it be salvaged using relaxation? We could use the formula for optimal  $\lambda$  that was deduced above, my preference is to remember the idea and apply it to problems. Here we go.

Relaxation means that we use the iterative function

$$\varphi_\lambda(x) = \lambda(x^3 - 10) + (1 - \lambda)x.$$

Its derivative is  $3\lambda x^2 + (1 - \lambda) = 1 - \lambda(1 - 3x^2)$ , and we want this to be zero when  $x = 2$ . We get

$$\lambda = \frac{1}{1 - 3 \cdot 2^2} = -\frac{1}{11} \approx -0.09.$$

Experiments show that with this  $\lambda$  the relaxed iteration actually converges to the right root, and with our traditional testing tolerance  $\varepsilon = 0.001$  stops after  $N = 8$  iterations. For comparison, the Newton method can make it in four, but we are glad to be able to salvage this divergent situation here.

If we decide to optimize our  $\lambda$  based on the starting point  $x_0 = 3$ , for instance because we would not know the location of the root, we would get  $\lambda = -\frac{1}{26} \approx -0.28$ . The iteration converges, this time with  $N = 15$  iterations. This is not so surprising, we optimized  $\varphi_\lambda$  around 3, but the iteration moved elsewhere soon. Still, not knowing the location of the root, this would be the best we could do, and it did work.

However, there are other alternatives.

**b)** We can rewrite  $x^3 - x - 10 = 0$  as  $x^3 = x + 10$ , then  $x = \frac{x+10}{x^2}$  and we have a different fixed-point version of our question, this time  $\varphi(x) = \frac{x+10}{x^2}$ . What can we expect of it?

$\varphi'(x) = -\frac{x+20}{x^3}$ , we are interested in  $|\varphi'(3)| = \frac{23}{27}$ . This is less than 1, so the situation looks hopeful. We try, and it turns out that this iteration also diverges, and quite badly at that. What happened? We quickly move elsewhere, and there the situation is worse. In particular, note that  $|\varphi'(2)| = \frac{11}{4} = 2.75$ , so  $\varphi$  stops being a contraction as we move towards the root.

However, now the derivatives are not as bad as in the previous attempt, so this time relaxation could be more successful. We will work with

$$\varphi_\lambda(x) = \lambda \frac{x+10}{x^2} + (1 - \lambda)x,$$

and  $\varphi'_\lambda(3) = 0$  happens for

$$\lambda = \frac{1}{1 + \frac{3+20}{3^3}} = \frac{27}{50} = 0.54.$$

Now the iteration converges, and stops after  $N = 14$  iterations. Again, we saved the day.

By the way, now we know that the root is about 2.3. If we try to optimize  $\lambda$  for this location, we get  $\lambda \approx 0.35$ . Then the iteration stops after  $N = 4$ , and we matched the speed of the Newton method.

This version of  $\varphi(x)$  is a good opportunity to note that we would actually prefer not to work with it when we have a viable alternative, because the  $x^2$  in the denominator could lead to a division-by-zero error. The chances are small, but there will be other formulas with comparable or even better performance that do not have this problem.

**c)** We start with the previous rewrite  $x^3 = x + 10$ , but then simply write  $x = (x + 10)^{1/3}$  and this time this fixed-point problem uses  $\varphi(x) = (x + 10)^{1/3}$ .

We have  $\varphi'(x) = -\frac{1}{3}(x+10)^{-2/3}$ , in particular  $|\varphi'(3)| = \frac{1}{3}13^{-2/3} \approx 0.06$ . This is very close to zero, so we are very hopeful. This way of rewriting the problem seems great right out of the box.

Indeed, the plain iteration stops after  $N = 5$  steps, which is just one step short of perfection. Relaxation is probably not worth our time.

This makes the main point of this example: Usually we can transform the given equation into a fixed-point problem in many ways, and often there is one that works just great without any further tinkering. In other words, the way in which we rewrite the problem may have greater impact than our attempts to save an unpleasant iteration using relaxation.

So let's practice some more.

d) We can rewrite  $x^3 - x - 10 = 0$  as  $x^3 - x = 10$ , that is,  $x(x^2 - 1) = 10$ . Then we can write  $x = \frac{10}{x^2 - 1}$  and we have another version of fixed-point problem, this time  $\varphi(x) = \frac{10}{x^2 - 1}$ . We will be hoping that we never get  $x_k = \pm 1$  in our runs, but the chances are small.

$\varphi'(x) = -\frac{20x}{(x^2 - 1)^2}$ , we are interested in  $|\varphi'(3)| = \frac{15}{16}$ . This is less than 1, but only barely. We will try a run.

Interestingly, after a while the iteration started to flip between two distinct values (about  $-10.099$  and  $0.099$ ) with just tiny changes.

Being so close to convergence just asks for relaxation. The optimal  $\lambda$  to make

$$\varphi_\lambda(x) = \lambda \frac{10}{x^2 - 1} + (1 - \lambda)x$$

flat around  $x_0 = 3$  is  $\lambda = \frac{16}{31} \approx 0.52$ .

Now the iteration converges, and stops after  $N = 37$ , which is quite a lot, this happens. Trying  $0.6$  we actually get a divergent oscillation, so we are playing it close to the edge here. On the other hand, smaller lambdas improve the runs. Optimizing around  $x = 2.3$  yields  $\lambda \approx 0.29$  and a run of respectable  $N = 5$  steps.

e) Starting with  $x(x^2 - 1) = 10$  again we can go to  $x^2 - 1 = \frac{10}{x}$ , then to  $x^2 = 1 + \frac{10}{x}$  and finally to  $x = \sqrt{1 + \frac{10}{x}}$ .

The iterative function  $\varphi(x) = \sqrt{1 + \frac{10}{x}}$  now features a double jeopardy, apart from division by zero (which will hopefully never happen) we now also have to worry about a negative argument under the root.

We have  $\varphi'(x) = \frac{-5}{x^2 \sqrt{1 + \frac{10}{x}}}$ , and around our initial guess we have  $|\varphi'(3)| = \frac{5\sqrt{3}}{9\sqrt{13}} \approx 0.27$ . This is significantly less than 1, so we are hopeful. Just to be on the safe side we check on what is happening towards the root:  $|\varphi'(2)| = \frac{5}{4\sqrt{6}} \approx 0.51 < 1$ . We are really hopeful.

The plain iteration converged after  $N = 12$  steps, which is pretty good for an out-of-the-box iteration.

To finish this off, we optimize the shape of

$$\varphi_\lambda(x) = \lambda \sqrt{\frac{10}{x} + 1} + (1 - \lambda)x$$

around  $x_0 = 3$ , obtaining  $\lambda \approx 0.79$ . The corresponding relaxed iteration stops after  $N = 5$  steps, which is almost the best one can get. Optimizing for  $x = 2.3$  we could probably improve this to  $N = 4$ , we leave it to the reader.

△

As we just saw, given an algebraic equation, we can transform it into a fixed-point problem in many ways, some good, some not so good, and often we can significantly improve our runs by relaxation.

## 20c.1 Some connections

We will conclude this chapter with exploration of relaxation as applied to root-finding problems.

We start with a problem of the form  $f(x) = 0$ . Applying the standard approach, we arrive at fixed-point iteration with  $\varphi(x) = f(x) + x$ . How would a relaxed iteration look like then?

$$\begin{aligned}\varphi_\lambda(x) &= \lambda\varphi(x) + (1 - \lambda)x = \lambda(f(x) + x) + (1 - \lambda)x \\ &= \lambda f(x) + x.\end{aligned}$$

This is a very nice formula.

Now we try it differently. We may prefer not to remember formulas, but understand the ideas instead. The relaxation went as follows: We started with the original fixed-point equation and multiplied it by  $\lambda$ , then adjusted. Let us try the same process with the root equation:

$$\begin{aligned}f(x) &= 0 \\ \lambda f(x) &= 0 \quad / \quad + x \\ \lambda f(x) + x &= x\end{aligned}$$

We obtained a fixed point problem with  $\varphi_\lambda(x) = \lambda f(x) + x$ , exactly as before. So in the end it does not matter which approach we use, we end up with the same situation.

We may notice that the alternative standard transformation to a fixed-point problem  $\varphi(x) = x - f(x)$  corresponds to our relaxation with  $\lambda = -1$ . But more can be done. How about this.

We start with the equation  $f(x) = 0$ , and now we divide it by  $-f'(x)$ , assuming that it is not zero. We get

$$\begin{aligned}-\frac{f(x)}{f'(x)} &= 0 \quad / \quad + x \\ x - \frac{f(x)}{f'(x)} &= x.\end{aligned}$$

We have  $\varphi(x) = x - \frac{f(x)}{f'(x)}$ . If we set up iteration for this approach, we obtain the iterating formula  $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$ . Looks familiar? It should, it is the Newton formula from chapter 19. Wow! Turns out the Newton formula is just a special case of the fixed-point approach to finding roots, and how well it worked for us.

The last interesting point: Consider the standard relaxed fixed-point iteration, that is, with

$$\varphi_\lambda(x) = \lambda f(x) + x.$$

Then  $\varphi'(x) = \lambda f'(x) + 1$ . If we want to optimize around a certain point  $x_c$ , we want to have  $\lambda f'(x_c) + 1 = 0$ , that is,  $\lambda = -\frac{1}{f'(x_c)}$ .

We now return to one idea that we entertained before but did not pursue it. What if we try to optimize the relaxation parameter at every step of iteration, thus making sure that all our iteration steps are as good as they could be? For  $x_k$  we get the optimal parameter  $\lambda_k = -\frac{1}{f'(x_k)}$ , our iteration then proceeds like this:

$$\begin{aligned}x_{k+1} &= \varphi_{\lambda_k}(x_k) = \lambda_k f(x_k) + x_k \\ &= -\frac{1}{f'(x_k)} f(x_k) + x_k = x_k - \frac{f(x_k)}{f'(x_k)}.\end{aligned}$$

We again obtain the Newton formula. We conclude that the Newton method is actually the standard fixed point iteration, relaxed with relaxation parameter optimized at every step to provide the best possible performance. No wonder the Newton method is so fast.

Proof of Newton

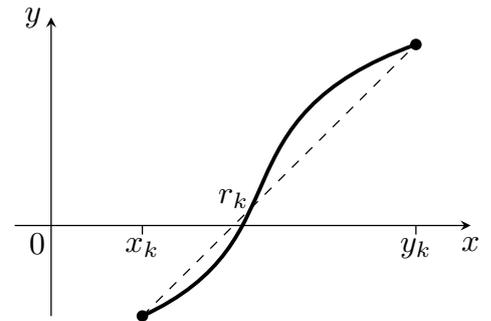
Roots of higher multiplicity

Faster methods-Improving Newton

Faster methods-Improving Secant

aster methods-Improving Bisection

Before we move to faster methods, it is worth our time to look at one failed attempt on improvement. If we know the values of  $f$  at  $x_k$  and  $y_k$ , we can make an educated guess where the root lies by connecting the known endpoints with a straight line and find its intersection with the  $x$ -axis. This estimate is then used in place of midpoint in bisection.



### Algorithm 20c.2.

⟨regula falsi method for finding root of a function  $f$ ⟩

Given: a function  $f$  continuous on interval  $[a, b]$  and a tolerance  $\varepsilon$ .

Assumption:  $f(a)$  and  $f(b)$  have opposite signs.

0. Set  $x_0 = a$ ,  $y_0 = b$ . Let  $k = 0$ .

1. Assumption:  $f(x_k)$  and  $f(y_k)$  have opposite signs.

$$\text{Let } r_k = \frac{x_k f(y_k) - y_k f(x_k)}{f(y_k) - f(x_k)}.$$

2. If  $|r_k - r_{k-1}| < \varepsilon$  or  $|f(r_k)| < \varepsilon$  then algorithm stops, output is  $r_k$ . Otherwise:

If  $f(x_k)$  and  $f(r_k)$  have opposite signs, set  $x_{k+1} = x_k$ ,  $y_{k+1} = r_k$ ,  
increase  $k$  by one and go back to step 1.

If  $f(r_k)$  and  $f(y_k)$  have opposite signs, set  $x_{k+1} = r_k$ ,  $y_{k+1} = y_k$ ,  
increase  $k$  by one and go back to step 1.

△

How good is this method? While it is true that these  $m_k$  are usually better estimates for the root than midpoints, surprisingly, this algorithm is not on average faster than the bisection point. Depending on the shape of the graph, it may run significantly longer than bisection before achieving the same precision. Or it can be quicker, it can play out both ways, but on average it evens out. If we want something better, we need to try something else.

Quadratic interpolation.

Consider points  $x_1, x_2, x_3$  and the corresponding function values  $f_1 = f(x_1)$ ,  $f_2 = f(x_2)$ ,  $f_3 = f(x_3)$ . The Lagrange interpolating polynomial then has the form

$$\begin{aligned} t \mapsto & t^2 \left( \frac{f_1}{(x_1 - x_2)(x_1 - x_3)} + \frac{f_2}{(x_2 - x_1)(x_2 - x_3)} + \frac{f_3}{(x_3 - x_1)(x_3 - x_2)} \right) \\ & + t \left( \frac{-f_1(x_2 + x_3)}{(x_1 - x_2)(x_1 - x_3)} + \frac{-f_2(x_1 + x_3)}{(x_2 - x_1)(x_2 - x_3)} + \frac{-f_3(x_1 + x_2)}{(x_3 - x_1)(x_3 - x_2)} \right) \\ & + \left( \frac{f_1 x_2 x_3}{(x_1 - x_2)(x_1 - x_3)} + \frac{f_2 x_1 x_3}{(x_2 - x_1)(x_2 - x_3)} + \frac{f_3 x_1 x_2}{(x_3 - x_1)(x_3 - x_2)} \right). \end{aligned}$$

We can write it as

$$t \mapsto \frac{1}{d}(at^2 + bt + c),$$

where

$$d = (x_1 - x_2)(x_1 - x_3)(x_2 - x_3);$$

$$a = f_1(x_2 - x_3) - f_2(x_1 - x_3) + f_3(x_1 - x_2)$$

$$b = -f_1(x_2 + x_3)(x_2 - x_3) + f_2(x_1 + x_3)(x_1 - x_3) - f_3(x_1 + x_2)(x_1 - x_2)$$

$$c = f_1 x_2 x_3 (x_2 - x_3) - f_2 x_1 x_3 (x_1 - x_3) + f_3 x_1 x_2 (x_1 - x_2).$$

Assuming that  $D = b^2 - 4ac$  is not negative, we can find the roots using the usual quadratic formula with  $a, b, c$ .

## 23. Solving systems of linear equations by elimination

Consider a system of linear equations

$$\begin{aligned}a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,m}x_m &= b_1 \\a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,m}x_m &= b_2 \\&\vdots = \vdots \\a_{n,1}x_1 + a_{n,2}x_2 + \cdots + a_{n,m}x_m &= b_n\end{aligned}$$

We assume that the reader knows the matrix notation  $A\vec{x} = \vec{b}$  of such a problem, where  $A = (a_{i,j})$  is the matrix of the system,  $\vec{b} \in \mathbb{R}^n$  is the vector of right hand sides and  $\vec{x} \in \mathbb{R}^m$  is the unknown.

In linear algebra we learn a general procedure for solving such a system, including the need to introduce parameters when such a system is underdetermined. However, in numerical analysis we do not really meet such systems, in scientific and engineering applications the matrix  $A$  is typically square and regular. This significantly simplifies the situation, in particular it means that such systems have unique solutions, which makes them a suitable target for numerical procedures. We will therefore follow the usual approach and assume that our matrices are square and regular.

### 23a. Solving systems of linear equation using Gaussian elimination

Students often learn the following procedure to solve systems of linear equations.

1. Form the extended matrix  $(A|\vec{b})$ .
2. Use elimination to turn it into the form  $(E_n|\vec{x}_0)$ .
3. The vector  $\vec{x}_0$  is the solution.

The second step was called the Gauss-Jordan elimination in the previous chapter 22, so we know how much work we have to do: asymptotically it is  $n^3$  operations.

This procedure works fine for us, but that was because we only did it by hand for small matrices. For larger ones there is a faster way.

Imagine that we use the cheaper Gaussian elimination that only costs about  $\frac{2}{3}n^3$  operations, doing something like this:

$$(A|\vec{b}) \mapsto (U|\vec{d}),$$

where  $U$  is an upper triangular matrix. What next? This matrix represents the following system.

$$\begin{aligned}u_{1,1}x_1 + u_{1,2}x_2 + \cdots + u_{1,n}x_n &= d_1 \\u_{2,2}x_2 + \cdots + u_{2,n}x_n &= d_2 \\&\vdots \\u_{n-2,n-2}x_{n-2} + u_{n-2,n-2}x_{n-1} + u_{n-2,n}x_n &= d_{n-2} \\u_{n-1,n-1}x_{n-1} + u_{n-1,n}x_n &= d_{n-1} \\u_{n,n}x_n &= d_n\end{aligned}$$

We can see that the last equation can be easily solved,  $x_n = \frac{d_n}{u_{n,n}}$ . Once we know  $x_n$ , the second last equation has only one unknown and we can solve for it,  $x_{n-1} = \frac{d_{n-1} - u_{n-1,n}x_n}{u_{n-1,n-1}}$ . This now allows us to solve the second last equation for  $x_{n-2}$  and so on, until we find  $x_1$ . We go from the last to the first, that's why it's called the **back substitution**.

#### Algorithm 23a.1.

⟨Solving an upper triangular system by back substitution⟩

Given: a system  $U\vec{x} = \vec{d}$ , where the matrix  $U$  is square, regular and upper triangular.

1. Find the solution  $\vec{x}_0$  using the formulas

$$\begin{aligned}x_n &= \frac{d_n}{u_{n,n}} \\x_{n-1} &= \frac{d_{n-1} - u_{n-1,n}x_n}{u_{n-1,n-1}} \\x_{n-2} &= \frac{d_{n-2} - u_{n-2,n}x_n - u_{n-2,n-1}x_{n-1}}{u_{n-2,n-2}} \\&\vdots \\x_1 &= \frac{d_1 - u_{1,n}x_n - u_{1,n-1}x_{n-1} - \cdots - u_{1,2}x_2}{u_{1,1}}\end{aligned}$$

In general,

$$x_k = \frac{1}{u_{k,k}} \left( d_k - \sum_{i=k+1}^n u_{k,i}x_i \right).$$

△

How much work does it take? The general formula shows clearly how many operations it takes to evaluate  $x_k$ , there are  $n - k$  multiplications, one less addition, but then we subtract the sum so we are back at  $n - k$ , and one division. We sum it all up,

$$\sum_{k=1}^n (2(n - k) + 1) = -2 \sum_{k=1}^n k + (2n + 1) \sum_{k=1}^n 1 = -2 \frac{1}{2}n(n + 1) + (2n + 1)n = n^2.$$

Before we make it into a theorem, we observe that a similar procedure would work with a lower triangular matrix  $L$ . The system  $L\vec{x} = \vec{d}$  then has the form

$$\begin{aligned}l_{1,1}x_1 &= d_1 \\l_{2,1}x_1 + l_{2,2}x_2 &= d_2 \\&\vdots \\l_{n,1}x_1 + l_{n,2}x_2 + \cdots + l_{n,n}x_n &= d_n\end{aligned}$$

and we easily solve for  $x_1$ , then  $x_2$  and so on, this is the **forward substitution**.

### Algorithm 23a.2.

⟨Solving a lower triangular system by forward substitution⟩

Given: A system  $L\vec{x} = \vec{d}$ , where the matrix  $L$  is square, regular and lower triangular.

1. Find the solution  $\vec{x}_0$  using the formulas

$$\begin{aligned}x_1 &= \frac{d_1}{l_{1,1}} \\x_2 &= \frac{d_2 - l_{2,1}x_1}{l_{2,2}} \\x_3 &= \frac{d_3 - l_{3,1}x_1 - l_{3,2}x_2}{l_{3,3}} \\&\vdots \\x_n &= \frac{d_n - l_{n,1}x_1 - l_{n,2}x_2 - \cdots - l_{n,n-1}x_{n-1}}{l_{n,n}}\end{aligned}$$

In general,

$$x_k = \frac{1}{l_{k,k}} \left( d_k - \sum_{i=1}^{k-1} u_{k,i} x_i \right).$$

△

Number of operations is calculated just like for the back substitution.

**Fact 23a.3.**

A system  $A\vec{x} = \vec{b}$  whose matrix is upper triangular, resp. lower triangular can be solved using back substitution, resp. forward substitution with computational complexity  $n^2$ .

Since  $n^2$  is negligible compared to the cost  $\frac{2}{3}n^3$  of Gaussian elimination, for larger matrices it is better not to use the Gauss-Jordan elimination (with complexity  $n^3$ ) and instead use the following algorithm.

**Algorithm 23a.4.**

⟨solving systems of linear equations by elimination⟩

Given: a system  $A\vec{x} = \vec{b}$ , where  $A$  is a square regular matrix.

1. Use Gaussian elimination to change the extended matrix  $(A|\vec{b})$  into an upper triangular matrix  $(U|\vec{d})$ .

2. Solve the system  $U\vec{x} = \vec{d}$  using back substitution.

△

Note that back and forward substitution is not just less work, but it is also very simple to implement, essentially it is just two nested loops.

**Example 23a.a:** We return to example 22a.a.

$$\begin{pmatrix} 2 & -2 & -6 & 2 \\ 1 & 3 & 0 & 1 \\ 2 & -8 & -9 & 3 \end{pmatrix}.$$

We can see it as the matrix of the system

$$2x - 2y - 6z = 2$$

$$x + 3y = 1$$

$$2x - 8y - 9z = 3.$$

In example we in fact found the solution, namely  $x = \frac{5}{2}$ ,  $y = -\frac{1}{2}$ ,  $z = \frac{2}{3}$ . Now we try the new way of solving this system.

In example 22a.a we reduced the matrix as follows:

$$\left( \begin{array}{ccc|c} 2 & -2 & -6 & 2 \\ 1 & 3 & 0 & 1 \\ 2 & -8 & -9 & 3 \end{array} \right) \sim \left( \begin{array}{ccc|c} 2 & -2 & -6 & 2 \\ 0 & 4 & 3 & 0 \\ 0 & 0 & \frac{3}{2} & 1 \end{array} \right).$$

This corresponds to the system of equations

$$2x - 2y - 6z = 2$$

$$4y + 3z = 0$$

$$\frac{3}{2}z = 1.$$

From the third equation we obtain  $z = \frac{2}{3}$ . Substituting this into the second we obtain  $y = -\frac{3}{4}z = -\frac{1}{2}$ . Finally, substituting into the first equation we get  $x = 1 + y + 3z = 1 - \frac{1}{2} + 2 = \frac{5}{2}$ . We obtained the right answer.

△

**23a.5 Remark:** Note that similar reasoning can be used to find an inverse matrix. In order to find a matrix  $X$  such that  $AX = E_n$  we are actually simultaneously solving systems of linear equations  $A\vec{x}^j = \vec{e}_j$ , where by  $\vec{x}^j$  we now denoted the  $j$ th column of  $X$ . Thus we can get an inspiration from our previous work and suggest the following procedure.

1. First we use GE to change  $(A|E_n)$  into  $(U|D)$ , where  $U$  is upper triangular.
2. We now treat every column of matrix  $D$  as a right hand side vector of a system of linear equations and use back substitution to determine corresponding column of the matrix  $X = A^{-1}$ . Thus the general formula reads

$$x_{k,l} = \frac{1}{u_{k,k}} \left( d_{k,l} - \sum_{i=k+1}^n u_{k,i} x_{i,l} \right).$$

How many operations does it take? We use Theorem 22b.1 with  $c = n$  and see that this requires  $\frac{5}{3}n^3 + O(n^2)$  operations.

Back substitution requires  $n^2$  operations, but that is for only one column and we need to handle  $n$  of them. Thus altogether this brings another  $n^3$  operations for the grand total of  $\frac{8}{3}n^3 + O(n^2)$ . We saved  $\frac{1}{3}n^3$  operations compared to the GJE way, which is quite a bit.

It should be noted that if  $A$  itself is already triangular, then the number of operations is markedly smaller. It is easy to show that an inverse matrix to an upper trinagular matrix is again upper triangular (and an analogous statement applies to lower triangular matrices). Thus we need to find only half of the terms. We can find direct formulas for them. Indeed, the desired equality  $AX = E_n$  becomes for an upper triangular  $A$  the following system of equations:

$$\begin{aligned} a_{1,1}x_{1,1} &= 1, \\ a_{2,2}x_{2,2} &= 1, \quad a_{1,1}x_{1,2} + a_{1,2}x_{2,2} = 0 \\ a_{3,3}x_{3,3} &= 1, \quad a_{2,2}x_{2,3} + a_{2,3}x_{3,3} = 0, \quad a_{1,1}x_{1,3} + a_{1,2}x_{2,3} + a_{1,3}x_{3,3} = 0 \\ &\vdots \end{aligned}$$

Now we can go by rows with back substitution, obtaining

$$\begin{aligned} x_{1,1} &= \frac{1}{a_{1,1}}, \\ x_{2,2} &= \frac{1}{a_{2,2}}, \quad x_{1,2} = -\frac{1}{a_{1,1}}a_{1,2}x_{2,2} \\ x_{3,3} &= \frac{1}{a_{3,3}}, \quad x_{2,3} = -\frac{1}{a_{2,2}}a_{2,3}x_{3,3}, \quad x_{1,3} = -\frac{1}{a_{1,1}}(a_{1,2}x_{2,3} + a_{1,3}x_{3,3}) \\ &\vdots \\ x_{k,k} &= \frac{1}{a_{k,k}}, \quad x_{j,k} = -\frac{1}{a_{j,j}}(a_{j,j+1}x_{j+1,k} + a_{j,j+2}x_{j+2,k} + \cdots + a_{j,k-1}x_{k-1,k} + a_{j,k}x_{k,k}) \\ &\text{for } j \text{ from } k-1 \text{ to } j=1. \end{aligned}$$

Calculating  $x_{k,k}, x_{k-1,k}, \dots, x_{1,k}$ , that is, the  $k$ th column of  $X$ , requires

$$1 + 2 + 4 + \cdots + (2k-2) = 1 + 2 \sum_{j=1}^{k-1} j = 1 + (k-1)k$$

operations, the total for the whole matrix is

$$\sum_{k=1}^n [k^2 - k + 1] = \frac{1}{6}n(n+1)(2n+1) - \frac{1}{2}n(n+1) + n = \frac{1}{3}n^3 + \frac{2}{3}n.$$

We see that the complexity is  $\frac{1}{3}n^3 + O(n^3)$ , which is even less than GE; moreover, we have direct formulas.

△

### 23b. Solving systems of linear equations by LUP factorization

Sometimes there is a need to solve the same system repeatedly, with different right hand sides  $\vec{b}$ . If we know all of them at the start, we can extend the system matrix  $A$  by all of them and solve all those systems at the same time.

However, sometimes the right hand sides come one at a time. We definitely would not like to do  $\frac{2}{3}n^3$  operations every time a new  $\vec{b}$  comes up. Is there a way to do some work beforehand to save operations later? One possibility follows from simple algebra, because the solution is given by  $\vec{x} = A^{-1}\vec{b}$  (assuming that  $A$  is regular, but this usually works out in applications). Determining  $A^{-1}$  costs  $n^3$  operations, and when somebody comes up with a new  $\vec{b}$ , we just do  $\vec{x}_0 = A^{-1}\vec{b}$ , which will cost  $2n^2 + O(n)$  operations. Definitely better than repeating GE.

However, there is a better approach. Note that the steps made in elimination depend only on the matrix  $A$ , so with a new right hand side we would perform exactly the same row operations, just applying them to new vectors  $\vec{b}$ . If we could store the information about operations, with a new right hand side we would not have to do another  $\frac{2}{3}n^3$  operations, we would just apply all those row operations to numbers  $b_i$ , which is, asymptotically, about  $n^2$  operations.

What do we really need? After we do one run  $(A|\vec{b}) \mapsto (U|\vec{d})$ , we obviously need to remember  $U$  so that we can do the necessary back substitution again. We also need to know which row operations changed  $\vec{b}$  into  $\vec{d}$  so that we can repeat this with some new  $\vec{b}$ , and this information is contained in the constants  $l_{i,k}$  from the Gaussian elimination (see 22d.1). Indeed, the fact that  $l_{i,k} = a \neq 0$  means that we are supposed to subtract the  $k$ th row  $a$ -times from the  $i$ th row, so we just do it with entries in the vector  $\vec{b}$ . We just need to remember in which order we have to do those operations, and GE says that we should take  $k = 1, \dots, n$  in sequence and for each of them do operations corresponding to  $l_{i,k}$  for  $i > k$ .

Note that the numbers  $l_{i,k}$  do not store information about row exchanges. For now we will therefore restrict our attention only to matrices that can be reduced by elimination without any row switches; that is, at all stages we find a non-zero number at the pivot location.

**Example 23b.a:** We again return to example 22a.a, where we reduced the given matrix as

$$\left( \begin{array}{ccc|c} 2 & -2 & -6 & 2 \\ 1 & 3 & 0 & 1 \\ 2 & -8 & -9 & 3 \end{array} \right) \sim \left( \begin{array}{ccc|c} 2 & -2 & -6 & 2 \\ 0 & 4 & 3 & 0 \\ 0 & 0 & \frac{3}{2} & 1 \end{array} \right).$$

Scanning through that example we can see that we did not switch any rows during the elimination, so the whole process is captured in coefficients  $l_{2,1} = \frac{1}{2}$ ,  $l_{3,1} = 1$ , and  $l_{3,2} = -\frac{3}{2}$ .

Now imagine that we are asked to solve the system

$$\begin{aligned} 2x - 2y - 6z &= -2 \\ x + 3y &= -2 \\ 2x - 8y - 9z &= 1. \end{aligned}$$

Since the left-hand sides are exactly the same as in the original system from example 22a.a, reducing

the new matrix

$$\begin{pmatrix} 2 & -2 & -6 & -2 \\ 1 & 3 & 0 & -2 \\ 2 & -8 & -9 & 1 \end{pmatrix}$$

would follow in the same way. We will therefore save time and just apply those operations to the column of right-hand sides. We have to be careful about the order. First we start with  $l_{i,1}$  and go through all possible  $i$  in sequence, then we apply  $l_{i,2}$ . We have to subtract the first entry  $\frac{1}{2}$  times from the second and once from the third. Then we have to add the new second entry  $\frac{3}{2}$  times to the third.

$$\begin{pmatrix} -2 \\ -2 \\ 1 \end{pmatrix} \sim \begin{pmatrix} -2 \\ -1 \\ 1 \end{pmatrix} \sim \begin{pmatrix} -2 \\ -1 \\ 3 \end{pmatrix} \sim \begin{pmatrix} -2 \\ -1 \\ \frac{3}{2} \end{pmatrix}.$$

This is the reduced right-hand side, we recycle the reduced left-hand sides and obtain the system

$$\begin{aligned} 2x - 2y - 6z &= -2 \\ 4y + 3z &= -1 \\ \frac{3}{2}z &= \frac{3}{2}. \end{aligned}$$

The back substitution then provides us with

$$z = 1, \quad y = \frac{1}{4}(-1 - z) = -1, \quad x = -1 + y + 3z = 1.$$

This is actually the correct solution for the new system. It seems that this idea works.

△

The numbers  $l_{i,k}$  just beg to be stored in a matrix. For the above example it would be

$$\begin{pmatrix} & & & \\ \frac{1}{2} & & & \\ 1 & -\frac{3}{2} & & \end{pmatrix}.$$

Now if somebody told me to code the fact that the  $k$ th row is important and I subtract it from the  $i$ th row, I'd have called it  $l_{k,i}$ . Why are indices the other way around? There are good reasons for this that will be soon revealed. For now we observe that when we perform an elimination and we suspect that we would have to work with the same system again, we would want to store the resulting upper triangular matrix  $U$  and the coefficients  $l_{i,k}$ . By a remarkable coincidence, these fit exactly to the place where  $U$  has zeros, so we can store both information in one handy matrix. Moreover, we can store these  $l_{i,k}$  on the go as we create zeros during the elimination, which is very convenient.

However, there is more. The matrix formed out of  $l_{i,k}$  is obviously unfinished, we can complete it by putting 1 on the diagonal and zeros above it, creating a lower-triangular matrix. Now comes the surprise.

**Theorem 23b.1.**

Let  $A$  be an  $n \times n$  matrix that can be reduced through elimination without row exchanges into an upper triangular matrix  $U$ , with row operations coded by  $l_{i,k}$ . Defining  $l_{i,i} = 1$  for all  $i = 1, \dots, n$  and  $l_{i,k} = 0$  for all  $1 \leq i < k \leq n$  we create a lower triangular matrix  $L$ . Then  $LU = A$ .

**Proof:** (outline) We know from linear algebra that when we have an  $n \times m$  matrix  $A$  and we want to subtract the  $k$  row  $l$  times from the row  $i$ , we can achieve this by multiplying  $A$  from the left by a matrix that is created by putting  $-l$  into the unit matrix  $E_n$  at position  $(i, k)$ :

$$\begin{array}{c}
 k \\
 \downarrow \\
 \begin{pmatrix}
 1 & 0 & 0 & \dots & 0 \\
 0 & 1 & 0 & & \\
 & & 1 & & \vdots \\
 & -l & & & \\
 0 & & & & 1
 \end{pmatrix} = \begin{pmatrix}
 a_{1,1} & a_{1,2} & \dots & a_{1,m} \\
 \vdots & & & \vdots \\
 a_{n,1} & a_{n,2} & \dots & a_{n,m}
 \end{pmatrix} = \begin{pmatrix}
 a_{1,1} & \overline{a_{1,2}} & \dots & a_{1,m} \\
 \vdots & \vdots & -l \times & \vdots \\
 a_{n,1} & \overline{a_{n,2}} & \dots & a_{n,m}
 \end{pmatrix}
 \end{array}$$

Such a matrix is called an elementary matrix.

If we denote the matrix corresponding to operation  $l_{i,k}$  as  $L_{i,k}$ , we can capture the whole elimination as

$$U = L_{n,n-1} \cdot L_{n,n-2} \cdot L_{n-1,n-2} \dots L_{3,2} \cdot L_{n,1} \dots L_{3,1} \cdot L_{2,1} A.$$

Now we multiply this equation from the left by inverses of the elementary matrices, and obtain

$$L_{2,1}^{-1} \cdot L_{3,1}^{-1} \dots L_{n,1}^{-1} \cdot L_{3,2}^{-1} \dots L_{n-1,n-2}^{-1} \cdot L_{n,n-2}^{-1} \cdot L_{n,n-1}^{-1} \cdot U = A.$$

It is not hard to show that the inverse matrix of an elementary matrix can be created very easily, we just change the sign at  $l$ . This means that while the matrix  $L_{i,k}$  has  $-l_{i,k}$  at position  $(i, k)$ , the inverse matrix  $L_{i,k}^{-1}$  has  $l_{i,k}$  at position.

In general, multiplying elementary matrices can change their contents in various ways. However, one can show that when we multiply them in exactly the order outlined above, then the entries  $l_{i,k}$  are added to those already there, so the whole product creates exactly the matrix  $L$  described in the statement. That is, the above formula reads

$$LU = A.$$

This concludes the outline. □

One popular sport in matrix theory is decompositions (or factorizations), when we express the given matrix as a product of special matrices. There are many useful factorizations, and the result above inspires us to make the following definition.

**Definition 23b.2.**  
 Consider a real  $n \times n$  matrix  $A$ . We say that  $n \times n$  matrices  $L, U$  are an **LU factorization** of  $A$  if  $U$  is an upper-triangular matrix,  $L$  is a lower-triangular matrix whose diagonal entries are all 1, and  $A = LU$ .

The above theorem has an immediate consequence.

**Corollary 23b.3.**  
 If Gaussian elimination without pivoting applied to  $A$  succeeds, then  $A$  admits an LU-factorization.  
 Moreover, the matrices  $L, U$  from Gaussian eliminations are the right ones.

Careful reading of the proof above shows that the relationship in fact goes two ways. If we have an LU-factorization  $A = LU$ , then  $l_{i,k}$  define an elimination process that reduces  $A$  to  $U$ . Since we know that for some matrices we are forced to switch rows during elimination, it follows that there are matrices for which there is no LU-factorization. On the other hand, if a singular matrix  $A$  does have an LU-factorization, then it has infinitely many of them. However, for a regular matrix there is only one LU-factorization possible.

**Example 23b.b:** We return again to example 22a.a. Putting together our results we easily check that

$$\begin{pmatrix} 2 & -2 & -6 \\ 1 & 3 & 0 \\ 2 & -8 & -9 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ 1 & -\frac{3}{2} & 1 \end{pmatrix} \cdot \begin{pmatrix} 2 & -2 & -6 \\ 0 & 4 & 3 \\ 0 & 0 & \frac{3}{2} \end{pmatrix}.$$

△

It should be noted that some authors do not require that the diagonal entries of  $L$  should be all one. This gives them more freedom, in particular we do not have uniqueness even for regular matrices. However, this is exactly what some other people do not like. The way we defined LU-factorization here seems the more popular choice.

The fact that elimination leads to an algebraic identity is not just a curio. Recall our main motivation: We wanted to save work when solving a system of equations repeatedly with new right-hand sides. We did find a way, but so far it is a programmer's solution: We have to redo a list of operations. The LU-factorization allows us to approach this problem using mathematical notation and familiar procedures.

To this end, consider a regular matrix  $A$  for which we know its LU-factorization  $A = LU$ . We want to solve the system  $A\vec{x} = \vec{b}$ . We substitute:  $(LU)\vec{x} = \vec{b}$ , that is,  $L(U\vec{x}) = \vec{b}$ .

We can denote  $\vec{y} = U\vec{x}$  and the system now reads  $L\vec{y} = \vec{b}$ . Since  $L$  is lower-triangular, we can solve this system using forward substitution at the cost of  $n^2$  operations.

Now we know  $\vec{y}$ , and  $U\vec{x} = \vec{y}$ , where  $U$  is upper triangular. This means that we can determine  $\vec{x}$  using back substitution, another  $n^2$  operations. Symbolically:

$$(L|\vec{b}) \mapsto \vec{y}, \quad (U|\vec{y}) \mapsto \vec{x}.$$

This process costs  $2n^2$  operations, which is much cheaper than using the gaussian elimination again.

We will not formalize this process yet, first we will address the problem that some matrices need not have an LU-factorization.

**Example 23b.c:** Consider  $A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$ . We have a nice candidate for the pivot in the first column, but without pivoting we cannot move it to the right place and the algorithm fails.

Now we try it algebraically. Is it possible to have factorization

$$A = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ l & 1 \end{pmatrix} \begin{pmatrix} a & b \\ 0 & d \end{pmatrix}?$$

The upper left corner of  $A$  forces  $a = 0$ , then for the lower left corner we obtain the equation  $l \cdot 0 + 0 = 1$  and obviously no choice of  $l$  can do this.

△

Note that the above matrix is “nice” as far as the usual requirements on a matrix go: It is real, regular, even symmetric. What more can one ask? And still it did not help. The question of existence for LU factorization has been studied and there is a theoretical answer.

**Theorem 23b.4.**

Let  $A$  be a real  $n \times n$  matrix. If its rank is  $n$  and its first  $k$  leading principal minors are non-zero, then it has an LU factorization.

Let  $A$  be a real regular  $n \times n$  matrix. It has an LU factorization if and only if all its leading principal minors are non-zero.

Given that evaluating those leading principals takes about  $n^4$  operations, it is actually easier to just try elimination. An even better idea is to figure out a way to have all the benefits with just some minor modification of our situation.

Recall that when we want to switch two rows in a matrix  $A$ , we can see it a multiplication  $PA$ , where  $P$  is a permutation matrix. In fact, we obtain it easily by taking the identity matrix  $E_n$  and switch the rows in it in the way we want to switch them in  $A$ . This allows us to capture the general process of elimination.

Given a regular matrix  $A$ , we can always reduce it to an upper triangular matrix  $U$  by applying elementary matrices as we outlined above, but once in a while we insert a permutation matrix  $P_{i,j}$  that switches rows  $i$  and  $j$ . Symbolically,

$$U = L \cdot L \cdots L \cdot P \cdot L \cdots L \cdot P \cdots L \cdot A.$$

This makes the situation somewhat more complicated, but it can be analyzed, and in particular one can explore what happens when we try to pull the permutation matrices out of the product. In the end we arrive at the formula  $LU = PA$ , where  $L$  is again a lower triangular matrix based on the row operations, but in this case some modifications have to be made to account for the permutations, it is not just coefficients  $l_{i,k}$  sitting at their places.

We can interpret the resulting formula as follows: We somehow guess beforehand what row exchanges have to be made during elimination, and capture them in the permutation matrix  $P$ . Then  $PA$  is a pre-processed matrix that can be reduced by elimination without any row exchanges, hence it has its LU-factorization  $PA = LU$ .

**Example 23b.d:** Consider the matrix  $A = \begin{pmatrix} 0 & 6 & 5 \\ 4 & -3 & 2 \\ 4 & -1 & 1 \end{pmatrix}$ .

Gaussian elimination would start by looking at the first column and finding the need to exchange rows. We have two fours to choose from and the one in the third row dominates its row more, so that will be our new pivot. After the first stage we obtain the matrices

$$U = \begin{pmatrix} 4 & -1 & 1 \\ 0 & -2 & 1 \\ 0 & 6 & 5 \end{pmatrix}, \quad L = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

We see that the second stage has no need of pivoting, and the third one is just the closing step, so we predict that just one row exchange will be enough. Namely, we want to exchange the first and the third row, which can be realized using the permutation matrix

$$P = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}.$$

Indeed,

$$PA = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 6 & 5 \\ 4 & -3 & 2 \\ 4 & -1 & 1 \end{pmatrix} = \begin{pmatrix} 4 & -1 & 1 \\ 4 & -3 & 2 \\ 0 & 6 & 5 \end{pmatrix}.$$

Now we apply Gaussian elimination to the latter matrix.

$$\begin{pmatrix} 4 & -1 & 1 \\ 4 & -3 & 2 \\ 0 & 6 & 5 \end{pmatrix} \sim \begin{pmatrix} 4 & -1 & 1 \\ 0 & -2 & 1 \\ 0 & 6 & 5 \end{pmatrix} \sim \begin{pmatrix} 4 & -1 & 1 \\ 0 & -2 & 1 \\ 0 & 0 & 8 \end{pmatrix}.$$

We first subtracted the first row from the second, so  $l_{2,1} = 1$ . Then we added the second three times to the third, so  $l_{3,2} = -3$ . This determines  $L$ , and  $U$  is just the last matrix in elimination. We have the following.

$$U = \begin{pmatrix} 4 & -1 & 1 \\ 0 & -2 & 1 \\ 0 & 0 & 8 \end{pmatrix}, \quad L = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & -3 & 1 \end{pmatrix}.$$

It is easy to check that

$$LU = \begin{pmatrix} 4 & -1 & 1 \\ 4 & -3 & 2 \\ 0 & 6 & 5 \end{pmatrix} = PA.$$

△

**Definition 23b.5.**

Consider a real  $n \times n$  matrix  $A$ . We say that  $n \times n$  matrices  $L, U$  and a permutation matrix  $P$  are an **LUP factorization** of  $A$  if  $U$  is an upper-triangular matrix,  $L$  is a lower-triangular matrix whose diagonal entries are all 1, and  $PA = LU$ .

This factorization does not fail us.

**Fact 23b.6.**

For every square matrix  $A$  there exists an *LUP* factorization.

This factorization can be again found using Gaussian elimination. However, this time we have to work more on it. We have to keep track of all row permutations, this is done by introducing a matrix  $P$ . At the beginning we set  $P = E_n$  and every time we exchange some rows, we do exactly the same to matrix  $P$ . and the end it is the right matrix  $P$  for the factorization.

However, this is not all, row exchanges also require some adjusting to the matrix  $L$  that we are slowly building. We will show a simplified algorithm that is tailored for regular matrices, and we include partial pivoting. It also works for singular matrices and produces one possible LU-factorization, but the resultign matrix  $U$  is not the one that we expect to see from regular Gaussian elimination, as we do not skip columns in this algorithm (cf Remark 23b.9).

**Algorithm 23b.7.**

⟨LUP factorization of a matrix⟩

Given: an  $n \times n$  matrix  $A = (a_{i,j})_{i,j=1}^n$  of real numbers.

**0.** Let  $U = A$  and  $L = P = E_n$  (unit matrix). Set  $k = 1$ .

**1.** If  $u_{i,k} = 0$  for all  $i \geq k$ , then go to step **3**.

Otherwise:

Among the rows  $i = k, \dots, n$  choose the row  $k'$  that has the largest possible value of  $|u_{i,k}|$ . If  $k' > k$ , then exchange rows  $k$  and  $k'$  in matrices  $U$  and  $P$ ; in the matrix  $L$  exchange the first  $k - 1$  entries of rows  $k$  and  $k'$ .

Continue with step **2**.

**2.** For  $i = k + 1, \dots, n$  do the following: Let  $l_{i,k} = \frac{u_{i,k}}{u_{k,k}}$ , set  $u_{i,k} = 0$  and for  $j > k$  compute  $u_{i,j} = u_{i,j} - l_{i,k}u_{k,j}$ .

**3.** If  $k < n$ , increase  $k$  by one and go back to step **1**.

Otherwise the algorithm stops.

**Output:** matrices  $P, L, U$ .

△

How do we use it in solving systems of equations? We have the following situation.

$$A\vec{x} = \vec{b} \iff PA\vec{x} = P\vec{b} \iff L(U\vec{x}) = P\vec{b}.$$

This suggests the following procedure.

**Algorithm 23b.8.**

⟨solving systems of linear equations using LUP factorization⟩

Given: a system  $A\vec{x} = \vec{b}$ , where  $A$  is a regular square matrix.

1. Find the LUP factorization  $LU = AP$ .
2. Using the forward substitution, solve the system  $L\vec{y} = P\vec{b}$  for  $\vec{y}$ .
3. Using the back substitution, solve the system  $U\vec{x} = \vec{y}$  for  $\vec{x}$ .

△

How much work does it take? Preparing the LUP factorization takes as many operations as GE, that is,  $\frac{2}{3}n^3 + O(n^2)$ . For every  $\vec{b}$  we then have to do the following:

We have to multiply  $P\vec{b}$ . This is nominally  $n^2$  operations, but note that  $P$  is a permutation matrix, so in fact there is no need to multiply, we just permute the entries in  $\vec{b}$ . If you do want to do it using multiplications, note that there is only one non-zero number in every row, so it will take only  $n$  multiplications.

Then we need to do the back and forward substitution for  $n^2$  each.

Conclusion: Preparation takes  $\frac{2}{3}n^3 + O(n^2)$  operations, thereafter every solution to a new system takes  $2n^2 + O(n)$  operations.

Compared to the idea with  $A^{-1}$  we saved  $\frac{1}{3}n^3$  operations.

This is the perfect opportunity to explain why LUP factorization is done for regular matrices. Its main application is exactly as we just saw, for solving systems of equations. If the original matrix  $A$  were not regular, then we could do the LUP factorization (see below), but the resulting matrix  $U$  would be singular as well. And then the back substitution would fail and the whole nice procedure we just developed would be useless. So we'd better stick with regular matrices.

Since we are using modifications of Gaussian elimination, practical problems with errors and overflows are exactly the same, so one has to be careful. Plus there is the problem of unfriendly systems that we keep referring to chapter 24.

**23b.9 Remark:**

As noted above, LUP factorization is traditionally used with regular matrices. What happens if we apply it to a singular matrix?

At the beginning of each stage we start by obtaining a good pivot. With a regular matrix and pivoting allowed, getting a non-zero pivot was guaranteed. Without pivoting we either had success or a total failure. With a singular matrix, a third possibility appears: We may have a column where all  $u_{i,k}$  for  $i \geq k$  are zero. What next?

This is the point where we realize that aims of general Gaussian elimination and LUP factorization differ. The elimination wants to have the tightest form of the matrix available, so it shifts its attention one column to the right and keeps the working row.

On the other hand, LUP factorization cares only about having zeros under the diagonal, which is true in the case we are just discussing, so we can take it as the usual successful stage and move on, passing to the next column and also to the next row. This means that, compared to the usual elimination, we have one less row to worry about. That is exactly the algorithm we used above. Let's see how this works in practice.

Consider the matrix  $\begin{pmatrix} 1 & -1 & 2 \\ 1 & -1 & 3 \\ -2 & 2 & 3 \end{pmatrix}$ . The first step of Gaussian elimination leads to  $U = \begin{pmatrix} 1 & -1 & 2 \\ 0 & 0 & 1 \\ 0 & 0 & 5 \end{pmatrix}$  and we store the coefficients  $l_{2,1}, l_{3,1}$  into  $L = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix}$ .

We pass to the second column and second row and we see that there are no candidates for our pivot. The simplified algorithm above is happy with the shape and passes to  $u_{3,3}$ . Since it is the last row, nothing more is done, the algorithm stops. We check that indeed

$$\begin{pmatrix} 1 & -1 & 2 \\ 1 & -1 & 3 \\ -2 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & -1 & 2 \\ 0 & 0 & 1 \\ 0 & 0 & 5 \end{pmatrix},$$

so we found a correct factorization.

The traditional Gaussian elimination would skip the second column and move its attention to the third, but still looking at the second row.  $u_{2,3} \neq 0$ , so we easily do the last step, subtract the second row from the third five times (so  $l_{3,2} = 5$ ) and obtain  $U = \begin{pmatrix} 1 & -1 & 2 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$ . Again, we check

that

$$\begin{pmatrix} 1 & -1 & 2 \\ 1 & -1 & 3 \\ -2 & 2 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -2 & 5 & 1 \end{pmatrix} \begin{pmatrix} 1 & -1 & 2 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}.$$

This incidentally shows that a singular matrix can really have more LUP factorizations.

As we noted above, LUP factorization is traditionally used for regular matrices, but it is good to know that if we encounter a singular one, we can still factorize it.

△

Just out of curiosity, this is a modification of LUP factorization that for singular matrices does the best shape, that is,  $U$  is in reduced row echelon form.

### Algorithm 23b.10.

⟨LUP factorization of a matrix⟩

Given: an  $n \times n$  matrix  $A = (a_{i,j})_{i,j=1}^n$  of real numbers.

**0.** Let  $U = A$  and  $L = P = E_n$  (unit matrix). Set  $k = 1$ ,  $K = 1$ .

**1.** If  $u_{i,K} = 0$  for all  $i \geq k$ , then go to step **3**. Otherwise:

Among the rows  $i = k, \dots, n$  with non-zero  $u_{i,K}$ , choose the row  $k'$  that has the largest  $|u_{i,K}|$ . If  $k' > k$ , then exchange rows  $k$  and  $k'$  in matrices  $U$  and  $P$ ; in the matrix  $L$  exchange the first  $k-1$  entries of rows  $k$  and  $k'$ .

Continue with step **2**.

**2.** For  $i = k+1, \dots, n$  do the following: Let  $l_{i,k} = \frac{u_{i,K}}{u_{k,K}}$ , set  $u_{i,K} = 0$  and for  $j > K$  compute  $u_{i,j} = u_{i,j} - l_{i,k}u_{k,j}$ .

Increase  $k$  by one and continue with step **3**.

**3.** If  $K < n$ , increase  $K$  by one and go back to step **1**.

Otherwise the algorithm stops.

**Output:** matrices  $P, L, U$ .

△

## 23c. Checking on and improving solution

When I was a schoolboy, math teachers would try to impress on me the need to check my answers when solving equations. We now learned some procedures for solving systems of equations and given that they are expected to feature errors when used on computers, this solution checking seems like a very good idea.

How does it work? Given  $A\vec{x} = \vec{b}$  we try to find the solution  $\vec{x}_0$ , but due to limitations of computers we find some other vector  $\vec{x}_a$  instead. What should we think about it?

The natural approach is to substitute into the left hand side of the equation and compare the outcome  $A\vec{x}_a$  with what should be there, that is,  $\vec{b}$ . The difference tells us something about the quality of our “solution”.

**Definition 23c.1.**

Consider a system of linear equations  $A\vec{x} = \vec{b}$ . Let  $\vec{x}_g$  be a candidate for solution. By its **residual** (or residual error) we mean the vector  $\vec{r} = \vec{b} - A\vec{x}_a$ .

What does the residual tell us about the quality of our solution? We are interested in the error we made, that is, in the vector  $\vec{E}_x = \vec{x}_0 - \vec{x}_a$ . Can we say anything about this vector? In the world of real numbers we would like to say that this error is “small”, but what does this mean in the world of vectors? The obvious candidate to look at is the magnitude of this vector, but to know something about it we would have to first find some connection between this error and the information that we have, that is, the residual. There is one.

$$\vec{r} = A\vec{x}_0 - A\vec{x}_a = A(\vec{x}_0 - \vec{x}_a) = A\vec{E}_x.$$

In an ideal world we could simply determine the error by calculating  $\vec{E}_x = A^{-1}\vec{r}$ . However, we now know that we would most likely find the inverse matrix using elimination, which is the same process that produced errors in  $\vec{x}_a$  in the first place, so we would not trust such a calculation any more that we trust  $\vec{x}_a$ .

So we give up the idea of finding the error precisely and ask a softer, yet important question: If we are lucky and the residual is small, can we then hope that also the error is small?

This is actually a pretty loaded question. The error is a vector, but that should not be such a bit problem, we all know how to figure out the size (magnitude) of a vector. However, we would need more. To see that, let’s pretend for a moment that we have just numbers (one equation, one unknown). The formula above then reads  $E_x = a^{-1}r$ . From that we would conclude that  $|E_x| = |a^{-1}| \cdot |r|$ , so indeed, if the residual is small, we obtain also a bound on the size of the error.

Unfortunately, when we pass to our real problem, we encounter the formula  $\vec{E}_x = A^{-1}\vec{r}$ . If we try to translate this into the language of “sizes”, we know what to do with vectors, but who ever heard about magnitude of a matrix? This is such a good question that it deserves its own chapter. We will explore this question and at the end arrive at some interesting answers, including the problem we just have here. So for conclusion of this discussion you will have to go to chapter .

However, there is something we can do now, so the work was not in vain. Note that the error that we made is determined by the equation  $A\vec{E}_x = \vec{r}$ , which is exactly the same system we just solved, just with a different right hand side. We learned how to solve systems repeatedly with less work, so we do it here, determine  $\vec{E}_x$  and then, theoretically, we could recover  $\vec{x}_0 = \vec{x}_a + \vec{E}_x$ . Unfortunately, we made some errors in solving for  $\vec{E}_x$  as well, but as we will see in chapter , this new error is very likely smaller than the original one. Thus we obtain a better approximation of the solution, and we can repeat this whole process until we are happy with the outcome.

**Algorithm 23c.2.**

⟨iterative improvement of solution⟩

Given: A system  $A\vec{x} = \vec{b}$ , where  $A$  is a square regular matrix.

**0.** Find a solution  $\vec{x}$  of the system  $A\vec{x} = \vec{b}$ ,

**1.** Determine the residual  $\vec{r} = \vec{b} - A\vec{x}$ . Solve the system  $A\vec{E}_x = \vec{r}$ .

If the error  $\vec{E}_x$  is not sufficiently small, do the correction  $\vec{x} := \vec{x} + \vec{E}_x$  and go back to step **1**.

△

What does it mean “sufficiently small”? Usually there is some tolerance  $\varepsilon$  given by a customer, so a good place to stop is when  $\|\vec{E}_x\| < \varepsilon$ . We know that this error has some error of its own, but in most cases this works reasonably well.

Another way to determine when to stop is when  $\vec{E}_x$  has entries comparable in size to the inherent roundoff error of the system. It is obviously not possible to get a better precision than the precision at which the computer is working, so when our error gets close to it, there is no point in more iterations.

## 24. Error in matrix calculations, condition number

In this chapter we will ask what impact does a small change on input have on the outcome of matrix calculation, this will also clarify the situation about checking solutions that we left unsolved. But before we do it, we have to introduce some concepts. If we want to talk about small changes in vectors and in matrices, we first need to have a way to measure them.

### 24a. Matrix norms

We start with vectors. The reader surely knows how to find a magnitude of a vector using squares and square root, but this is not the only possible way to do that. Depending on applications, other approaches might be more feasible. One particular motivation to look elsewhere is the fact that the usual Euclidean norm features square root, which sometimes causes trouble.

When we want to come up with some new way to express the size of a vector, we have to make sure that this new way is practical and that it feels like a size (or distance, we know that with vectors we easily pass from one to another). Around the beginning of the 20th century mathematicians formulated the properties needed for a notion of size to be reasonable. One popular type of measuring size (and thus distance) is called a norm.

#### Definition 24a.1.

Let  $V$  be a vector space. A mapping  $\|\cdot\|: V \mapsto \mathbb{R}$  is called a **norm** if it has the following properties:

- $\|\vec{x}\| \geq 0$  for all  $\vec{x} \in \mathbb{R}^n$ ;
- $\|\vec{x}\| = 0$  if and only if  $\vec{x} = \vec{0}$ ;
- $\|c\vec{x}\| = |c| \cdot \|\vec{x}\|$  for all  $\vec{x} \in \mathbb{R}^n$  and  $c \in \mathbb{R}$ ;
- $\|\vec{x} + \vec{y}\| \leq \|\vec{x}\| + \|\vec{y}\|$  for all  $\vec{x}, \vec{y} \in \mathbb{R}^n$  (triangle inequality).

The properties seem reasonable. We definitely do not expect “size” to be negative. The second property allows us to recognize the zero vector. The third property makes sure that scaling works well, and the fourth one tells us that this notion of size cannot distort our feeling of space too much. The name comes from the interpretation that in a triangle, one side cannot be longer than the other two put together. In other words, a straight trip cannot be longer than when we take a detour through some other place.

Note that in a vector space there are two operation—addition and multiplication by a scalar—and a norm offers some way to work with both of them. The most popular norms for real vectors are these:

$$\begin{aligned}\|\vec{x}\| &= \sqrt{\sum_{k=1}^n |x_k|^2} && \text{(Euclidean norm),} \\ \|\vec{x}\|_\infty &= \max_{k=1, \dots, n} |x_k| && \text{(max norm),} \\ \|\vec{x}\|_1 &= \sum_{k=1}^n |x_k| && \text{(sum norm).}\end{aligned}$$

They also work for complex vectors. All of them somehow express how we feel about vectors. The third norm can be thought of as a “taxi driver distance”. If he has to drive from one place to another in a city with a square grid of streets, then the distance taken is calculated in exactly this way.

The second norm seems natural for numerical analysis. A vector represents a collection of some values that we want to approximate. If we want to say that our approximation as a whole has a small error, then it is reasonable to interpret it as saying that all those values have this small error.

It should be noted that information can pass from one norm to another, because if we work in  $\mathbb{R}^n$  for a fixed  $n \in \mathbb{N}$ , then the information supplied by these norms cannot differ substantially. To be precise, we have the following estimates.

**Fact 24a.2.**

Let  $n \in \mathbb{N}$ . Then for every vector  $\vec{x} \in \mathbb{R}^n$  the following estimates are true:

- (i)  $\|\vec{c}\|_\infty \leq \|\vec{x}\|_2 \leq \|\vec{x}\|_1$ ,
- (ii)  $\|\vec{c}\|_2 \leq \sqrt{n}\|\vec{x}\|_\infty$ ,  $\|\vec{x}\|_1 \leq \sqrt{n}\|\vec{c}\|_2$ ,  $\|\vec{x}\|_1 \leq n\|\vec{c}\|_\infty$ .

By the way, we can see the largest possible difference on the vector  $(1, 1, \dots, 1)$ .

This fact shows that our choice of a particular norm should not change the substance of things, therefore it is mostly a matter of convenience. There are more norms (even infinitely many), but that belongs to another part of mathematics, we here look at those that are most popular in numerical analysis.

Now we would like to invent some notion of size for matrices. We could just ask for a norm, but with matrices we have an extra operation, that is, multiplication, and we would like to have a rule for it as well.

**Definition 24a.3.**

Let  $V$  be a vector space of matrices. A mapping  $\|\cdot\|: V \mapsto \mathbb{R}$  is called a **matrix norm** if it is a norm and also satisfies

- $\|AB\| \leq \|A\| \cdot \|B\|$  for all  $A, B \in M_{n \times n}$ .

It is useful to note what are not matrix norms. In many applications people use the spectral radius as an indication of influence of a matrix. However, it is not a matrix norm. It is interesting that it actually satisfies all the properties related to operations, like  $\rho(AB) \leq \rho(A) \cdot \rho(B)$ , it fails only one condition, the one of zero element. One can have a matrix that is not zero but has zero spectral radius, for instance  $\begin{pmatrix} 0 & 13 \\ 0 & 0 \end{pmatrix}$ .

Another parameter that tells us something about a matrix is the determinant, but it fails the very first condition. This can be fixed using absolute value, but  $|\det(A)|$  is still not a norm, because it also fails the zero condition. The matrix above has determinant equal to zero, yet it is not a zero matrix.

To get matrix norms we have to try something new, actually it is not completely new, inspiration from vectors is obvious. The following norms are most popular in numerical analysis.

$$\|A\|_\infty = \|A\|_R = \max_{i=1, \dots, n} \sum_{j=1}^n |a_{i,j}| \quad (\text{row-sum norm}),$$

$$\|A\|_1 = \|A\|_C = \max_{j=1, \dots, n} \sum_{i=1}^n |a_{i,j}| \quad (\text{column-sum norm}),$$

$$\|A\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^n |a_{i,j}|^2} \quad (\text{Frobenius norm}).$$

**Example 24a.a:** Consider the  $n \times n$  matrix  $A$  that has numbers 1 in the first row and 0 elsewhere. We easily find that

$$\|A\|_\infty = n, \quad \|A\|_F = \sqrt{n}, \quad \|A\|_1 = 1.$$

On the other hand, the matrix  $A^T$  has those 1s in the first column and we see that

$$\|A^T\|_\infty = 1, \quad \|A^T\|_F = \sqrt{n}, \quad \|A^T\|_1 = n.$$

△

This example shows that unlike the popular norms for vectors, here we have no hierarchy, it is not true that one norm would always be bounded by another. Depending which matrix we look at, one norm can be larger than the other or the other way around (or they are equal). However, just like with norms in  $\mathbb{R}^n$ , these matrix norms cannot differ too much, in fact the example above shows the worst case scenario.

**Fact 24a.4.**

Let  $n \in \mathbb{N}$ . For any  $n \times n$  matrix  $A$  we have the following.

$$\begin{aligned} \|A\|_1 &\leq n \|A\|_\infty, & \|A\|_\infty &\leq n \|A\|_1, \\ \|A\|_F &\leq \sqrt{n} \|A\|_1, & \|A\|_F &\leq \sqrt{n} \|A\|_\infty, \\ \|A\|_1 &\leq \sqrt{n} \|A\|_F, & \|A\|_\infty &\leq \sqrt{n} \|A\|_F. \end{aligned}$$

One thing this example shows is that the row-sum and column-sum norms are sensitive to transposition. The Frobenius norm does not care, which is obvious from the formula, the actual position of entries in a matrix are not taken into account in that sum.

There is a whole theory about matrix norms, but we do not have time for it here, we'll just need basic facts that will help us in our work. Just as an example of what can be done we show one observation.

**Fact 24a.5.**

For every  $n \in \mathbb{N}$  and for every matrix norm  $\|\cdot\|_M$  we have  $\|E_n\| = 1$ .

Indeed, according to the last property we have

$$\|E_n\| = \|E_n E_n\| \leq \|E_n\| \cdot \|E_n\|.$$

Since  $\|E_n\| \neq 0$  (see the second property), we can divide the inequality and obtain what we need.

We have norms, what do we want to do with them? Recall our motivation from chapter 23. We would like to know whether we can use the information that residual  $\vec{r} = A\vec{E}_x$  is small to conclude that also the error  $\vec{E}_x$  is small. We are in a position now to make the “small” precise, we choose some vector norm  $\|\cdot\|$  and we assume that  $\|\vec{r}\| < \varepsilon$ . Can we deduce some upper bound on  $\|\vec{E}_x\|$ ?

In case of real numbers we would change  $|r| = |aE_x|$  into  $|r| = |a| \cdot |E_x|$  and take it from there. It would be really nice if we could take some matrix norm  $\|\cdot\|_M$  and split  $\|A\vec{E}_x\| = \|A\|_M \cdot \|\vec{E}_x\|$ , but this is not possible in general. One reason is obvious, if we just choose some norm for vectors and another for matrices, then there is no reason why they should collaborate. We will therefore have to choose smartly, but even then it would be too much to ask for equality in that formula we desire.

**Definition 24a.6.**

Consider a norm  $\|\vec{x}\|$  for vectors from  $\mathbb{R}^n$  and a norm  $\|A\|_M$  for matrices from  $M_{n \times n}$ . We say that these norms are **compatible** if  $\|A\vec{x}\| \leq \|A\|_M \cdot \|\vec{x}\|$  for all  $A \in M_{n \times n}$  and  $\vec{x} \in \mathbb{R}^n$ .

Now we could go through lists of norms for vectors and matrix norms and start investigating which pairs work and which do not, but usually people prefer a different approach. It turns out that for every norm for vectors there is a matrix norm that is a perfect mate.

**Theorem 24a.7.**

(i) Let  $\|\cdot\|$  be a vector norm. The number defined for  $A \in M_{n \times n}$  by the formula

$$\|A\|_M = \sup \left\{ \frac{\|A\vec{x}\|}{\|\vec{x}\|}; \vec{x} \in \mathbb{R}^n \setminus \{\vec{0}\} \right\} = \sup \{ \|A\vec{x}\|; \vec{x} \in \mathbb{R}^n \wedge \|\vec{x}\| \leq 1 \}$$

determines a matrix norm compatible with  $\|\cdot\|$ . We call it the matrix norm **induced** by  $\|\cdot\|$ .

(ii) Let  $\|\cdot\|_M$  be a matrix norm. The number defined for  $\vec{x} \in \mathbb{R}^n$  by the formula

$$\|\vec{x}\| = \left\| \begin{pmatrix} x_1 & 0 & \dots & 0 \\ x_2 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ x_n & 0 & \dots & 0 \end{pmatrix} \right\|_M$$

is a norm on  $\mathbb{R}^n$  compatible with  $\|\cdot\|_M$ . It is called the norm induced by  $\|\cdot\|_M$ .

We see that also every matrix norm can find its mate, but this time the relationship is not as close. If we take the vector norm induced by  $\|\cdot\|_M$  and induce a matrix norm by it, we need not arrive back at  $\|\cdot\|_M$ .

This is related to the fact that matrix norms that are induced by someone are in some way special. Here is one observation.

**Fact 24a.8.**

For every  $n \in \mathbb{N}$  and for every matrix norm  $\|\cdot\|_M$  induced by some norm on  $\mathbb{R}^n$  we have  $\|E_n\| = 1$ .

This follows right from the definition.

Now what can we say about popular norms? The notation is helpful, the matrix norm  $\|A\|_1$  is induced by the norm  $\|\vec{x}\|_1$ , in particular they are compatible; similarly, the matrix norm  $\|A\|_\infty$  is induced by the norm  $\|\vec{x}\|_\infty$ . This is one of the reasons we like them a lot, all these norms are easy to evaluate and form natural pairs. As usual there are some drawbacks, one is the sensitivity of these two matrix norms to taking transposition, another disadvantage is that they are not the best to use in theory, they do not fit very well with other notions.

What about the Frobenius norm? Since  $\|E_n\|_F = \sqrt{n}$ , this norm cannot be induced by any vector norm. Note that according to the theorem, the Frobenius norm induces a certain vector norm, and it is easy to see that it is the Euclidean norm  $\|\vec{x}\|_2$ . Unfortunately, the norm induced by  $\|\vec{x}\|_2$  is not (and cannot be) the Frobenius norm. This explains why the Frobenius norm does not work all that well in theory applications. On the other hand, this norm has many nice properties, for instance it can be calculated easily, is indifferent to taking transposes, in many situations it is more favourable (on average) than the sum norms. Moreover, it is compatible with the natural Euclidean norm for vectors, so they can be used in various estimates.

This brings us to the natural question: What matrix norm gets induced by the Euclidean norm  $\|\vec{x}\|_2$ ? It is one that we did not meet before, we call it the **spectral norm** and it is given by the formula  $\|A\|_2 = \rho(A^*A)$ . This norm is perhaps the best for theory, but it is a real pain to evaluate it in practice, because we would have to find the largest eigenvalue of the matrix  $A^*A$  and there is no direct formula for it.

We thus arrived at one prominent difference between norms for vectors and matrix norms. When it comes to vectors from  $\mathbb{R}^n$ , there is a default norm. When one says “norm of a vector”, everybody assumes that the Euclidean norm  $\|\cdot\|_2$  is meant unless we specify otherwise. The Euclidean norm is easy to evaluate, works well in theory and also works reasonably well in applications (even though we sometimes prefer different norms for convenience).

In contrast, there is nothing like “the matrix norm”. There is no matrix norm that would satisfy most needs, people pick specific norms for specific needs.

Now we have norms and we can do something about our errors.

## 24b. Error of a solution

Recall our motivating problem: We have  $A\vec{E}_x = \vec{r}$  and we are asking whether we can argue that  $\vec{E}_x$  is small assuming that  $\vec{r}$  is small. We can do it as follows.

$$A\vec{E}_x = \vec{r} \implies \vec{E}_x = A^{-1}\vec{r} \implies \|\vec{E}_x\| \leq \|A^{-1}\|_M \|\vec{r}\|.$$

It would be natural to consider row-sum norms here, but we can use any pair of compatible norms. This is interesting, but we want to get a bit more. In numerical analysis we often prefer information about relative error, so we will try to deduce such a formula. However, before we do it we change our setup.

We have a perfect system  $A\vec{x}_0 = \vec{b}_0$ , where  $\vec{b}_0$  is the given right hand side and  $\vec{x}_0$  is the precise solution. Instead we obtained some other vector  $\vec{x}_a$  and we can calculate  $A\vec{x}_a = \vec{b}_a$ . In this setting, the residual is  $\vec{b}_0 - \vec{b}_a$ . The interesting part about this new setting is that we can interpret it differently: We have two systems of equations and we are trying to see some ties between them, namely understand what is going on with relationships between  $\vec{b}$ 's and  $\vec{x}$ 's. We have the following result.

### Theorem 24b.1.

Consider a vector norm  $\|\cdot\|$  and a compatible matrix norm  $\|\cdot\|_M$ .

Assume that vectors  $\vec{x}_0, \vec{x}_a$  and  $\vec{b}_0, \vec{b}_a$  are related by the formulas  $A\vec{x}_0 = \vec{b}_0$  and  $A\vec{x}_a = \vec{b}_a$ . Denote  $\vec{E}_x = \vec{x}_0 - \vec{x}_a$  and  $\vec{E}_b = \vec{b}_0 - \vec{b}_a$ . Then we have the following estimates.

$$\begin{aligned} \|\vec{E}_b\| &\leq \|A\|_M \cdot \|\vec{E}_x\|, \\ \frac{\|\vec{E}_x\|}{\|\vec{x}\|} &\leq \|A\|_M \cdot \|A^{-1}\|_M \frac{\|\vec{E}_b\|}{\|\vec{b}\|}. \end{aligned}$$

**Proof:** Subtracting the two systems we obtain

$$A(\vec{x}_0 - \vec{x}_a) = \vec{b}_0 - \vec{b}_a \implies A\vec{E}_x = \vec{E}_b,$$

hence  $\|\vec{E}_b\| \leq \|A\|_M \cdot \|\vec{E}_x\|$ .

We solve the first equality for  $\vec{x} = A^{-1}\vec{b}$  and then estimate  $\|\vec{x}\| \leq \|A^{-1}\|_M \cdot \|\vec{b}\|$ . Now we put the two estimates together,

$$\frac{\|\vec{E}_x\|}{\|\vec{x}\|} \leq \frac{\|A^{-1}\|_M \cdot \|\vec{E}_b\|}{\|\vec{x}\|} \leq \frac{\|A^{-1}\|_M \cdot \|\vec{E}_b\|}{\frac{1}{\|A\|_M} \|\vec{b}\|} = \|A^{-1}\|_M \cdot \|A\|_M \frac{\|\vec{E}_b\|}{\|\vec{b}\|}.$$

Done. □

The number that appears in the estimates is an important characterisation of a matrix.

### Definition 24b.2.

For an  $n \times n$  matrix  $A$  we define its **condition number** as  $\text{cond}_{\|\cdot\|}(A) = \|A\| \cdot \|A^{-1}\|$ .

If we introduce the notation  $\varepsilon_x$  for the relative error  $\frac{\|\vec{E}_x\|}{\|\vec{x}\|}$  of a vector, we can now rewrite the second inequality above as follows:

$$\varepsilon_x \leq \text{cond}_{\|\cdot\|}(A) \varepsilon_b.$$

Note that the condition number depends on the matrix norm we use. Obviously we always match norms so that all bits of information that come into our estimates fit together.

We return to our first interpretation, where  $\vec{E}_b$  is actually the residual  $\vec{r}$  according to our older notation. We obtain the following fact:

• Assume that solving a system  $A\vec{x} = \vec{b}$  we obtain an approximate solution  $\vec{x}_a$ . Let  $\vec{r} = \vec{b} - A\vec{x}_a$  be the residual. Then we have the following estimates for the absolute and relative error of  $\vec{x}$ :

$$\begin{aligned}\|\vec{E}_x\| &\leq \|A^{-1}\|_M \|\vec{r}\|, \\ \varepsilon_x &\leq \text{cond}_{\|\cdot\|}(A) \frac{\|\vec{r}\|}{\|\vec{b}\|}.\end{aligned}$$

Here is a corollary in practical terms.

**Fact 24b.3.**

Let  $\vec{x}_a$  be a numerical solution to a system  $A\vec{x} = \vec{b}$ , let  $\vec{r}$  be its residual. If  $\|\vec{r}\|_\infty > 10^{-k}$ , then the solution  $\vec{x}_a$  has  $k - \log_{10}(\text{cond}_\infty(A))$  correct digits.

It is obvious that we prefer matrices with small condition number. How small can it get?

**Fact 24b.4.**

For every matrix norm  $\|\cdot\|$  and for every matrix  $A$  we have  $\text{cond}_\infty(A) \geq 1$ .

The proof is simple, we use one of the defining properties of matrix norms.

$$\text{cond}_\infty(A) = \|A\| \cdot \|A^{-1}\| \geq \|AA^{-1}\| = \|E_n\| \geq 1.$$

That's about all concerning residual. However, the result we derived in theorem 24b.1 can tell us much more.

## 24c. Error propagation in systems of equations

With every numerical method we investigate its numerical stability, that is, how it reacts to errors that appear in numbers for one reason or another. This is done under the assumption that calculations are performed precisely, but for systems of equations we have finite methods, that is, when performed precisely, they provide a true solution. Thus in fact we will not be analyzing any particular method, but stability properties of equations themselves!

We already have one useful answer ready, we just look at the setup of theorem 24b.1 in a different way. We may interpret it as follows: There is a system  $A\vec{x} = \vec{b}_0$  that was supposed to be solved with expected solution  $\vec{x}_0$ , so  $A\vec{x}_0 = \vec{b}_0$ . Unfortunately, the vector of right hand sides got somehow corrupted, so we solved a different system  $A\vec{x} = \vec{b}_a$  instead, obtaining (precise) solution  $\vec{x}_a$ , so  $A\vec{x}_a = \vec{b}_a$  is really true. How does the error of  $\vec{b}$  influence the error of  $\vec{x}$ ?

The theorem gives an answer to this question. The larger the condition number is, the more the error may magnify. Note the conditional. The inequality provides an upper bound for the error of  $\vec{x}$ , and often it happens that things are much better. However, we cannot rely on that, so we prefer to work with matrices with small condition number.

Of course, if the right hand sides came with errors (for instance because we work in floating point format), then also the matrix very likely has some errors in it and instead of the real one  $A_0$  we get some other matrix  $A_a$  (hopefully close). Solving  $A_a\vec{x} = \vec{b}_a$  we obtain the precise solution  $\vec{x}_a$ . We denote  $E_A = A_0 - A_a$ ,  $\vec{E}_x = \vec{x}_0 - \vec{x}_a$ ,  $\vec{E}_b = \vec{b}_0 - \vec{b}_a$ , and also introduce the notion of relative error for matrices  $\varepsilon_A = \frac{\|E_A\|_M}{\|A\|_M}$ . After some work get an answer.

**Theorem 24c.1.**

Assume that matrices  $A_0, A_a$ , vectors  $\vec{b}_0, \vec{b}_a$  and solutions  $\vec{x}_0, \vec{x}_a$  are related by the formulas  $A_0\vec{x}_0 = \vec{b}_0$  and  $A_a\vec{x}_a = \vec{b}_a$ . Then

$$\varepsilon_x \leq \text{cond}(A) \left( \varepsilon_b + \varepsilon_A \cdot \frac{\|\vec{x}_a\|}{\|\vec{x}_0\|} \right).$$

If it happened that  $\vec{x}_a$  is about the same in norm as  $\vec{x}_0$ , we would get (roughly)

$$\varepsilon_x \leq \text{cond}(A)_{\|\cdot\|} (\varepsilon_b + \varepsilon_A).$$

In other words, if  $\text{cond}_{\|\cdot\|}(A)$  is large, then the error on input may be blown up. We see that there is a potential for trouble and we do not have to go far to find it.

**Example 24c.a:** Consider the matrix  $A = \begin{pmatrix} 50 & 25 \\ 51 & 25 \end{pmatrix}$  of a system with the right hand side  $\vec{b} = \begin{pmatrix} 250 \\ 254 \end{pmatrix}$ . We easily find the solution  $x = 4, y = 2$ .

We try the condition number:

$$\begin{aligned} \text{cond}_\infty(A) &= \left\| \begin{pmatrix} 50 & 25 \\ 51 & 25 \end{pmatrix} \right\|_\infty \cdot \left\| \begin{pmatrix} -1 & 1 \\ 51 & 2 \end{pmatrix} \right\|_\infty \\ &= 76 \cdot \frac{101}{25} \approx 307. \end{aligned}$$

That's quite a lot, if we are unlucky, relative errors on input are magnified up to 300 times. Let's see.

We will make a small permutation in  $\vec{b}$ , we take  $\begin{pmatrix} 250 \\ 256 \end{pmatrix}$  instead. In the row-sum norm this represents a relative error of about 0.008, less than one percent. However, the new system has the solution  $x = 6, y = -2$ , which is totally off.

By the way, we see that after we changed  $\vec{b}$ , only two digits in its entries it can be trusted. According to Fact 24b.3, we should trust two less digits of our result, which leaves nothing, exactly what we got here.

Now we try a small change in the matrix, with relative error under a percent again. The system  $\begin{pmatrix} 50 & 25 & | & 250 \\ 50.5 & 25 & | & 254 \end{pmatrix}$  has the solution  $x = 8, y = -6$ , this is again very far from the real solution.

△

Our example shows that a bad error estimate can easily happen, our matrix wasn't really monstrous. Note that the way that we derived the solution did not matter at all. These values for  $x$  and  $y$  are not given by some procedure, but by algebraic equations and they are determined uniquely. Thus the huge growth in error is not caused by some method but it is encoded into the nature of the problem itself.

It follows that there are problems (systems of linear equations) that do not cause trouble by its nature (for instance with small condition number) and systems that are dangerous by themselves. There is a traditional terminology used in such context. We say that a matrix is **well-conditioned** if it behaves well with respect to error propagation. One way to recognize a well-conditioned matrix is to show that its condition number is small. It is not quite clear what this means, the smallest possible condition number is one, so let's say that a well-conditioned matrix should not be significantly worse. Please don't ask us about specific border value.

Matrices that cause troubles (those with huge condition numbers are hot candidates) are called **ill-conditioned**. The matrix in the above example is obviously ill-conditioned. This terminology carries over to other types of problems, for instance we may say that a system of equations is well or ill-conditioned, but also a differential equation may be well or ill-conditioned and so on.

To appreciate the difference we look at a simple example.

**Example 24c.b:** Consider a system of two linear equations. Each equation represent a line in  $\mathbb{R}^2$  and the solution of the system is the intersection of these two lines. On the right we see two such systems (full lines), they have the same right hand sides. In both equations we change the right hand side in the first equation by the same amount (dashed line) Note that howe the new solution compare to the original ones. You can see that in the first system, the change in the solution is much smaller than in the second one.

We see where the problem is: In the second system, the directions of these lines are almost the same. When things become fuzzy (small errors in the matrix or in the right hand side), the intersection can move around a lot. Translating this into the language of matrices, a system whose matrix has rows that are almost linearly dependent will be sensitive to changes. Note that the rows in our  $A$  in example 24c.a has lines that are almost identical.

△

The observation about rows of matrices being nearly linearly dependent is good, but unfortunately we do not have a tool (some norm or another parameter) that could recognize it. What we know is that the conditional number can forse a matrix (or system of equations) to be well conditioned if it is small. If a matrix has large condition number, then it actually may be quite well behaved, but we cannot know.

Now the bad part. To determine the condition number we first have to find the inverse  $A^{-1}$ , but that is a lot of work and if a matrix is ill-conditioned then we could not trust this inverse anyway. People found some ways to estimate the condition number, they also developed some ways that are not precise but with a bit of luck drop useful hints.

- After solving a system  $A\vec{x} = \vec{b}$ , try to solve it again with slightly permuted  $A$  and  $\vec{b}$ . If the solution changes significantly, it is a warning sign.
- Divide each row of a matrix by its maximal number, so that all terms in  $A$  are (in absolute value) at most 1. Find  $A^{-1}$ . If it has some very large entries, it is a warning sign.

Of course, if these tests work out well then it does not really mean that our matrix is well conditioned, we might have been lucky.

## 24d. Numerical stability of elimination

When we do elimination, we create a sequence of matrices

$$A = A_1 \mapsto A_2 \mapsto A_3 \mapsto \cdots \mapsto A_N = U.$$

Each matrix is obtained from the previous one using a row operation that can be represented by a special matrix  $L$ , which is essentially the unit matrix with appropriate  $l_{i,k}$  added as one entry off the diagonal. That is,  $A_{m+1} = L_m A_m$ , and we are interested in what happens to errors in  $A_m$  after such step. We can use similar approach as above and deduce that the growth of the error can be controlled if the condition numbers of  $L_m$  and  $A_m$  are small.

At this point it will be useful to focus on the row-sum norm, because due to the specific form of  $L_m$  we can easily evaluate

$$\|L_m\|_\infty = \|L_m^{-1}\|_\infty = 1 + |l_{i,k}|.$$

We see that it is in our best interest to keep this as small as possible, which brings us back to benefits of the partial pivoting we already discussed in chapter 22.

Considering the condition number of  $A_m$ , it depends on the conditional number of the original matrix  $A$  and also on what we do with it. We see that we should be trying to keep condition numbers of all matrices created during elimination as small as possible. Note that pivoting itself does not change the condition number, because the norms that we are using do not care about the order of rows. However, it does influence the condition number of the resulting matrix after elimination.

It would be nice if we could say that when we do one stage of elimination and we chose the pivot wisely, then the condition number does not increase. Unfortunately, this is not true; however, pivoting in most cases does not allow the condition number to grow to much, which is also nice.

Numerical analysts studied elimination for almost a hundred years now (it really is important in applications) and there are deep results on what happens when we do elimination on computers with floating point arithmetics, but they are beyond the scope of this book. What we can take from those results is that pivoting is good, not only because of error propagation, they may even help alleviate roundoff errors in matrix calculations. But despite all this, we should be careful anyway because things can still go wrong.

We will illustrate some points made here in the following example.

**Example 24d.a:** Consider the matrix  $A = \begin{pmatrix} \varepsilon & 1 \\ 1 & 1 \end{pmatrix}$ , where  $\varepsilon$  is a very small number. We do not really like such an unsuitable candidate for the pivot, but we try it anyway and do the first stage of Gaussian elimination, obtaining the matrix  $A_2 = \begin{pmatrix} \varepsilon & 1 \\ 0 & -\frac{1-\varepsilon}{\varepsilon} \end{pmatrix}$ . How much do we like the new matrix?

We easily find that  $A^{-1} = \begin{pmatrix} \frac{-1}{1-\varepsilon} & \frac{1}{1-\varepsilon} \\ \frac{1}{1-\varepsilon} & \frac{-\varepsilon}{1-\varepsilon} \end{pmatrix}$  and  $A_2^{-1} = \begin{pmatrix} 1 & -\frac{\varepsilon}{1-\varepsilon} \\ 0 & \frac{\varepsilon}{1-\varepsilon} \end{pmatrix}$  and we are ready to look at condition numbers. We will show how they work for the three most popular norms.

	$A$	$A_2$
$\text{cond}_\infty$	$\frac{4}{1-\varepsilon} \approx 4$	$\frac{1}{\varepsilon}$
$\text{cond}_1$	$\frac{4}{1-\varepsilon} \approx 4$	$\frac{1}{\varepsilon}$
$\text{cond}_F$	$\frac{3+\varepsilon^2}{1-\varepsilon} \approx 3$	$\sqrt{1 + \varepsilon^2 + \frac{(1-\varepsilon)^2}{\varepsilon^2}} \sqrt{1 + \frac{2\varepsilon^2}{(1-\varepsilon)^2}} \approx \frac{1}{\varepsilon}$

We see that no matter what norm we try, row elimination made the condition number much bigger, it goes to infinity as  $\varepsilon \rightarrow 0^+$ .

We take the good advice and try partial pivoting, so we start our elimination with the matrix  $B = \begin{pmatrix} 1 & 1 \\ \varepsilon & 1 \end{pmatrix}$  and obtain  $B_2 = \begin{pmatrix} 1 & 1 \\ 0 & 1 - \varepsilon \end{pmatrix}$ .

Inverse matrices are  $B^{-1} = \begin{pmatrix} \frac{1}{1-\varepsilon} & \frac{-1}{1-\varepsilon} \\ \frac{-\varepsilon}{1-\varepsilon} & \frac{1}{1-\varepsilon} \end{pmatrix}$  and  $B_2^{-1} = \begin{pmatrix} 1 & \frac{-1}{1-\varepsilon} \\ 0 & \frac{1}{1-\varepsilon} \end{pmatrix}$ , hence

	$B$	$B_2$
$\text{cond}_R$	$\frac{4}{1-\varepsilon} \approx 4$	$\frac{2(2-\varepsilon)}{1-\varepsilon} \approx 4$
$\text{cond}_S$	$\frac{4}{1-\varepsilon} \approx 4$	$\frac{2(2-\varepsilon)}{1-\varepsilon} \approx 4$
$\text{cond}_F$	$\frac{3+\varepsilon^2}{1-\varepsilon} \approx 3$	$\frac{2+(1-\varepsilon)^2}{1-\varepsilon} \approx 3$

As expected,  $B$  has the same condition number as  $A$ . However, the outcome of elimination couldn't have been more different, with  $A$  it grew beyond control, here the condition number actually slightly improved.

Now we look at the other claim, about alleviating roundoff errors. We will attempt to solve the system  $A = \left( \begin{array}{cc|c} \varepsilon & 1 & 1 + \varepsilon \\ 1 & 1 & 2 \end{array} \right)$  whose obvious solution is  $x = y = 1$ .

We will analyze what happens when we do operations with  $k$  digit precision and  $\varepsilon < 10^{-k}$ . Note that in this case the computer does not have a problem storing  $\varepsilon$ , but the number  $1 + \varepsilon$  is seen as 1. The computer therefore thinks that it is supposed to solve the system  $\left( \begin{array}{cc|c} \varepsilon & 1 & 1 \\ 1 & 1 & 2 \end{array} \right)$

Elimination without pivoting should lead to the matrix  $\left( \begin{array}{cc|c} \varepsilon & 1 & 1 \\ 0 & 1 - 1/\varepsilon & 2 - 1/\varepsilon \end{array} \right)$ , but computer sees it differently. Since  $\varepsilon < 10^{-k}$ , we have  $\frac{1}{\varepsilon} > 10^k$ , so according to computer  $1 - \frac{1}{\varepsilon} = \frac{1}{\varepsilon}$ . This the

computer actually sees the matrix  $\left(\begin{array}{cc|c} \varepsilon & 1 & 1 \\ 0 & -1/\varepsilon & -1/\varepsilon \end{array}\right)$  and using back substitution easily concludes that the solution is  $x = 0, y = 1$ , which is obviously wrong.

If we allow pivoting, computer starts with the matrix  $\left(\begin{array}{cc|c} 1 & 1 & 2 \\ \varepsilon & 1 & 1 \end{array}\right)$ , which should theoretically reduce to  $\left(\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1 - \varepsilon & 1 - 2\varepsilon \end{array}\right)$ , but the computer sees  $\left(\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1 & 1 \end{array}\right)$ . Back substitution yields the correct solution. The pivoting really helped (this time).

△

## 25. Solving systems of linear equations by iteration

In chapter 20 we transformed the task of solving an equation into a fixed point problem. We can do the same with systems of linear equations. Given a system  $A\vec{x} = \vec{b}$ , we can do

$$A\vec{x} = \vec{b} \iff A\vec{x} + \vec{x} = \vec{b} + \vec{x} \iff (A + E_n)\vec{x} - \vec{b} = \vec{x}.$$

Then we could define a mapping  $\mathbb{R}^n \mapsto \mathbb{R}^n$  by the formula  $\vec{x} \mapsto (A + E_n)\vec{x} - \vec{b}$  and try iteration. As a motivational example it wasn't bad, but we will look at better ways. Before we do it, we will look closer at that iterating business.

Inspired by the above formula, we will look at iteration schemes of the form  $\vec{x}_{k+1} = B\vec{x} + \vec{c}$ . Such a scheme produces a sequence of vectors  $\{\vec{x}_k\}$  and we need to learn how to work with them. In particular, we need to know what convergence means for sequences of vectors. We follow inspiration from sequences of real numbers, where the convergence  $x_k \rightarrow x$  is equivalent to the fact that the distance between  $x_k$  and  $x$  goes to zero. We know how to do distances in vector spaces and distances are numbers for which convergence is known, so this should be easy. Similarly we work with sequences of matrices.

**Definition 25.1.**

Let  $n \in \mathbb{N}$ , let  $\{\vec{x}_k\}$  be a sequence of vectors from  $\mathbb{R}^n$ . We say that it **converges** to  $\vec{x} \in \mathbb{R}^n$  with respect to a vector norm  $\|\cdot\|$  if  $\|\vec{x}_k - \vec{x}\| \rightarrow 0$ .  
 Let  $n \in \mathbb{N}$ , let  $\{A_k\}$  be a sequence of  $n \times n$  matrices. We say that it **converges** to an  $n \times n$  matrix  $A$  with respect to a matrix norm  $\|\cdot\|$  if  $\|A_k - A\| \rightarrow 0$ .

It can be easily shown that a sequence of vectors  $\vec{x}_k$  converges to a vector  $\vec{x}$  if and only if for every  $k$ , the  $k$ th coordinate of vectors  $\vec{x}_k$  converge to the  $k$ th coordinate of  $\vec{x}$ . It is also easy to show that because we have mutual comparison between norms in  $\mathbb{R}^n$ , the notion of convergence does not actually depend on the choice of a norm, so we can use any norm we feel like. In numerical analysis we like to use the max norm  $\|\cdot\|_\infty$ .

Analogous facts are also true about matrices, so from now on we just say that a sequence of vectors or matrices converges to some element without referring to the norm used to check it.

### 25a. Convergence of iteration scheme

If a sequence of vectors comes from the iteration scheme that we are interested in, then there is a complete answer to the problem of convergence.

**Theorem 25a.1.**

An iterative method  $\vec{x}_{k+1} = B\vec{x}_k + \vec{c}$  converges if and only if  $\rho(B) < 1$ .

Unfortunately, finding the spectral radius is not so simple, so we would prefer another criterion, perhaps not so good, but more convenient. We get help from the following theorem.

**Fact 25a.2.**

Consider an  $n \times n$  matrix  $B$ .  
 Every induced matrix norm  $\|\cdot\|$  satisfies  $\rho(B) \leq \|B\|$ .  
 Conversely, for every  $\varepsilon > 0$  there is an induced matrix norm  $\|\cdot\|$  such that  $\|B\| - \varepsilon < \rho(B)$ .

The first statement is the one that we want, it allows us to force the spectral radius to be small. It is actually simple to prove. We take any eigenvalue  $\lambda$  of  $B$  with eigenvector  $\vec{x}$  and estimate

$$|\lambda| \cdot \|\vec{x}\| = \|\lambda\vec{x}\| = \|B\vec{x}\| \leq \|B\| \cdot \|\vec{x}\|.$$

We cancel and obtain  $|\lambda| \leq \|B\|$  for any eigenvalue, in particular it is true for the largest eigenvalue that gives the spectral radius.

As a corollary we obtain a practical statement.

**Theorem 25a.3.**

If a matrix  $B$  satisfies  $\|B\|_M < 1$  for some compatible matrix norm, then the corresponding iterative method  $\vec{x}_{k+1} = B\vec{x}_k + \vec{c}$  converges to  $\vec{x}_f$  for arbitrary choice of  $\vec{x}_0$  and we have

$$\|\vec{x}_f - \vec{x}_{k+1}\| \leq \frac{\|B\|_M}{1 - \|B\|_M} \|\vec{x}_{k+1} - \vec{x}_k\|.$$

We can obtain this result also in a different way. Recall that in chapter 20 we mentioned that the Banach contraction theorem works in many general settings, in particular it works in vector spaces with a norm. Thus one way to show convergence of our iteration scheme is to prove that the mapping  $\varphi(\vec{x}) = B\vec{x} + \vec{c}$  is a contraction. Indeed, we can estimate

$$\begin{aligned} \|\varphi(\vec{x}) - \varphi(\vec{y})\| &= \|(B\vec{x} + \vec{c}) - (B\vec{y} + \vec{c})\| \\ &= \|B(\vec{x} - \vec{y})\| \leq \|B\| \cdot \|\vec{x} - \vec{y}\|. \end{aligned}$$

The condition  $q = \|B\| < 1$  shows that we indeed have a contraction.

Now we are ready.

## 25b. Iteration for systems of equations

We transform equations into fixed point problems by creating an expression of the form  $x = \dots$ . To get an inspiration we look at a small system.

$$a_{1,1}x_1 + a_{1,2}x_2 = b_1$$

$$a_{2,1}x_1 + a_{2,2}x_2 = b_2$$

We want some formulas for  $x_1$  and  $x_2$ , why not using the two equations for it? We obtain

$$x_1 = \frac{1}{a_{1,1}}(b_1 - a_{1,2}x_2)$$

$$x_2 = \frac{1}{a_{2,2}}(b_2 - a_{2,1}x_1)$$

Generalizing this idea to larger systems seems obvious. Iteration works as follows. Given a vector  $\vec{x}_k$ , we substitute its coordinates into the expressions on the right and obtain a new vector  $\vec{x}_{k+1}$ .

We will need to access coordinates of these vectors, which is a bit awkward, the  $i$ th coordinate of a vector  $\vec{x}$  is usually denoted  $x_i$ , but now we use subscript to index the sequence. To avoid confusion, we will use the notation  $(\vec{x}_k)_i$  for the coordinate here.

**Algorithm 25b.1.**

⟨JIM: Jacobi iteration method⟩

Given: a system  $A\vec{x} = \vec{b}$  of linear equations with a square  $n \times n$  matrix, tolerance  $\varepsilon$ , arbitrary initial vector  $\vec{x}_0$ .

0. Set  $k = 0$ .

1. For all  $i = 1, \dots, n$  compute

$$(\vec{x}_{k+1})_i = \frac{b_i}{a_{i,i}} - \frac{1}{a_{i,i}} \left( \sum_{j=1}^{i-1} a_{i,j}(\vec{x}_k)_j + \sum_{j=i+1}^n a_{i,j}(\vec{x}_k)_j \right).$$

If  $\|\vec{x}_{k+1} - \vec{x}_k\|_\infty \geq \varepsilon$ , increase  $k$  by one and go back to step 1.

△

As usual, in the stopping condition we could also use condition with relative change, that is,  $\frac{\|\vec{x}_{k+1} - \vec{x}_k\|_\infty}{\|\vec{x}_k\|_\infty} \geq \varepsilon$ . We use the infinity norm, because it is traditional in this setting, it controls directly sizes of all coordinates. It is also wise to set a condition on maximal number of iterations.

It is obvious that the Jacobi iteration can be used only if the diagonal entries in  $A$  are non-zero. That's life. In case of trouble we may try to switch rows, but in practical applications where iterative methods are used we usually need not worry.

To learn something about convergence we need to rephrase this iteration into the form we studied above, that is, with matrices. It will be easier if we decompose  $A$  into three prominent parts, that is, we separate its diagonal, entries under the diagonal, and entries above the diagonal. Mathematically, we write

$$A = D + L + U,$$

where  $D$  is a diagonal matrix,  $L$  is lower triangular,  $U$  is upper triangular and both  $L$  and  $U$  have diagonal entries equal to zero. This decomposition obviously always exists and is unique. Note that if  $A$  has non-zero diagonal entries  $d_{i,i}$ , then  $D^{-1}$  exists, it is a diagonal matrix and has  $\frac{1}{d_{i,i}}$  on the diagonal.

Now it is easy to check that Jacobi's iteration scheme can be written as follows:

$$\vec{x}_{k+1} = D^{-1}\vec{b} - D^{-1}(L + U)\vec{x}_k.$$

In this form we easily check that if the sequence  $\vec{x}_k$  converges to a fixed point  $\vec{x}_f$ , then this fixed point solves the original system.

$$\vec{x}_f = -D^{-1}(L + U)\vec{x}_f + D^{-1}\vec{b} \iff D\vec{x}_f = -(L + U)\vec{x}_f + \vec{b} \iff (D + L + U)\vec{x}_f = \vec{b}.$$

Written in matrix form, the Jacobi iteration fits with the general setup discussed in the previous section, with  $B_{\text{JIM}} = -D^{-1}(L + U)$  and  $\vec{c} = D^{-1}\vec{b}$ . However, we cannot say anything about the norm of  $B_J$  in general, so the scheme converges only sometimes. We will look at it below, after we introduce our second iteration method.

The key observation for our improved method can be made in our simple example of two equations.

$$\begin{aligned} x_1 &= \frac{1}{a_{1,1}}(b_1 - a_{1,2}x_2) \\ x_2 &= \frac{1}{a_{2,2}}(b_2 - a_{2,1}x_1) \end{aligned}$$

Jacobi says: When you have a vector  $\vec{x}_k$ , put it into the formulas on the right and obtain  $\vec{x}_k$ . Gauss with Seidel say: Isn't it a bit conservative? When we calculate  $x_2$ , we actually already know the new, improved value of  $x_1$ , so why don't we use it?

This sounds like something that could speed things up and we easily generalize this to more equations.

### Algorithm 25b.2.

⟨GSM: Gauss-Seidel iteration⟩

Given: a system  $A\vec{x} = \vec{b}$  of linear equations with a square  $n \times n$  matrix, tolerance  $\varepsilon$ , arbitrary initial vector  $\vec{x}_0$ .

0. Set  $k = 0$ .

1. For all  $i = 1, \dots, n$  compute

$$(\vec{x}_{k+1})_i = \frac{b_i}{a_{i,i}} - \frac{1}{a_{i,i}} \left( \sum_{j=1}^{i-1} a_{i,j}(\vec{x}_{k+1})_j + \sum_{j=i+1}^n a_{i,j}(\vec{x}_k)_j \right).$$

If  $\|\vec{x}_{k+1} - \vec{x}_k\|_\infty \geq \varepsilon$ , increase  $k$  by one and go back to step 1.

△

Again, relative change can be used as a stopping condition and a maximal allowed number of iterations should be specified.

In order to obtain a matrix form we again decompose  $A$  as above and write

$$\vec{x}_{k+1} = -D^{-1}(L\vec{x}_{k+1} + U\vec{x}_k) + D^{-1}\vec{b}.$$

This is the formula as used in the algorithm, but in order to use the general setup we would need to have  $\vec{x}_{k+1}$  all by itself on the left. No problem, we solve the equation. We multiply both sides by  $D$  and then sort it out.

$$\begin{aligned} D\vec{x}_{k+1} &= -L\vec{x}_{k+1} - U\vec{x}_k + \vec{b} \implies D\vec{x}_{k+1} + L\vec{x}_{k+1} = -U\vec{x}_k + \vec{b} \\ &\implies (D + L)\vec{x}_{k+1} = -U\vec{x}_k + \vec{b} \\ &\implies \vec{x}_{k+1} = -(D + L)^{-1}U\vec{x}_k + (D + L)^{-1}\vec{b}. \end{aligned}$$

Thus the matrices as in the general iteration are  $B_{\text{GSM}} = -(D + L)^{-1}U$  and  $\vec{c} = (D + L)^{-1}\vec{b}$ .

We check that if we obtain a fixed point, then we have a solution of our equation.

$$\vec{x} = -(D + L)^{-1}U\vec{x} + (D + L)^{-1}\vec{b} \iff (D + L)\vec{x} = -U\vec{x} + \vec{b} \iff A\vec{x} = \vec{b}.$$

Yes it does.

We have two iterative schemes for solving systems of linear equations and it is time to ask whether they are any good.

We know that convergence depends on the spectral radius  $\rho(B)$ . Unfortunately, it takes some work to find it, so we will try a different approach and identify types of matrices for which these two iterations are known to work. But before we do it, we do mention some results. It is known that matrices coming from certain applications do have small eigenvalues because of the nature of the thing that is studied. Then iteration works well. It would seem that Gauss-Seidel iteration is better, but this is not exactly true, it can easily happen that one converges and the other does not. So the Jacobi iteration does have its uses, one interesting aspect is that it is immune to row permutations, whereas the Gauss-Seidel iteration is sensitive to it, switching a few rows (that is, changing the order of equations) can cause it to stop converging.

An interesting result is that if  $\rho(B_{\text{JIM}}) = 0$  or  $\rho(B_{\text{GSM}}) = 0$ , then we should get the precise answer after a finite number of steps. However, this does not happen too often.

For some matrices it is known that GSM works better. For instance, if a matrix  $A$  is three-diagonal (non-zero numbers are allowed only on the diagonal and next to it, very useful matrix in differential equations), then  $\rho(B_{\text{GSM}}) = \rho(B_{\text{JIM}})^2$ . If these numbers are smaller than 1 then GSM is significantly faster, for  $\vec{x}_k$  close to  $\vec{x}_f$  it determines correct digits twice as fast compared to JIM.

Now we look at other ways to recognize “good” matrices for iteration. We introduce two notions, the first is easy to see, the second is more involved and understanding its meaning requires a deeper dip into the matrix theory. Let’s just say that again, it is something useful in many applications.

**Definition 25b.3.**

Consider an  $n \times n$  matrix  $A$ .

We say that  $A$  is **strictly diagonally dominant** if

$$|a_{i,i}| > \sum_{j \neq i} |a_{i,j}|$$

for all  $i = 1, \dots, n$ .

We say that  $A$  is **positive definite** if  $\vec{x}^T A \vec{x} > 0$  for all non-zero vectors  $\vec{x} \in \mathbb{R}^n$ .

Here is the good news.

**Theorem 25b.4.**

If  $A$  is strictly diagonally dominant, then both JIM and GSM converge for arbitrary choice of initial vector.

If  $A$  is symmetric and positive definite, then GSM converges for arbitrary choice of initial vector.

That is a very good news.

How does iteration compare to elimination? We know that elimination requires about  $\frac{2}{3}n^3$  operations. On the other hand, we easily conclude that both the Jacobi and the Gauss-Seidel iterations require  $2n^2 + O(n)$  operations for one iteration, which is very nice. However, this iteration has to be repeated, say  $N$  times before we obtain a reasonably precise answer. The total number of operations is therefore about  $2n^2N$  and we obviously do not know  $N$ .

What we do see is that iteration will be better than iteration in case that  $N$  is significantly smaller than  $n$ . For some matrices this is known to be true. If  $n$  is about a million (or even billion), then we can easily afford a hundred iterations and we still get the solution ten thousand times faster than using elimination. By the way, under a hundred iterations to get a good approximation of a solution is fairly typical. So if iteration works, it can be very helpful.

Thing can get even better. If a matrix is sparse (most of the entries are zero) and the non-zeros are spaces regularly, we can actually code it into the formula for  $(\vec{x}_k)_i$  and add only the terms that matter. The case of a three-diagonal matrix deserves another mention. Since there are only three non-zero numbers in every row, determining  $(\vec{x}_k)_i$  requires six operations, so one whole iteration takes measly  $6n$  operations. This is an incredible difference.

We see that we really want to have convergent iterations, and we hope for quick convergence. In chapter 20 we learned about a method that sometimes speeds up iteration, we called it relaxation and the trick was that we were looking for a suitable compromise between the iteration we want and constant iteration that had guaranteed and fast convergence. The same idea can be used with iteration for matrices, we will apply it to Gauss-Seidel iteration.

We introduce a parameter  $\lambda > 0$  indicating how much we trust GSM. Let  $\vec{x}_{k+1}^G$  be the vector obtained from  $\vec{x}_k$  using Gauss-Seidel iteration. We decide to use instead the vector  $\vec{x}_{k+1} = (1 - \lambda)\vec{x}_k + \lambda\vec{x}_{k+1}^G$ . It is obvious that taking  $\lambda = 1$  we get GSM. When we look what it does to individual coordinates, we arrive at the following procedure.

**Algorithm 25b.5.**

⟨SOR, Successive OverRelaxation method⟩

Given: a system  $A\vec{x} = \vec{b}$  of linear equations with a square  $n \times n$  matrix, parameter of relaxation  $\lambda$ , arbitrary initial vector  $\vec{x}_0$ .

**0.** Set  $k = 0$ .

**1.** For all  $i = 1, \dots, n$  compute

$$(\vec{x}_{k+1})_i = (1 - \lambda)(\vec{x}_k)_i - \frac{\lambda}{a_{i,i}} \left( \sum_{j=1}^{i-1} a_{i,j}(\vec{x}_{k+1})_j + \sum_{j=j+1}^n a_{i,j}(\vec{x}_k)_j \right) + \frac{\lambda b_i}{a_{i,i}}.$$

If  $\|\vec{x}_{k+1} - \vec{x}_k\|_\infty \geq \varepsilon$ , increase  $k$  by one and go back to step **1**.

△

We used to translate this into matrix form, but we will do it differently this time, we start from the given equation and rearrange it using  $\lambda$ , just like we did in chapter 20. This will also confirm that the scheme solves the given equation. We start by rearranging the given equation in the way

we used when working our GSM above.

$$\begin{aligned} A\vec{x} = \vec{b} &\iff \vec{x} = -D^{-1}(L\vec{x} + U\vec{x}) + D^{-1}\vec{b} \iff \lambda\vec{x} = -\lambda D^{-1}(L\vec{x} + U\vec{x}) + \lambda D^{-1}\vec{b} \\ &\iff \vec{x} = (1 - \lambda)\vec{x} - \lambda D^{-1}(L\vec{x} + U\vec{x}) + \lambda D^{-1}\vec{b}. \end{aligned}$$

That's the blueprint for the scheme, now we make a proper iteration out of it.

$$\begin{aligned} \vec{x}_{k+1} &= (1 - \lambda)\vec{x}_k - \lambda D^{-1}L\vec{x}_{k+1} - \lambda D^{-1}U\vec{x}_k + \lambda D^{-1}\vec{b} \\ &\implies (E_n + \lambda D^{-1}L)\vec{x}_{k+1} = (1 - \lambda)E_n\vec{x}_k - \lambda D^{-1}U\vec{x}_k + \lambda D^{-1}\vec{b} \\ &\implies (D + \lambda L)\vec{x}_{k+1} = (1 - \lambda)D\vec{x}_k - \lambda U\vec{x}_k + \lambda\vec{b} \\ &\implies \vec{x}_{k+1} = (D + \lambda L)^{-1}[(1 - \lambda)D - \lambda U]\vec{x}_k + \lambda(D + \lambda L)^{-1}\vec{b}. \end{aligned}$$

This fits the general iteration setup with  $B_\lambda = (D + \lambda L)^{-1}[(1 - \lambda)D - \lambda U]$  and  $\vec{c}_\lambda = \lambda(D + \lambda L)^{-1}\vec{b}$ . We can see that as we send  $\lambda$  to zero, the diagonal becomes dominant, which is a good thing. However, one should not overdo this.

How do we find a good relaxation parameter? We usually fish between numbers 0 and 2, the following statement shows why.

**Theorem 25b.6.**

(Ostrovsky)

Let  $A$  be a symmetric  $n \times n$  matrix with positive diagonal entries. Then  $\rho(B_\lambda) < 1$  if and only if  $A$  is positive definite and  $0 < \lambda < 2$ .

That's about the best help one gets, good  $\lambda$  must be discovered experimentally. Obviously, this is not worth the trouble when solving just one system. However, in some applications people end up solving again and again a system that changes little each time, so one can expect that its convergence will behave in analogous ways. Then it makes sense to try a slightly different  $\lambda$  each time and see what happens.

We are near the end. We conclude by asking about numerical stability. Of course, calculating  $\vec{x}_{k+1}$  in floating point introduces errors. However, this is an iterative scheme that is (with a bit of luck) trying to get us closer to the real solution, so it fixes these errors for us.

Which brings us to some recommendations. When do we want to try iteration? For instance when the matrix of the system is ill conditioned, because then we cannot trust elimination. Another good reason is if the given system is really really huge. And yet another good reason is when the system has a sparse matrix with structure that can be used to efficiently program the iteration step.

## 26. (Homogeneous) systems of linear differential equations

Passing from individual differential equations to systems is natural, given that ODEs are an important tool for natural scientists and the nature is naturally more dimensional. In this chapter we will look at systems of linear differential equations of order 1 with several variables. A typical linear ODE of order one with two variables may look like as follows.

$$3y' - z' + 13y + 23z = e^x.$$

Here it is understood that  $y = y(x)$  and  $z = z(x)$  are unknown functions.

However, this is still too complicated for our purposes. We will focus only on differential equations where just one of the unknown function is differentiated. Moreover, we will adopt a different style of writing equations, where we isolate this derivative on one side. An example of a typical linear ODE with three unknown functions in the proper form could be this:

$$y_2' = 13y_1 + 14y_2 - 23y_3 + \cos(x).$$

Equations like these appear naturally in applications, so we are not too restrictive here. Having  $n$  unknown functions, it is natural to ask for  $n$  linearly independent equations, and we want each of them to have a different derivative on the left. In other words, for each unknown function  $y_i$  we expect to see one equation of the form  $y_i' = \dots$

Here is a typical example:

$$\begin{aligned}y_1' &= 2y_1 + y_2 - 3 \\y_2' &= y_1 + 2y_2 + 3x - 4.\end{aligned}$$

Since each equation features the derivative of a different function, these equations are obviously linearly independent.

We expect infinitely many solutions for such a system, and as usual we are interested in a general solution with parameters, namely  $n$  parameters in this case.

To determine a specific solution we therefore need  $n$  conditions. We will again focus on initial conditions at a given time  $x_0$ , and for equations of order one we asked just about the value of a function itself. This fits well with the new setting, since asking for a specific value at time  $x_0$  from each unknown function constitutes  $n$  conditions.

We will formalize our understanding.

### Definition 26.1.

By a **system of linear ODEs of order 1 with constant coefficients** we mean a system of the form

$$\begin{aligned}y_1' &= a_{11}y_1 + a_{12}y_2 + \dots + a_{1n}y_n + b_1(x) \\y_2' &= a_{21}y_1 + a_{22}y_2 + \dots + a_{2n}y_n + b_2(x) \\&\vdots \\y_n' &= a_{n1}y_1 + a_{n2}y_2 + \dots + a_{nn}y_n + b_n(x)\end{aligned}$$

where  $b_i(x)$  are right hand-sides and  $a_{ij} \in \mathbb{R}$  are coefficients.

The system is called **homogeneous** if  $b_i(x) = 0$  for all  $i = 1, \dots, n$ .

An Initial Value Problem (IVP) or Cauchy problem for such a system has initial conditions

$$y_1(x_0) = y_{1,0}, y_2(x_0) = y_{2,0}, \dots, y_n(x_0) = y_{n,0}.$$

Note that we allow only for constant coefficients here. It would be possible to work in a more general setting (and in fact structural theorems that follow would still be valid), but we would not know how to solve such systems anyway.

Just like with linear equations, the notation is a bit inconsistent in that we use the variable notation with  $b_i(x)$  but not with  $y_i$ . The reason for this is the same, we all know that the unknowns are functions so it is not necessary to remind ourselves of that, but we want to emphasize that the right-hand sides may depend on  $x$ .

One of the popular ways to solve small systems of algebraic linear equations is by intuitive elimination, where we gradually reduce the number of unknowns while also reducing the number of equations to be solved. The key step is to determine a certain unknown from one of the remaining equations and substitute this formula into the other equations.

This approach also works in our new setting.

**Example 26.a:** Consider the initial value problem

$$\begin{aligned}y_1' &= 2y_1 + y_2 - 3 & y_1(0) &= 3 \\y_2' &= y_1 + 2y_2 + 3x - 4, & y_2(0) &= 1.\end{aligned}$$

As usual, we start by deriving a general solution of the given system.

1. Looking at the equations, we can either isolate  $y_2$  from the first equation or  $y_1$  from the second. We will go with the former idea, obtaining a key reduction formula

$$y_2 = y_1' - 2y_1 + 3. \quad (\star)$$

We substitute this into the second equation, obtaining

$$\begin{aligned}[y_1' - 2y_1 + 3]' &= y_1 + 2[y_1' - 2y_1 + 3] + 3x - 4 \\y_1'' - 2y_1' + 0 &= y_1 + 2y_1' - 4y_1 + 6 + 3x - 4 \\y_1'' - 4y_1' + 3y_1 &= 3x + 2\end{aligned}$$

As expected, the number of equations and the number of unknowns were reduced by one, and we arrived at one linear differential equation of order two. This is perfectly all right, we are trading equations and unknowns for increased degree.

We do know how to solve the resulting equation, so briefly:

a) Solving the homogeneous system through its characteristic equation  $\lambda^2 - 4\lambda + 3 = 0$  we obtain a homogeneous solution  $y_{1h}(x) = a e^x + b e^{3x}$ .

b) We use the guessing method to find a particular solution. The initial form  $y_1(x) = Ax + B$  has special number  $\lambda = 0$  and therefore does not require any correction. We substitute this into our equation and obtain  $y_{1p}(x) = x + 2$ .

We arrive at the general solution  $y_1(x) = x + 2 + a e^x + b e^{3x}$  of the second-order linear ODE.

Now comes the back substitution stage. Having determined the last remaining unknown, we return to reduction formulas and determine the other unknowns. In this case we return to  $(\star)$  and obtain

$$\begin{aligned}y_2(x) &= [x + 2 + a e^x + b e^{3x}]' - 2[x + 2 + a e^x + b e^{3x}] + 3 \\&= -2x - a e^x + b e^{3x}.\end{aligned}$$

Now we have a general solution for our system:

$$\begin{aligned}y_1(x) &= x + 2 + a e^x + b e^{3x}, \\y_2(x) &= -2x - a e^x + b e^{3x}, \quad x \in \mathbb{R}.\end{aligned}$$

2. Knowing the functions  $y_1, y_2$ , we rewrite the given initial conditions:

$$\begin{aligned}0 + 2 + a e^0 + b e^0 &= 3, & a + b &= 1, \\-2 \cdot 0 - a e^0 + b e^0 &= 1 & \implies & -a + b = 1.\end{aligned}$$

This yields  $a = 0, b = 1$ . The solution of the given problem is

$$\begin{aligned}y_1(x) &= x + 2 + e^{3x}, \\y_2(x) &= -2x + e^{3x}, \quad x \in \mathbb{R}.\end{aligned}$$

Remark: If we substitute  $y_{1h}(x) = a e^x + b e^{3x}$  into the reduction formula ( $\star$ ), we obtain the function  $y_{2h}(x) = -a e^x + b e^{3x}$ . It is easy to confirm that the pair  $y_{1h}, y_{2h}$  actually solves the associated homogeneous system

$$\begin{aligned}y_1' &= 2y_1 + y_2 \\y_2' &= y_1 + 2y_2.\end{aligned}$$

$\triangle$

This worked out rather well, and it is a viable approach.

**Fact 26.2.**

Every system of  $n$  linear ODEs of order 1 can be transformed via elimination to one linear ODE of order  $n$ .

We restricted our attention to equations with just one derivative present. However, as mathematicians we can easily imagine other situations, for instance this system:

$$\begin{aligned}y_1' + 2y_2' &= 5y_1 - y_2 + 3e^x \\y_1' - y_2' &= 2y_1 + 5y_2 + 6x.\end{aligned}$$

Here the cure is simple, we apply the Gaussian elimination to the part of the system with derivatives. In fact, we will want to apply the Gauss-Jordan version (so it *is* good for something after all). We start by subtracting the first equation from the second.

$$\begin{aligned}y_1' + 2y_2' &= 5y_1 - y_2 + 3e^x \\-3y_2' &= -3y_1 + 6y_2 + 6x - 3e^x.\end{aligned}$$

Now we divide the second row by  $-3$ , and then subtract it twice from the first row.

$$\begin{aligned}y_1' &= 3y_1 + 3y_2 + 4x + e^x \\y_2' &= y_1 - 2y_2 - 2x + e^x.\end{aligned}$$

We arrived at a system of the form that we want to see here.

This idea can be generalized. We can simply consider systems of linear equations without any restriction on derivatives, and define the order of such a linear differential equation with more unknown functions as the highest degree of derivative we can see there. Then we have the following statement:

**Fact 26.3.**

Every system of  $n$  linear ODEs of orders  $n_i$  with  $n$  variables can be transformed via elimination to one linear ODE of order  $\sum n_i$ .

This sounds good, and one might think that this is could be the end of this chapter: We simply eliminate all systems to single linear equations of higher order and we know how to solve these. Unfortunately, it is not so simple when it comes to actual calculations.

**Example 26.b:** Consider the system

$$\begin{aligned}y_1' &= y_1 + 2y_2 + y_3 + 1 \\y_2' &= -y_1 + 2y_2 + 2y_3 \\y_3' &= 2y_1 + y_2 + y_3 + x.\end{aligned}$$

We can start with the first equation, it offers reduction formulas for  $y_2$  and  $y_3$ . The latter looks more appealing (no fractions will appear), so we go this way:

$$y_3 = y_1' - y_1 - 2y_2 - 1. \quad (\star)$$

We substitute this for  $y_3$  into the second and third equation, work out the derivative in the last one on the left, and the two remaining equations simplify to

$$\begin{aligned}y_2' - 2y_1' &= -3y_1 - 2y_2 - 2 \\y_1'' - 2y_1' - 2y_2' &= y_1 - y_2 + x - 1.\end{aligned}$$

None of these equations offers a way to express  $y_1$  or  $y_2$ . How do we go on?

Now we claimed above that some elimination is possible, but it is not intuitive any more, one needs to play with differential calculus for that, see section 27c. This negates the main advantage of elimination, namely its intuitive ease.

△

The system we played with here is not in any way special, it was a typical complication. The elimination works really swell for two-by-two systems, but it is not very friendly for larger systems. The preferred approach is to solve systems directly. In fact, instead of transforming systems into equations, people actually often go in the opposite way and transform single linear equations into systems.

**Example 26.c:** Consider the equation  $y''' + \sin(x)y'' - xy' + e^x y = \ln(x)$ . We start by introducing the nickname  $y_1$  for  $y$  and rewrite this equation as

$$y_1''' + \sin(x)y_1'' - xy_1' + e^x y_1 = \ln(x).$$

Note that now we have a  $1 \times 1$  system of linear ODEs, so we are on the right track, just the order does not fit.

We will therefore reduce it by hiding one derivative into a new unknown:

$$y_1' = y_2 \tag{*}$$

Note that in fact,  $y' = y_2$ , we also have  $y_2' = [y_1']' = y_1''$ ,  $y_2'' = [y_1'']' = y_1'''$  and so on, in other words,  $y_2$  really allows us to reduce the order of derivatives for  $y_1$ . We substitute into the given equation where possible and obtain

$$y_2'' + \sin(x)y_2' - xy_2 + e^x y_1 = \ln(x).$$

We successfully reduced the order of this equation, but we are not happy with the order yet, so we reduce again. We introduce  $y_2' = y_3$ , then also  $y_3' = y_2''$  and so on. We also note that  $y_3 = [y_2]' = [y_1']' = y_1''$ . In other word, the functions  $y_i$  give direct access to derivatives of the original unknown  $y$ . Substituting  $y_3$  into the latest equation where possible we obtain

$$y_3' + \sin(x)y_3 - xy_2 + e^x y_1 = \ln(x).$$

We are happy with the order now, so we stop our reduction. As the last step we gather all the reduction equations and the latest version of the reduced equation (rewritten properly) to obtain the system

$$\begin{aligned}y_1' &= && y_2 \\y_2' &= && y_3 \\y_3' &= -e^x y_1 + xy_2 - \sin(x)y_3 + \ln(x).\end{aligned}$$

This is a standard  $3 \times 3$  system.

Note that if the order of the original equation were higher, then we would introduce  $y_3' = y_4$ , and we would have  $y_4 = y_3''$ . You can surely see the pattern in the reduction equations. Consequently, the system created by the reduction formulas has a simple and regular form. Moreover, comparing coefficients of the last transformed equation with the given one we also see a pattern there. In fact, knowing these rules allows one to simply write down the resulting system just by looking at the given equation, without going through all the stages.

We also meet initial value problems. For instance, our equations could have the following conditions attached:  $y(0) = 1$ ,  $y'(0) = 13$ ,  $y''(0) = -1$ .

However, we observed above that the derivatives of the original unknown function are accessible through the new functions, so we easily rewrite these conditions as

$$\begin{aligned}y_1(0) &= 1 \\y_2(0) &= 13 \\y_3(0) &= -1.\end{aligned}$$

This is exactly the right form of initial conditions that we use for systems, so everything fits perfectly.

△

This simple algorithmic procedure has no hidden traps in it, which makes it simple and eminently suitable for computer algebra systems.

#### Algorithm 26.4.

⟨transforming linear equations into systems⟩

Given: a linear differential equation.

**0.** Denote the original unknown function as  $y_1$  and rewrite the given equation, obtaining the first transformed equation.

Set  $k = 1$ .

**1.** If the latest transformed equation is of order more than one, then

**a)** Introduce a new unknown function  $y_{k+1}$  to hide one derivative of  $y_k$  in it by the reduction formula  $y_{k+1} = y'_k$ .

**b)** Replace all derivatives of  $y_k$  in the transformed equation with derivatives of  $y_{k+1}$  in the obvious manner:  $y_k^{(i)} = y_{k+1}^{(i-1)}$ .

This results in a new transformed equation featuring functions  $y_1, \dots, y_k, y_{k+1}$  and with order reduced by one.

**c)** Increase  $k$  by 1 and return to step 1.

**2.** If the latest transformed equation is of order 1, compose the resulting system of equation by gathering all reduction formulas  $y'_1 = y_2, \dots, y'_{k-1} = y_k$  and adding the latest transformed equation to this list.

The algorithm stops.

**3.** If the given problem also features initial conditions of the form  $y^{(k)}(x_0) = y_k$ , we rewrite them using the fact that  $y^{(k)} = y_{k+1}$ .

△

If the given equation is

$$y^{(n)} + a_{n-1}(x)y^{(n-1)} + \dots + a_1(x)y' + a_0(x)y = b(x),$$

then the resulting system is

$$\begin{aligned}y'_1 &= && y_2 \\y'_2 &= && y_3 \\y'_3 &= && y_4 \\&\vdots && \\y'_n &= -a_0(x)y_1 - a_1(x)y_2 - \dots - a_{n-1}(x)y_n + b(x).\end{aligned}$$

This algorithm always succeeds, and this observation actually constitutes the proof of the following statement.

**Fact 26.5.**

Every linear ODE of order  $n$  (and every system of linear ODEs with sum of orders  $n$ ) can be equivalently transformed into a system of  $n$  linear ODEs of order 1 of the type we study here.

Which brings us to an interesting observation. It turns out that high order linear differential equations and systems of special order-one differential equations are connected in both directions, and knowing one gives the knowledge of the other. In chapter 15 we introduced some statements about linear differential equations, in particular an existence and uniqueness theorem. Using the correspondence we have now, we can easily transfer these statements also to systems of linear ODEs. However, it turns out that our special systems of order one are more tractable, and the preferred way is to transfer observation from systems to single equations of higher order. In particular, the existence and uniqueness result for linear differential equations is typically proved as a direct consequence of the existence and uniqueness result for systems of linear ODEs.

**Theorem 26.6.** (on existence and uniqueness for systems)

Consider a system of linear ODEs of order 1.

If  $b_i(x)$  are continuous on an open interval  $I$ , then for every  $x_0 \in I$  and all  $y_{1,0}, y_{2,0}, \dots, y_{n,0} \in \mathbb{R}$  there exists a solution of the corresponding IVP on  $I$  and it is unique.

Thus, to put our theory on a firm footing, we should now prove this statement. However, the proof is rather involved and we prefer to leave it to more advanced books.

We will now develop the preferred approach to solving systems of linear differential equations.

## 26a. Matrix approach

Just like in linear algebra, in order to handle systems well we will capture them using matrix and vector notation.

A system

$$\begin{aligned} y_1' &= a_{11}y_1 + a_{12}y_2 + \cdots + a_{1n}y_n + b_1(x) \\ y_2' &= a_{21}y_1 + a_{22}y_2 + \cdots + a_{2n}y_n + b_2(x) \\ &\vdots \\ y_n' &= a_{n1}y_1 + a_{n2}y_2 + \cdots + a_{nn}y_n + b_n(x) \end{aligned}$$

can be written as

$$\begin{pmatrix} y_1' \\ \vdots \\ y_n' \end{pmatrix} = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} y_1(x) \\ \vdots \\ y_n(x) \end{pmatrix} + \begin{pmatrix} b_1(x) \\ \vdots \\ b_n(x) \end{pmatrix}.$$

We will naturally denote  $\vec{y}(x) = \begin{pmatrix} y_1(x) \\ \vdots \\ y_n(x) \end{pmatrix}$  to have just one unknown (a vector now). Adopting

the convention that  $\begin{pmatrix} y_1' \\ \vdots \\ y_n' \end{pmatrix} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}'$ , we can write the given system as  $\vec{y}' = A\vec{y} + b(x)$ , where

$A = (a_{i,j})_{i,j=1}^n$  is the **matrix of the system** and  $\vec{b}(x) = (b_i)_{i=1}^n$  is the **vector of right-hand sides**.

The system is homogeneous if  $\vec{b} = \vec{0}$ , where  $\vec{0}$  is the zero vector in  $\mathbb{R}^n$ .

Moreover, if initial conditions are given, then they can be expressed as  $\vec{y}(x_0) = \vec{y}_0$ .

Again, we write  $\vec{b}(x)$  to emphasize that this is a vector of functions, although the notation looks a bit inconsistent.

**Example 26a.a:** Consider the initial value problem

$$\begin{aligned} y_1' &= 2y_1 + y_2 - 3 & y_1(0) &= 3 \\ y_2' &= y_1 + 2y_2 + 3x - 4, & y_2(0) &= 1. \end{aligned}$$

It can be written as  $\vec{y}' = A\vec{y} + \vec{b}(x)$ , where

$$\vec{y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}, \quad A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}, \quad \vec{b}(x) = \begin{pmatrix} -3 \\ 3x - 4 \end{pmatrix}.$$

In example 26.a we found a general solution of the form

$$\vec{y}(x) = \begin{pmatrix} x + 2 + a e^x + b e^{3x} \\ -2x - a e^x + b e^{3x} \end{pmatrix}.$$

Interestingly, it can be written as

$$\vec{y}(x) = \begin{pmatrix} x + 2 \\ -2x \end{pmatrix} + \begin{pmatrix} a e^x + b e^{3x} \\ -a e^x + b e^{3x} \end{pmatrix}.$$

We can easily check that

$$\vec{y}_p(x) = \begin{pmatrix} x + 2 \\ -2x \end{pmatrix}$$

solves the given system, while

$$\vec{y}_h(x) = \begin{pmatrix} a e^x + b e^{3x} \\ -a e^x + b e^{3x} \end{pmatrix} = a \begin{pmatrix} e^x \\ -e^x \end{pmatrix} + b \begin{pmatrix} e^{3x} \\ e^{3x} \end{pmatrix}$$

is a solution of the associated homogeneous system. In fact, each of the vectors that we see on the right is a solution of that homogeneous system, so we would guess that is it a basis of the space of homogeneous solutions.

This should not surprise us, as we expect to see the same behavior as we observed in chapter 15.

We also note that the homogeneous solution has two parameters, which suggests that the space of all solutions is two-dimensional.

△

This example shows what we expect to find when we treat a system of (linear) differential equations as a matrix-coded problem. All solutions will be vectors of functions, and we manipulate them in the usual way from linear algebra. We can easily return to the original point of view where we work with individual functions by reading entries in vectors of functions.

As usual, we first focus on the simpler case.

## 26b. Homogeneous systems

We start our exploration by confirming the expected facts regarding homogeneous systems.

**Theorem 26b.1.** (on structure of solution set for homogeneous systems)  
 Consider a homogeneous system of linear ODEs  $\vec{y}' = A\vec{y}$ , where  $A \in \mathbb{R}^{n \times n}$ .  
 The set of all solutions of this system on some open interval  $I$  is a linear space of dimension  $n$ .

The proof is fairly standard in its structure and content, and the convenient matrix notations makes it even simpler (cf. theorem 15.3).

**Proof:** 1. We will show that the set of all solutions on  $I$  (denoted  $M$ ) is a subset of the space of vector functions on  $I$  that is closed under the basic linear operation.

To this end, consider two solutions  $\vec{y}_1, \vec{y}_2 \in M$  and a scalar  $\alpha$ . Then we have

$$[\alpha\vec{y}_1 + \vec{y}_2]' = \alpha\vec{y}_1' + \vec{y}_2' = A\vec{y}_1 + A\vec{y}_2 = \vec{0} + \vec{0} = \vec{0}.$$

We see that also  $\alpha\vec{y}_1 + \vec{y}_2$  solves the given homogeneous system, therefore it belongs to  $M$ .

Consequently,  $M$  is a linear space.

2. Choose some  $x_0 \in I$ .

Take any  $i \in \{1, \dots, n\}$ . According to the existence theorem, there must be some solution  $\vec{y}_i$  of the given system that also satisfies the initial conditions  $y_i(x_0) = \vec{e}_i$ , where  $\vec{e}_i$  are the canonical unit vectors. In this way we obtain vector functions  $\{\vec{y}_1, \dots, \vec{y}_n\}$ . As solutions they definitely belong to  $M$ . We claim that they are linearly independent on  $I$ .

Indeed, consider some null linear combination  $\sum \alpha_i \vec{y}_i = \vec{0}$  on  $I$ . This must be in particular true at  $x_0$ , so  $\sum \alpha_i \vec{y}_i(x_0) = \vec{0}$ , that is,  $\sum \alpha_i \vec{e}_i = \vec{0}$ . Since the canonical basis  $\{\vec{e}_i\}$  consists of linearly independent vectors, we conclude that  $\alpha_i = 0$  for all  $i$ , in other words, only the trivial linear combination of  $\{\vec{y}_i\}$  can produce the zero vector.

We confirmed the desired linear independence, in particular the dimension of  $M$  is at least  $n$ .

Now take any solution  $\vec{y}_0$  of the given system. Then  $\vec{\beta} = \vec{y}_0(x_0)$  is some vector from  $\mathbb{R}^n$ , and this  $\vec{y}_0$  in fact solves the initial value problem given by the equation  $\vec{y}' = A\vec{y}$  and the initial condition  $\vec{y}(x_0) = \vec{\beta}$ .

Consider also the vector function  $\vec{y}_c = \sum \beta_i \vec{y}_i$ . This vector function also solves the system  $\vec{y}' = A\vec{y}$ . Moreover, we have

$$\vec{y}_c(x_0) = \sum \beta_i \vec{y}_i(x_0) = \sum \beta_i \vec{e}_i = \vec{\beta}.$$

Consequently,  $\vec{y}_c$  solves exactly the same initial value problem as  $\vec{y}_0$ . By the uniqueness theorem, these two must agree, that is,  $\vec{y}_0 = \sum \beta_i \vec{y}_i$ .

We just proved that the set  $\{\vec{y}_i\}$  generates the space  $M$ , so it is in fact a basis and  $\dim(M) = n$ .  $\square$

A basis of a solution space is naturally very important and there is special terminology related to it.

**Definition 26b.2.**

Consider a homogeneous system of linear ODEs  $\vec{y}' = A\vec{y}$ , where  $A \in \mathbb{R}^{n \times n}$ .

By a **fundamental system of solutions** of this system on an open interval  $I$  we mean an arbitrary basis of the space of all solutions of this system on  $I$ .

If  $\{\vec{y}_1, \dots, \vec{y}_n\}$  is a fundamental system of solutions, then we define its **fundamental matrix** on  $I$  by  $Y(x) = (\vec{y}_1(x) \ \cdots \ \vec{y}_n(x))$  (an  $n \times n$  matrix).

**Example 26b.a:** Consider the homogeneous system

$$\begin{aligned} y_1' &= 2y_1 + y_2 - 3 \\ y_2' &= y_1 + 2y_2 + 3x - 4. \end{aligned}$$

We found its general solution

$$\vec{y}_h(x) = a \begin{pmatrix} e^x \\ -e^x \end{pmatrix} + b \begin{pmatrix} e^{3x} \\ e^{3x} \end{pmatrix}$$

It would seem that there is a basis

$$\left\{ \begin{pmatrix} e^x \\ -e^x \end{pmatrix}, \begin{pmatrix} e^{3x} \\ e^{3x} \end{pmatrix} \right\}.$$

This would set up the fundamental matrix

$$Y(x) = \begin{pmatrix} e^x & e^{3x} \\ -e^x & e^{3x} \end{pmatrix}.$$

Notice an interesting thing:

$$\vec{y}_h(x) = \begin{pmatrix} e^x & e^{3x} \\ -e^x & e^{3x} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix}.$$

△

This last equality is sometimes very convenient.

**Fact 26b.3.**

Consider a homogeneous system of linear ODEs  $\vec{y}' = A\vec{y}$ , where  $A \in \mathbb{R}^{n \times n}$ . If  $Y(x)$  is its fundamental matrix on  $I$ , then a general solution of this system on  $I$  is  $\vec{y}_h(x) = Y(x) \cdot \vec{c}$  for  $\vec{c} \in \mathbb{R}^n$ .

There also is a convenient analogue of Wronski's result for determining linear independence of solutions.

**Theorem 26b.4.**

Consider a homogeneous system of linear ODEs  $\vec{y}' = A\vec{y}$ , where  $A \in \mathbb{R}^{n \times n}$ . Let  $\vec{y}_1, \dots, \vec{y}_n$  be solutions of this system on an open interval  $I$ .  $\{\vec{y}_1, \dots, \vec{y}_n\}$  is a fundamental system of solutions of this system on  $I$  if and only if  $\det(Y(x)) \neq 0$  on  $I$ , which is true if and only if  $\det(Y(x_0)) \neq 0$  for some  $x_0 \in I$ .

In short, any fundamental matrix  $Y(x)$  is nonsingular for all  $x \in I$ . We will need this observation in chapter 27.

We have the theory, and now for the practical part: Where do we get that fundamental system?

**Fact 26b.5.**

Consider a homogeneous system of linear ODEs  $\vec{y}' = A\vec{y}$  with matrix  $A \in \mathbb{R}^{n \times n}$ . If  $\lambda_0$  is an eigenvalue of  $A$  with associated eigenvector  $\vec{v}$ , then  $\vec{y} = \vec{v}e^{\lambda_0 x}$  is a solution of the given system on  $\mathbb{R}$ . If  $\lambda_1, \dots, \lambda_k$  are distinct eigenvalues of the matrix  $A$ , then the corresponding solutions form a linearly independent set.

If you need to refresh your memory regarding eigenvalues, see chapter 31.

**Example 26b.b:** Consider the homogeneous system

$$\begin{aligned} y_1' &= 2y_1 + y_2 \\ y_2' &= y_1 + 2y_2. \end{aligned}$$

We find the eigenvalues of the matrix of the system  $A = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$ .

We solve the appropriate equation:

$$\begin{aligned} 0 &= \det(A - \lambda E_n) = \det \begin{vmatrix} 2 - \lambda & 1 \\ 1 & 2 - \lambda \end{vmatrix} = (2 - \lambda)^2 - 1^2 \\ &= \lambda^2 - 4\lambda + 3. \end{aligned}$$

We obtain the eigenvalues  $\lambda = 1, 3$ .

Now we find the associated eigenvectors.

$\lambda = 1$ : We need to solve the homogeneous system  $(A - \lambda E_n)\vec{v} = \vec{0}$ , that is,

$$\begin{pmatrix} 2-1 & 1 \\ 1 & 2-1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

This reads

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

and we see that the equations are linearly dependent, as they should be. It is enough to solve the first equation  $v_1 + v_2 = 0$ . We can choose one variable, but we have to make sure that the resulting vector will not be trivial, so we have to choose a non-zero number. We set  $v_1 = 1$  and obtain  $v_2 = -1$ .

We have the pair  $\lambda = 1$ ,  $\vec{v} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$  and according to the statement above, we can create one vector of functions for our basis:

$$\vec{y}_a = \begin{pmatrix} 1 \\ -1 \end{pmatrix} e^{1 \cdot x} = \begin{pmatrix} e^x \\ -e^x \end{pmatrix}.$$

$\lambda = 3$ : We need to solve the homogeneous system with the matrix

$$\left( \begin{array}{cc|c} 2-3 & 1 & 0 \\ 1 & 2-3 & 0 \end{array} \right) = \left( \begin{array}{cc|c} -1 & 1 & 0 \\ 1 & -1 & 0 \end{array} \right).$$

Again, the rows are linearly dependent, which suggests that perhaps we did not make any mistake yet, and therefore we just look at one equation, for instance the first:  $-v_1 + v_2 = 0$ . Choosing  $v_1 = 1$  we get  $v_2 = 1$  and obtain the pair  $\lambda = 3$ ,  $\vec{v} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ . The second vector of functions is

$$\vec{y}_b = \begin{pmatrix} 1 \\ 1 \end{pmatrix} e^{3 \cdot x} = \begin{pmatrix} e^{3x} \\ e^{3x} \end{pmatrix}.$$

Now we can form a general solution:

$$\vec{y} = a\vec{y}_a + b\vec{y}_b = a \begin{pmatrix} e^x \\ -e^x \end{pmatrix} + b \begin{pmatrix} e^{3x} \\ e^{3x} \end{pmatrix},$$

that is,

$$\begin{pmatrix} y_1(x) \\ y_2(x) \end{pmatrix} = \begin{pmatrix} a e^x + b e^{3x} \\ -a e^x + b e^{3x} \end{pmatrix}, \quad x \in \mathbb{R}.$$

It is nice to write the answer in the same language as the question, so we shed the matrix overcoat and put it like this:

$$\begin{aligned} y_1(x) &= a e^x + b e^{3x}, \\ y_2(x) &= -a e^x + b e^{3x}, \quad x \in \mathbb{R}. \end{aligned}$$

This is the same answer that we obtained in example 26.a.

Remark: Note that the equation  $\lambda^2 - 4\lambda - 3 = 0$  that yielded the eigenvalues is exactly the same as the equation that provided us with characteristic numbers when we were solving the corresponding problem using elimination, and thus also the characteristic numbers were the same as the eigenvalues now. This is no surprise.

The polynomial  $\lambda^2 - 4\lambda - 3$  and the numbers  $\lambda = 1, 3$  capture the substance of the underlying situation (from physics for instance). When we transform a system into one equation or vice versa, we are just changing the language of the mathematical description, but the underlying process is still the same, and therefore the lambdas also stay the same.

△

The procedure that we just tried works for all systems of linear differential equations that have

distinct real eigenvalues. To make our knowledge complete we have to find out how to handle other cases. For complex eigenvalues we have the usual answer.

**Fact 26b.6.**

Consider a homogeneous system of linear ODEs  $\vec{y}' = A\vec{y}$  with matrix  $A \in \mathbb{R}^{n \times n}$ . Let  $\lambda_0$  be an eigenvalue of  $A$  with associated eigenvector  $\vec{v}$ . If  $\lambda_0$  is a complex number, that is,  $\text{Im}(\lambda_0) \neq 0$ , then  $\text{Re}(\vec{v}e^{\lambda_0 x})$  and  $\text{Im}(\vec{v}e^{\lambda_0 x})$  are linearly independent solutions of the given system on  $\mathbb{R}$ .

We used the same approach with linear equations of higher order, and there we obtained very convenient formulas using this principle. Indeed, from a complex solution  $e^{(\alpha+\beta i)x}$  we recovered  $e^{\alpha x} \cos(\beta x)$  and  $e^{\alpha x} \sin(\beta x)$ . For systems of equations the situation is not so simple, because for a complex eigenvalue  $\lambda$  and a real-valued matrix, the associated eigenvector  $\vec{v}$  must also have some complex components, and thus we cannot readily identify what the real and imaginary parts of  $\vec{v}e^{\lambda x}$  are in general.

**Example 26b.c:** Consider the oscillation equation  $\ddot{x} + \omega^2 x = 0$ . We transform it into a system of equation by denoting  $\dot{x} = v$ . We chose this name because we know that if  $x$  is the position of, say, a pendulum (angle or displacement), then  $\dot{x}$  is the velocity. We obtain the system

$$\begin{aligned}\dot{x} &= v \\ \dot{v} &= -\omega^2 x.\end{aligned}$$

The matrix of this system is  $\begin{pmatrix} 0 & 1 \\ -\omega^2 & 0 \end{pmatrix}$ . We find the eigenvalues:

$$0 = \det \begin{pmatrix} -\lambda & 1 \\ -\omega^2 & -\lambda \end{pmatrix} = \lambda^2 + \omega^2.$$

The roots are  $\lambda = \pm \omega i$ . Now we find an eigenvector associated with  $\lambda = \omega i$ :

$$\left( \begin{array}{cc|c} 0 - \omega i & 1 & 0 \\ -\omega^2 & 0 - \omega i & 0 \end{array} \right) = \left( \begin{array}{cc|c} -\omega i & 1 & 0 \\ -\omega^2 & -\omega i & 0 \end{array} \right)$$

We note that the second row is just the first one multiplied by  $-\omega i$ , so again it is enough to solve just one equation in this system, for instance the first one:  $-\omega i v_1 + v_2 = 0$ . We choose  $v_1 = 1$  and obtain  $v_2 = \omega i$ . The eigenvector is  $\begin{pmatrix} 1 \\ \omega i \end{pmatrix}$  and the corresponding solution is

$$\begin{aligned}\vec{y} &= \begin{pmatrix} 1 \\ \omega i \end{pmatrix} e^{\omega i t} = \begin{pmatrix} 1 \\ \omega i \end{pmatrix} (\cos(\omega t) + i \sin(\omega t)) \\ &= \begin{pmatrix} \cos(\omega t) + i \sin(\omega t) \\ \omega i \cos(\omega t) - \omega \sin(\omega t) \end{pmatrix}.\end{aligned}$$

We used  $t$  for the free variable, as  $x$  is already taken.

Now we take the real and imaginary parts to obtain two vectors for our fundamental system.

$$\vec{y}_a = \begin{pmatrix} \cos(\omega t) \\ -\omega \sin(\omega t) \end{pmatrix}, \quad \vec{y}_b = \begin{pmatrix} \sin(\omega t) \\ \omega \cos(\omega t) \end{pmatrix}.$$

As usual, there is no need to work out the case  $\lambda = -\omega i$ , since it would yield the same solutions.

We obtain a general solution of the form

$$\begin{aligned}\begin{pmatrix} x(t) \\ v(t) \end{pmatrix} &= a \begin{pmatrix} \cos(\omega t) \\ -\omega \sin(\omega t) \end{pmatrix} + b \begin{pmatrix} \sin(\omega t) \\ \omega \cos(\omega t) \end{pmatrix} \\ &= \begin{pmatrix} a \cos(\omega t) + b \sin(\omega t) \\ -a\omega \sin(\omega t) + b\omega \cos(\omega t) \end{pmatrix}.\end{aligned}$$

We rewrite it properly:

$$\begin{aligned}x(t) &= a \cos(\omega t) + b \sin(\omega t), \\v(t) &= -a\omega \sin(\omega t) + b\omega \cos(\omega t), \quad t \in \mathbb{R}.\end{aligned}$$

Going back to the original equation, its general solution is

$$x(t) = a \cos(\omega t) + b \sin(\omega t), \quad t \in \mathbb{R}.$$

As a bonus for solving it through a system we also obtained information about velocity. Note that our transformation requires that  $v = x'$  and we see that our solution satisfies this, which gives us hope that we did not make any mistake when solving it.

△

Eigenvalues of higher multiplicity are a bit more involved.

**Fact 26b.7.**

Consider a homogeneous system of linear ODEs  $\vec{y}' = A\vec{y}$  with matrix  $A \in \mathbb{R}^{n \times n}$ . Let  $\lambda_0$  be an eigenvalue of  $A$  of multiplicity  $m$  with associated eigenvector  $\vec{v}$ .

Consider vectors defined as follows:

$$\begin{aligned}\vec{v}_1 &= \vec{v}, \text{ so it is a solution of } (A - \lambda_0 E_n)\vec{x} = \vec{0}, \\ \vec{v}_2 &\text{ is a solution of } (A - \lambda_0 E_n)\vec{x} = \vec{v}_1, \\ \vec{v}_3 &\text{ is a solution of } (A - \lambda_0 E_n)\vec{x} = \vec{v}_2, \\ &\vdots \\ \vec{v}_m &\text{ is a solution of } (A - \lambda_0 E_n)\vec{x} = \vec{v}_{m-1}.\end{aligned}$$

Then the following functions are solutions of the given system on  $\mathbb{R}$  and form a linearly independent set:

$$\begin{aligned}\vec{y} &= \vec{v}_1 e^{\lambda_0 x}, \\ \vec{y} &= \left[ \int (\vec{v}_1) dx + \vec{v}_2 \right] e^{\lambda_0 x} = (\vec{v}_1 x + \vec{v}_2) e^{\lambda_0 x}, \\ \vec{y} &= \left[ \int (\vec{v}_1 x + \vec{v}_2) dx + \vec{v}_3 \right] e^{\lambda_0 x} = \left( \frac{1}{2} \vec{v}_1 x^2 + \vec{v}_2 x + \vec{v}_3 \right) e^{\lambda_0 x}, \\ &\vdots \\ \vec{y} &= \left( \frac{1}{(m-1)!} \vec{v}_1 x^{m-1} + \frac{1}{(m-2)!} \vec{v}_2 x^{m-2} + \cdots + \vec{v}_{m-1} x + \vec{v}_m \right) e^{\lambda_0 x}.\end{aligned}$$

This is definitely more complicated than the corresponding procedure for higher order linear differential equations. The vectors  $\vec{v}_i$  are called generalized eigenvectors.

**Example 26b.d:** Consider the initial value problem

$$\begin{aligned}y_1' &= y_1 - y_2 & y_1(0) &= 13 \\ y_2' &= y_1 + 3y_2, & y_2(0) &= 23.\end{aligned}$$

1. First we find its general solution. The matrix of the system is  $\begin{pmatrix} 1 & -1 \\ 1 & 3 \end{pmatrix}$  and we will find its eigenvalues:

$$\begin{aligned}0 &= \det \begin{pmatrix} 1 - \lambda & -1 \\ 1 & 3 - \lambda \end{pmatrix} = (1 - \lambda)(3 - \lambda) - 1 \cdot (-1) \\ &= \lambda^2 - 4\lambda + 4 = (\lambda - 2)^2.\end{aligned}$$

We see that  $\lambda = 2$  is an eigenvalue of multiplicity 2.

We will find the necessary chain of generalized eigenvectors, working with the matrix

$$A - 2E_n = \begin{pmatrix} -1 & -1 \\ 1 & 1 \end{pmatrix}.$$

The first one is just the ordinary eigenvector, so we look at the system

$$\left( \begin{array}{cc|c} -1 & -1 & 0 \\ 1 & 1 & 0 \end{array} \right).$$

The second equation reads  $v_1 + v_2 = 0$ , choosing  $v_1 = 1$  we get  $v_2 = -1$  and we have the first vector in the chain  $\vec{v}_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ .

We now use this vector as the right-hand side for our system:

$$\left( \begin{array}{cc|c} -1 & -1 & 1 \\ 1 & 1 & -1 \end{array} \right).$$

The rows are linearly dependent again, so it is enough to solve one equation, for instance the second:  $v_1 + v_2 = 1$ . Note that now we can choose  $v_1 = 0$ , since the equation is no longer homogeneous and thus the resulting vector will not be trivial. We get  $v_2 = 1$  and hence  $\vec{v}_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ .

Now we form two solutions according to the theorem above.

$$\begin{aligned} \vec{y}_a &= \begin{pmatrix} 1 \\ -1 \end{pmatrix} e^{2x} = \begin{pmatrix} e^{2x} \\ -e^{2x} \end{pmatrix}, \\ \vec{y}_b &= \left[ x \begin{pmatrix} 1 \\ -1 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right] e^{2x} = \begin{pmatrix} x e^{2x} \\ (-x - 1)e^{2x} \end{pmatrix}. \end{aligned}$$

This is enough to form a basis, so we write a general solution:

$$\vec{y}(x) = a \begin{pmatrix} e^{2x} \\ -e^{2x} \end{pmatrix} + b \begin{pmatrix} x e^{2x} \\ -(x + 1)e^{2x} \end{pmatrix} = \begin{pmatrix} a e^{2x} + b x e^{2x} \\ -a e^{2x} - b(x + 1)e^{2x} \end{pmatrix}.$$

As usual, we prefer the form

$$\begin{aligned} y_1(x) &= a e^{2x} + b x e^{2x}, \\ y_2(x) &= -a e^{2x} - b(x + 1)e^{2x}, \quad x \in \mathbb{R}. \end{aligned}$$

2. Now we address the initial conditions. Knowing the formulas for  $y_1$  and  $y_2$ , we can write them as

$$\begin{aligned} a &= 13 \\ -a - b &= 23 \end{aligned} \implies a = 13, \quad b = -36.$$

The solution is

$$\begin{aligned} y_1(x) &= 13e^{2x} - 36x e^{2x}, \\ y_2(x) &= -13e^{2x} + 36(x + 1)e^{2x}, \quad x \in \mathbb{R}. \end{aligned}$$

△

And that's it, folks. Now we can solve any homogeneous system, that is, as long as we can find the eigenvalues. Indeed, once the matrix is larger than  $4 \times 4$ , then determining eigenvalues in the usual way would require being able to solve polynomials of such high degrees, and we already know that there are no formulas for that.

It would be possible to find roots of characteristic polynomials numerically, but the preferred way is to approximate the eigenvalues directly. The chapter 31 does just that.

## 27. Nonhomogeneous systems of linear differential equations

We will follow the usual pattern. First, we introduce a structural theorem that shifts the burden of finding a complete solution to the homogeneous case.

**Theorem 27.1.** (on structure of solution set of systems of linear ODE)  
 Let  $\vec{y}_p$  be some particular solution of a given system of linear ODEs on an open interval  $I$ .  
 A vector function  $\vec{y}_0$  is a solution of this equation on  $I$  if and only if  
 $\vec{y}_0 = \vec{y}_p + \vec{y}_h$  for some solution  $\vec{y}_h$  of the associated homogeneous system in  $I$ .  
 Consequently, if  $\vec{y}_h$  is a general solution of the associated homogeneous system on  $I$ , then  $\vec{y}_p + \vec{y}_h$  is a general solution of the given equation on  $I$ .

The proof follows along the same lines as we saw when proving the analogous statement for high order linear equations, see theorem 16.2.

**Proof:** Assume that  $\vec{y}_p$  is a solution of the given system  $\vec{y}' = A\vec{y} + \vec{b}(x)$  on  $I$ , denote the system (S).

1) Let  $\vec{y}_h$  be some homogeneous solution on  $I$ , then  $\vec{y}_h' = A\vec{y}_h$ . We claim that  $\vec{y} = \vec{y}_p + \vec{y}_h$  is a solution of (S) on  $I$ . Indeed, for any  $x \in I$  we get

$$\begin{aligned} [\vec{y}_p + \vec{y}_h]'(x) &= \vec{y}_p'(x) + \vec{y}_h'(x) = (A\vec{y}_p(x) + \vec{b}(x)) + A\vec{y}_h(x) \\ &= A(\vec{y}_p + \vec{y}_h)(x) + \vec{b}(x). \end{aligned}$$

2) Let  $\vec{y}_0$  be any solution of (S). Consider the vector function  $\vec{y}_h = \vec{y}_0 - \vec{y}_p$ . Then obviously  $\vec{y}_p + \vec{y}_h = \vec{y}_0$  and  $\vec{y}_h$  solves the associated homogeneous system in  $I$ . Indeed,

$$\begin{aligned} \vec{y}_h'(x) &= [\vec{y}_0 - \vec{y}_p]'(x) = \vec{y}_0'(x) - \vec{y}_p'(x) = (A\vec{y}_0(x) + \vec{b}(x)) - (A\vec{y}_p(x) + \vec{b}(x)) \\ &= A(\vec{y}_0 - \vec{y}_p)(x) = A\vec{y}_h(x). \end{aligned}$$

The claim is proved. □

Now it remains to figure out how to get one particular solution of the given system.

### 27a. Variation of parameter

For nonhomogeneous systems of linear differential equations, the variation of parameters is the preferred method, especially where computer algebra systems are concerned.

We start by recalling the variation as applied to first order linear differential equations, see algorithm 9.5. It turns out that it carries over to first-order systems as well. We will show it on an example first and then express it in an algorithm.

**Example 27a.a:** We return to our favorite example.

$$\begin{aligned} y_1' &= 2y_1 + y_2 - 3 \\ y_2' &= y_1 + 2y_2 + 3x - 4. \end{aligned}$$

The variation procedure calls for solving the associated homogeneous equation first, so we should probably interpret this as solving the homogeneous system now. Moreover, the solution should be written as  $c \cdot u(x)$ .

We actually solved the homogeneous system in example 26b.a and we obtained

$$\begin{aligned} y_{1h}(x) &= a e^x + b e^{3x}, \\ y_{2h}(x) &= a(-e^x) + b e^{3x}. \end{aligned}$$

The solutions are of the form  $y_i(x) = c_u u_i(x) + c_v v_i(x)$ , which is close enough to the original variation. Following the variation way, we should start looking for a solution of the given system

in the form

$$\begin{aligned}y_{1p}(x) &= a(x)e^x + b(x)e^{3x}, \\y_{2p}(x) &= a(x)(-e^x) + b(x)e^{3x}.\end{aligned}$$

The original variation then offered two possibilities. We could substitute the new guess into the original equation, experience a miracle of cancelling and obtain an equation for  $c'(x)$ , or simply remember that this equation is  $c'(x)u(x) = b(x)$ .

We will see what happens when we substitute our guess into the given system.

$$\begin{aligned}[a(x)e^x + b(x)e^{3x}]' &= 2(a(x)e^x + b(x)e^{3x}) + (-a(x)e^x + b(x)e^{3x}) - 3 \\[-a(x)e^x + b(x)e^{3x}]' &= (a(x)e^x + b(x)e^{3x}) + 2(-a(x)e^x + b(x)e^{3x}) + 3x - 4.\end{aligned}$$

This reads

$$\begin{aligned}a'(x)e^x + a(x)e^x + b'(x)e^{3x} + b(x)3e^{3x} &= a(x)e^x + 3b(x)e^{3x} - 3 \\-a'(x)e^x - a(x)e^x + b'(x)e^{3x} + b(x)3e^{3x} &= -a(x)e^x + 3b(x)e^{3x} + 3x - 4,\end{aligned}$$

that is,

$$\begin{aligned}a'(x)e^x + b'(x)e^{3x} &= -3 \\a'(x)(-e^x) + b'(x)e^{3x} &= 3x - 4.\end{aligned}$$

Our belief in miracles got rewarded again, and all  $a(x)$  and  $b(x)$  disappeared, leaving us with two equations featuring two unknown functions  $a'(x)$ ,  $b'(x)$ ; this sounds right. By the way, note that if we write the homogeneous solution as  $y_i(x) = c_u u_i(x) + c_v v_i(x)$ , we can see the equations that we deduced now as  $c'_u(x)u_i(x) + c'_v(x)v_i(x) = b_i(x)$ , which fits remarkably well with the pattern of variation for one equation.

Now we solve the system. One can use the Cramer rule (which is actually rather convenient when it comes to systems with functions instead of numbers). Elimination is also a good bet, both in the formal shape of Gaussian elimination and in its intuitive forms.

We will go the easy way and observe that adding the equations gets us rid of  $a'(x)$ . We obtain

$$2b'(x)e^{3x} = 3x - 7 \implies b'(x) = \frac{1}{2}(3x - 7)e^{-3x}.$$

Subtracting the second equation from the first we obtain

$$2a'(x)e^x = -3x + 1 \implies a'(x) = \frac{1}{2}(-3x + 1)e^{-x}.$$

The last major step is integration, here we have to use integration by parts.

$$\begin{aligned}a(x) &= \int \frac{1}{2}(-3x + 1)e^{-x} dx = -\frac{1}{2}(-3x + 1)e^{-x} - \frac{1}{2}(-3)e^{-x} \\&= \left(\frac{3}{2}x + 1\right)e^{-x}, \\b(x) &= \int \frac{1}{2}(3x - 7)e^{-3x} dx = -\frac{1}{3}\frac{1}{2}(3x - 7)e^{-3x} - \frac{1}{9}\frac{3}{2}e^{-3x} \\&= \left(\frac{1}{2}x + 1\right)e^{-3x}.\end{aligned}$$

We found functions  $a(x)$ ,  $b(x)$  and we can substitute them into our guess for a particular solution.

$$\begin{aligned}y_{1p}(x) &= \left(\frac{3}{2}x + 1\right)e^{-x}e^x + \left(\frac{1}{2}x + 1\right)e^{-3x}e^{3x} \\&= \frac{3}{2}x + 1 + \frac{1}{2}x + 1 = 2x + 2, \\y_{2p}(x) &= -\left(\frac{3}{2}x + 1\right)e^{-x}e^x + \left(\frac{1}{2}x + 1\right)e^{-3x}e^{3x} \\&= -\frac{3}{2}x - 1 + \frac{1}{2}x + 1 = -x.\end{aligned}$$

By a remarkable coincidence, these are exactly the same results that we obtained using elimination in example 26.a. It is just a coincidence, given that this problem has infinitely many particular solutions that we choose from, but it often happens that different methods find the same particular solution, and we know now that this solution is correct.

We will complete this solution by writing a general solution of our problem using the  $\vec{y}_p + \vec{y}_h$

structure:

$$\begin{aligned}y_1(x) &= x + 2 + a e^x + b e^{3x}, \\y_2(x) &= -x - a e^x + b e^{3x}, \quad x \in \mathbb{R}.\end{aligned}$$

Remark: We committed the crime of leaving out integration constants in our integrations above. It was a premeditated crime, since we knew that we were looking for just one particular solution, so any antiderivative would do.

There is an alternative approach. Like good boys and girls (or other intelligent galactic creatures), we could include integration constants like this:

$$\begin{aligned}a(x) &= \int \frac{1}{2}(-3x + 1)e^{-x} dx = \left(\frac{3}{2}x + 1\right)e^{-x} + A, \\b(x) &= \int \frac{1}{2}(3x - 7)e^{-3x} dx = \left(\frac{1}{2}x + 1\right)e^{-3x} + B.\end{aligned}$$

Now we substitute these into the variational guess:

$$\begin{aligned}y_{1p}(x) &= \left[\left(\frac{3}{2}x + 1\right)e^{-x} + A\right]e^x + \left[\left(\frac{1}{2}x + 1\right)e^{-3x} + B\right]e^{3x} \\&= 2x + 2 + A e^x + B e^{3x}, \\y_{2p}(x) &= \left[-\left(\frac{3}{2}x + 1\right)e^{-x} + A\right]e^x + \left[\left(\frac{1}{2}x + 1\right)e^{-3x} + B\right]e^{3x} \\&= -x - A e^x + B e^{3x}.\end{aligned}$$

In this way we obtain a general solution directly.

We will prefer the first approach here because it ties in with our structural approach and the theory, but this alternative works equally well.

△

The procedure that we outlined above works in general. It is a hands-on version of variation, when we treat the system through individual equations and individual functions. In fact, it is my preferred approach if I am to solve systems by hand.

Recall that a general homogeneous solution can be found in the form  $\vec{y}_h = \sum c_j \vec{u}_j$ , where  $\vec{u}_j$  are vectors of functions that solve the associated homogeneous equation. Each  $\vec{u}_j$  is therefore a vector whose  $i$ th entry is used when creating the solution  $y_{ih}$ . To avoid double indexing, we will write the general solution as

$$\vec{y}_h = c_1 \vec{u} + c_2 \vec{v} + c_3 \vec{w} + \cdots$$

and then we find the functions  $y_i$  in rows of the resulting vector:  $\vec{y}_{ih} = c_1 \vec{u}_i + c_2 \vec{v}_i + c_3 \vec{w}_i + \cdots$ .

### Algorithm 27a.1.

⟨variation of parameters method—row version⟩

Given: a system  $\vec{y}' = A\vec{y} + \vec{b}(x)$ , with  $A$  a real-valued  $n \times n$  matrix.

1. Find a general solution  $\vec{y}_h$  of the associated homogeneous system  $\vec{y}' = A\vec{y}$ , and express it as

$$\begin{aligned}y_{1h}(x) &= c_1 u_1(x) + c_2 v_1(x) + c_3 w_1(x) + \cdots, \\y_{2h}(x) &= c_1 u_2(x) + c_2 v_2(x) + c_3 w_2(x) + \cdots, \\&\vdots \\y_{nh}(x) &= c_1 u_n(x) + c_2 v_n(x) + c_3 w_n(x) + \cdots.\end{aligned}$$

2. Seek a solution of the form

$$\begin{aligned}y_1(x) &= c_1(x)u_1(x) + c_2(x)v_1(x) + c_3(x)w_1(x) + \cdots, \\&\vdots \\y_n(x) &= c_1(x)u_n(x) + c_2(x)v_n(x) + c_3(x)w_n(x) + \cdots.\end{aligned}$$

Unknown functions  $c_i(x)$  are found by solving the system of equations

$$c'_1(x)u_1(x) + c'_2(x)v_1(x) + c'_3(x)w_n(x) + \cdots = b_1(x),$$

$$\vdots$$

$$c'_1(x)u_n(x) + c'_2(x)v_n(x) + c'_3(x)w_n(x) + \cdots = b_n(x).$$

Solve this system of equations (using e.g. elimination or Cramer rule) for  $c'_1(x), \dots, c'_n(x)$ .

Use integration to find some antiderivatives  $c_1(x), \dots, c_n(x)$ .

Substitute these into modified  $y_1, \dots, y_n$  to get  $y_{1p}, \dots, y_{np}$ .

**3.** The general solution is  $y_i = y_{ip} + y_{ih}$  for  $i = 1, \dots, n$ .

△

The row-version of variation may be nice for human computers, but it is less convenient to silicon-based computers. They would definitely prefer an abstract, matrix-coded version.

### Algorithm 27a.2.

⟨variation of parameters method—matrix version⟩

Given: a system  $\vec{y}' = A\vec{y} + \vec{b}(x)$ , with  $A$  a real-valued  $n \times n$  matrix.

**1.** Find a general solution  $\vec{y}_h$  of the associated homogeneous system  $\vec{y}' = A\vec{y}$ , and express it as  $\vec{y} = Y(x) \cdot \vec{c}$ .

**2.** Seek a solution of the form  $\vec{y}_p = Y(x) \cdot \vec{c}(x)$ .

Substituting it into the original equation one gets the equation  $Y(x) \cdot \vec{c}'(x) = \vec{b}(x)$ .

Then  $\vec{c}'(x) = Y(x)^{-1}\vec{b}(x)$ . Use integration to obtain  $\vec{c}(x)$  and substitute it into  $\vec{y}_p(x) = Y(x) \cdot \vec{c}(x)$ .

**3.** A general solution is then  $\vec{y} = \vec{y}_p + \vec{y}_h$ .

△

Note how variation in this matrix point of view strongly resembles variation for one equation in algorithm 9.5. Moreover, all the operations in this procedure can be easily handled by any competent computer algebra system.

In fact, the matrix approach is nothing else but the row approach where we hide the unpleasant details using a convenient notation. To see this we revisit our example.

**Example 27a.b:** Now we consider the system

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix}' = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} + \begin{pmatrix} -3 \\ 3x - 4 \end{pmatrix}.$$

We will write  $A$  for the system matrix and  $\vec{b}$  for the right-hand side vector.

In example 26b.a we found a solution in the matrix form

$$\vec{y}_h(x) = \begin{pmatrix} e^x & e^{3x} \\ -e^x & e^{3x} \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = Y(x)\vec{c}.$$

Now we start variation, and focus on vector functions of the form

$$\vec{y}_p(x) = Y(x)\vec{c}(x) = \begin{pmatrix} e^x & e^{3x} \\ -e^x & e^{3x} \end{pmatrix} \begin{pmatrix} a(x) \\ b(x) \end{pmatrix}.$$

According to algorithm, we should now solve the equation

$$Y(x)\vec{c}'(x) = \vec{b}(x).$$

When we look at the contents of  $Y(x)$  and  $\vec{c}(x)$ , we find that these are exactly the equations that we obtained for  $a'(x)$  and  $b'(x)$  when solving this problem via row-variation in the example above. But this time we handle it using matrix tools and derive the following formula.

$$\vec{c}'(x) = Y(x)^{-1}\vec{b}(x).$$

We find the inverse matrix in the usual way.

$$\begin{aligned} \left( \begin{array}{cc|cc} e^x & e^{3x} & 1 & 0 \\ -e^x & e^{3x} & 0 & 1 \end{array} \right) &\sim \left( \begin{array}{cc|cc} e^x & e^{3x} & 1 & 0 \\ 0 & 2e^{3x} & 1 & 1 \end{array} \right) \sim \left( \begin{array}{cc|cc} e^x & e^{3x} & 1 & 0 \\ 0 & e^{3x} & \frac{1}{2} & \frac{1}{2} \end{array} \right) \\ &\sim \left( \begin{array}{cc|cc} e^x & 0 & \frac{1}{2} & -\frac{1}{2} \\ 0 & e^{3x} & \frac{1}{2} & \frac{1}{2} \end{array} \right) \sim \left( \begin{array}{cc|cc} 1 & 0 & \frac{1}{2}e^{-x} & -\frac{1}{2}e^{-x} \\ 0 & 1 & \frac{1}{2}e^{-3x} & \frac{1}{2}e^{-3x} \end{array} \right). \end{aligned}$$

We see the inverse matrix on the right, now we can find  $\vec{c}'(x)$ :

$$\begin{aligned} \begin{pmatrix} a'(x) \\ b'(x) \end{pmatrix} &= \begin{pmatrix} \frac{1}{2}e^{-x} & -\frac{1}{2}e^{-x} \\ \frac{1}{2}e^{-3x} & \frac{1}{2}e^{-3x} \end{pmatrix} \cdot \begin{pmatrix} -3 \\ 3x-4 \end{pmatrix} \\ &= \begin{pmatrix} -3\frac{1}{2}e^{-x} - (3x-4)\frac{1}{2}e^{-x} \\ -3\frac{1}{2}e^{-3x} + (3x-4)\frac{1}{2}e^{-3x} \end{pmatrix} \\ &= \begin{pmatrix} (-\frac{3}{2}x + \frac{1}{2})e^{-x} \\ (\frac{3}{2}x - \frac{7}{2})e^{-3x} \end{pmatrix}. \end{aligned}$$

Note that the calculations themselves were exactly as with the row variation, as well as the outcome and the integration that we now face. We repeat it to obtain

$$\vec{c}(x) = \begin{pmatrix} a(x) \\ b(x) \end{pmatrix} = \begin{pmatrix} (\frac{3}{2}x + 1)e^{-x} \\ (\frac{1}{2}x + 1)e^{-3x} \end{pmatrix}.$$

Now we can substitute this into our guess for  $\vec{y}_p$  to obtain

$$\begin{aligned} \vec{y}_p(x) &= Y(x)\vec{c}(x) = \begin{pmatrix} e^x & e^{3x} \\ -e^x & e^{3x} \end{pmatrix} \begin{pmatrix} (\frac{3}{2}x + 1)e^{-x} \\ (\frac{1}{2}x + 1)e^{-3x} \end{pmatrix} \\ &= \begin{pmatrix} (\frac{3}{2}x + 1)e^{-x}e^x + (\frac{1}{2}x + 1)e^{-3x}e^{3x} \\ -(\frac{3}{2}x + 1)e^{-x}e^x + (\frac{1}{2}x + 1)e^{-3x}e^{3x} \end{pmatrix} \\ &= \begin{pmatrix} 2x + 2 \\ -x \end{pmatrix}. \end{aligned}$$

Again, the calculations were exactly the same as with the row-variation.

△

The procedure seems to work, but as mathematicians we would like to see some justification of the steps above.

When we create our guess  $\vec{y}(x) = Y(x)\vec{c}(x)$ , we are supposed to substitute it into the given system:

$$[Y(x)\vec{c}(x)]' = AY(x)\vec{c}(x) + \vec{b}(x).$$

What next? We would like to differentiate on the left. Surprisingly enough, the product rule that we know for functions also works for matrices and vectors of functions. Here is a proof for a general matrix  $G(x)$  and vector function  $\vec{f}(x)$ . First,

$$G(x)\vec{f}(x) = \begin{pmatrix} g_{1,1}(x) & \cdots & g_{1,n}(x) \\ \vdots & & \vdots \\ g_{n,1}(x) & \cdots & g_{n,n}(x) \end{pmatrix} \begin{pmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{pmatrix} = \begin{pmatrix} \sum_j g_{1,j}(x)f_j(x) \\ \vdots \\ \sum_j g_{n,j}(x)f_j(x) \end{pmatrix}.$$

Therefore

$$\begin{aligned}
 [G(x)\vec{f}(x)]' &= \left[ \begin{pmatrix} \sum_j g_{1,j}(x)f_j(x) \\ \vdots \\ \sum_j g_{n,j}(x)f_j(x) \end{pmatrix} \right]' = \begin{pmatrix} \sum_j g'_{1,j}(x)f_j(x) + \sum_j g_{1,j}(x)f'_j(x) \\ \vdots \\ \sum_j g'_{n,j}(x)f_j(x) + \sum_j g_{n,j}(x)f'_j(x) \end{pmatrix} \\
 &= \begin{pmatrix} \sum_j g'_{1,j}(x)f_j(x) \\ \vdots \\ \sum_j g'_{n,j}(x)f_j(x) \end{pmatrix} + \begin{pmatrix} \sum_j g_{1,j}(x)f'_j(x) \\ \vdots \\ \sum_j g_{n,j}(x)f'_j(x) \end{pmatrix} \\
 &= \begin{pmatrix} g'_{1,1}(x) & \cdots & g'_{1,n}(x) \\ \vdots & & \vdots \\ g'_{n,1}(x) & \cdots & g'_{n,n}(x) \end{pmatrix} \begin{pmatrix} f_1(x) \\ \vdots \\ f_n(x) \end{pmatrix} + \begin{pmatrix} g_{1,1}(x) & \cdots & g_{1,n}(x) \\ \vdots & & \vdots \\ g_{n,1}(x) & \cdots & g_{n,n}(x) \end{pmatrix} \begin{pmatrix} f'_1(x) \\ \vdots \\ f'_n(x) \end{pmatrix} \\
 &= G'(x)\vec{f}(x) + G(x)\vec{f}'(x).
 \end{aligned}$$

Applying this to our equations we obtain

$$Y'(x)\vec{c}(x) + Y(x)\vec{c}'(x) = AY(x)\vec{c}(x) + \vec{b}(x).$$

Now we expect some cancelling to happen. We would like to argue that  $Y'(x)\vec{c}(x)$  and  $AY(x)\vec{c}(x)$  are the same expression. We could confirm it by rewriting these expressions into rows, in effect going back to the row variation. We prefer to give a matrix-based argument.

The justification starts by recalling a general trick from linear algebra. Given an  $n \times n$  matrix  $G = (g_{ij})$  and a vector  $\vec{h}$  of length  $n$ , we can write the product  $G\vec{h}$  as

$$G\vec{h} = \sum_i h_i \overrightarrow{(g_{i,j})_j},$$

where  $\overrightarrow{(g_{i,j})_j}$  represents the  $i$ th column of the matrix  $G$ . Applying it to our situation, we first recall that the  $i$ th column of  $Y(x)$  is actually  $\vec{y}_i$ , the  $i$ th vector of our fundamental system and, in particular, a solution of the associated homogeneous system. Thus we deduce that

$$Y'(x)\vec{c}(x) = \sum_i c_i(x)\vec{y}_i' = \sum_i c_i(x)A\vec{y}_i.$$

The other expression  $AY(x)\vec{c}(x)$  represents a linear combination of columns of  $AY(x)$ , but every column of this matrix is in fact  $A\vec{y}_i$ , so

$$AY(x)\vec{c}(x) = \sum_i c_i(x)A\vec{y}_i.$$

This confirms that indeed,  $Y'(x)\vec{c}(x) = AY(x)\vec{c}(x)$  and the cancelling does happen. We obtain the equation

$$Y(x)\vec{c}'(x) = \vec{b}(x)$$

as stated by the algorithm.

The next questionable step is the application of  $Y(x)^{-1}$ . How do we know that this inverse matrix actually exists? For the answer we return to chapter 26, where we recognized linear independence of the proposed basis by checking whether the matrix formed by these vectors is regular. In other words, the matrix  $Y(x)$  is invertible.

The last unjustified step is the integration phase. There we can offer two answers. In all the theorems in this chapter we assume that we work on an interval  $I$  where all functions involved are continuous, and then also the expressions that we integrate are continuous and hence integrable. That is the theoretical point of view. The practical point of view is that in this step we actually rely on luck, because we are not always able to actually find antiderivatives for continuous functions.

This is the one weak point of variation, and it is not different from variation for one equation of order one as we saw it before. We simply have to get lucky once in a while.

For another example of variation (in its row version) see 27b.c below.

## 27b. Method of undetermined coefficients

The method of undetermined coefficients proved to be fairly efficient for solving high order linear differential equations. Could it be applied also to systems?

The answer is in the positive, but with some modifications that make it less pleasant. There are two problems to consider.

First, imagine that there is an expression of suitable type, say,  $13e^{5x}$ , on the right in one of the equations. We know that we should feed our equation with  $Ae^{5x}$  (if we ignore possible corrections for a moment), but since all unknown functions  $y_1, \dots, y_n$  appear (at least potentially) in every equation, we do not know which of the functions  $y_i$  should receive the term  $Ae^{5x}$  when we form our guesses. Consequently, we have to include such a term in all function, and each copy with its own constant, because it can easily happen that several functions join forces to create the desired output.

In short, whenever we see some expression on the right in one of the equations, all candidates  $y_i$  have to include the corresponding general term. But then it can easily happen that the same type appears in several equations, which would make us add another term of this type into our guesses. However, we do not want unnecessary constants, so we merge terms of the same type and ultimately, the one with the highest degree of polynomial swallows all the others. In effect, when putting a certain type into guesses for  $y_i$ , we only consider the one with highest degree of polynomial.

Second, imagine that we have a term of a certain type on the right of some equation, and its special number  $\lambda$  also happens to be an eigenvalue of the system. Then we have to correct, and if this eigenvalue is of higher multiplicity, then we use stronger correction. However, due to mutual interaction of equations, it may turn out that the original guess actually survives passage through the system. In other words, we never know whether the correction is actually needed. This means that we should add to our guess (in fact to all guesses for  $y_i$  per the previous observation) not just the term with the proper correction, but also corresponding terms with smaller corrections all the way to a term without any correction at all.

In effect, this creates a full polynomial of a raised degree. To see this, imagine that the basic form of guess is  $(Ax + B)e^{ax}$ , and that  $a$  happens to be a double eigenvalue, that is  $m = 2$ . Then we should add into our guesses the terms

$$x^2(Ax + B)e^{ax} + x(Cx + D)e^{ax} + (Ex + F)e^{ax} = (Ax^3 + (B + C)x^2 + (D + E)x + F)e^{ax}.$$

Since we do not want unnecessary constants, we simply treat this as a full polynomial of degree  $1 + 2$ .

All this can make the guessing method very labor-intensive.

### Algorithm 27b.1.

⟨method of undetermined coefficients for systems of linear ODEs⟩

Given: a system  $\vec{y}' = A\vec{y} + \vec{b}(x)$ , with  $A$  a real-valued  $n \times n$  matrix.

Assumptions: Right-hand sides are linear combinations of quasipolynomials, that is, of expressions of the type  $p(x)e^{\alpha x} \cos(\beta x) + q(x)e^{\alpha x} \sin(\beta x)$

1. Find a general solution  $y_{1h}, \dots, y_{nh}$  of the associated homogeneous system  $\vec{y}' = A\vec{y}$ , in particular determine the eigenvalues  $\lambda$  of the matrix  $A$ .

2. Scan the right-hand sides for all quasipolynomials and make a list.

Collate all terms of the same type as determined by the special number  $\lambda = \alpha + \beta i$ , and in each group, leave only the term with the highest power of polynomial. If there is a tie, pick just any.

3. Create candidates  $y_1, \dots, y_n$  for a particular solution as follows:

- a) For each quasipolynomial on the list, guess the basic form of a solution. Assume that the degree of the polynomial in it is  $d$ .
- b) For the special number of this term, determine its multiplicity  $m$  as an eigenvalue, including the case  $m = 0$  if it is not an eigenvalue at all. Modify the basic form to feature a full polynomial of degree  $d + m$ .
- c) Add the modified form of the guess to all candidates for  $y_i$ .
4. Substitute the created guesses  $y_1, \dots, y_n$  into the given system and by comparing terms, set up a system of linear equations that determines unknown constants. Solve this system, substitute the known constants into the guessed forms  $y_1, \dots, y_n$  and obtain particular solutions  $y_{1p}, \dots, y_{np}$ .
5. The general solution is  $y_i = y_{ip} + y_{ih}$  for  $i = 1, \dots, n$ .
- △

**Example 27b.a:** Consider the system

$$\begin{aligned}y_1' &= \dots + \cos(2x) + 2x + 1 - 13e^{2x} \\y_2' &= \dots + x^2 + x e^{2x} + 23 \sin(x) \\y_3' &= \dots + e^x + 14 \sin(2x) - 13.\end{aligned}$$

We actually do not care much about the actual matrix of the system here, but we need to know that its eigenvalues are  $\lambda = 0, 2, 2$ .

What types do we see among the right-hand sides?

- There are two terms of the trigonometric type with frequency  $\beta = 2$  (special number  $\lambda = 2i$ ), namely  $\cos(2x)$  and  $14 \sin(2x)$ . Both have a polynomial of degree zero attached, so the basic form of the guess should be  $A \cos(2x) + B \sin(2x)$ . Since the special number does not match any eigenvalue, we do not have to increase the degree of the polynomial.

- There are three polynomial terms (special number  $\lambda = 0$ ), namely  $2x + 1$ ,  $x^2$  and  $-13$ . The highest degree is two, so the basic form of guess would be  $Ax^2 + Bx + C$ . However, the special number matches one of the eigenvalues (multiplicity 1), so we have to increase the degree by one and include a full cubic polynomial in all guesses.

- There are two terms of exponential type with factor  $\alpha = 2$  (special number  $\lambda = 2$ ), namely  $-13e^{2x}$  and  $x e^{2x}$ . The higher degree wins, so the basic form of our guess is  $(Ax + B)e^{2x}$ . However, the special number matches a double eigenvalue, so we have to increase the degree of the polynomial by two, and a cubic polynomial should be attached to this exponential.

- There is the term  $13 \sin(x)$  of the trigonometric type with frequency  $\beta = 1$  (special number  $\lambda = i$ ), so the basic form of the guess should be  $A \cos(x) + B \sin(x)$ . Since the special number does not match any eigenvalue, we do not have to increase the degree of the polynomial.

- Finally, there is the term  $e^x$  of exponential type with factor  $\alpha = 1$  (special number  $\lambda = 1$ ), so the basic form of our guess is  $A e^x$ . Since the special number does not match any eigenvalue, we do not have to increase the degree of the polynomial.

We therefore seek a solution of our system of the form

$$\begin{aligned}y_1 &= A \cos(2x) + B \sin(2x) + Cx^3 + Dx^2 + Ex + F + (Gx^3 + Hx^2 + Ix + J)e^{2x} \\&\quad + K \cos(x) + L \sin(x) + M e^x, \\y_2 &= O \cos(2x) + P \sin(2x) + Qx^3 + Rx^2 + Sx + T + (Ux^3 + Vx^2 + Wx + X)e^{2x} \\&\quad + Y \cos(x) + Z \sin(x) + a e^x, \\y_3 &= b \cos(2x) + c \sin(2x) + dx^3 + fx^2 + gx + h + (jx^3 + kx^2 + lx + m)e^{2x} \\&\quad + n \cos(x) + o \sin(x) + p e^x.\end{aligned}$$

Substituting these into the given system, we would hope to obtain 39 equations for 39 unknown coefficients. This explains why we did not really care about the actual system, we are definitely

not going through this.

△

**Example 27b.b:** For the last time we return to the example 26.a,

$$\begin{aligned}y_1' &= 2y_1 + y_2 - 3 \\y_2' &= y_1 + 2y_2 + 3x - 4.\end{aligned}$$

Its homogeneous solution is

$$\begin{aligned}y_{1h}(x) &= a e^x + b e^{3x}, \\y_{2h}(x) &= -a e^x + b e^{3x}, \quad x \in \mathbb{R}.\end{aligned}$$

Now we find candidates for a particular solution. Surveying the right-hand sides we see that they feature just one type, namely a polynomial (special number  $\lambda = 0$ ), in two incarnations, one with degree zero and the other with degree one. According to our observations above, the higher degree wins. The special number does not match any eigenvalue, we therefore do not need to increase the degree of polynomial, and thus we should include polynomials of order 1 in our guesses:

$$\begin{aligned}y_1(x) &= Ax + B, \\y_2(x) &= Cx + D.\end{aligned}$$

To find the constants, we substitute our guesses into the given system:

$$\begin{aligned}[Ax + B]' &= 2(Ax + B) + (Cx + D) - 3 && \implies (-2A - C)x + (A - 2B - D) = -3 \\[Cx + D]' &= (Ax + B) + 2(Cx + D) + 3x - 4 && \implies (-A - 2C)x + (-B + C - 2D) = 3x - 4.\end{aligned}$$

Comparing both sides we see four equations that we easily solve:

$$\begin{aligned}-2A - C &= 0 & A &= 1 & A &= 1 \\A - 2B - D &= -3 & C &= -2 & C &= -2 \\-A - 2C &= 3 & \implies -2B - D &= -4 & \implies B &= 2 \\-B + C - 2D &= -4 & -B - 2D &= -2 & D &= 0.\end{aligned}$$

We found the following particular solution.

$$\begin{aligned}y_{1p}(x) &= x + 2, \\y_{2p}(x) &= -2x.\end{aligned}$$

It agrees with the one obtained by elimination and variation, so we are quite confident that it is correct.

△

**Example 27b.c:** Consider the following initial value problem:

$$\begin{aligned}y_1' &= y_1 + y_2 + 1 & y_1(0) &= 2 \\y_2' &= y_1 + y_2 + 1 - e^x, & y_2(0) &= 0.\end{aligned}$$

1. First we find its general solution.

a) Homogeneous version: the matrix is  $A = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ . In the usual way we find  $\lambda = 0, 2$ . We determine associated eigenvectors and obtain the homogeneous solution

$$\begin{aligned}y_{1h} &= a + b e^{2x}, \\y_{2h} &= -a + b e^{2x}.\end{aligned}$$

b) Now we find a particular solution.

We start with the variation method (row version), and look for a solution of the form

$$\begin{aligned}y_{1p} &= a(x) + b(x)e^{2x}, \\y_{2p} &= -a(x) + b(x)e^{2x}.\end{aligned}$$

Substituting our candidates into the given system we obtain equations

$$\begin{aligned} a'(x) + b'(x)e^{2x} &= 1 \\ -a'(x) + b'(x)e^{2x} &= 1 - e^x. \end{aligned}$$

Normally we would solve this by adding and subtracting these equations, but to show an alternative, we will use the Cramer rule:

$$\begin{aligned} D &= \det \begin{pmatrix} 1 & e^{2x} \\ -1 & e^{2x} \end{pmatrix} = e^{2x} + e^{2x} = 2e^{2x}, \\ D_a &= \det \begin{pmatrix} 1 & e^{2x} \\ 1 - e^x & e^{2x} \end{pmatrix} = e^{2x} - e^{2x}(1 - e^x) = e^{3x}, \\ D_b &= \det \begin{pmatrix} 1 & 1 \\ -1 & 1 - e^x \end{pmatrix} = 1 - e^x + 1 = 2 - e^x. \end{aligned}$$

Note that  $D$  is in fact the determinant of the fundamental matrix  $Y(x)$ , so it will never be zero. We can therefore find

$$\begin{aligned} a'(x) &= \frac{D_a}{D} = \frac{1}{2}e^x \\ \implies a(x) &= \frac{1}{2}e^x, \\ b'(x) &= \frac{D_b}{D} = e^{-2x} - \frac{1}{2}e^{-x} \\ \implies b(x) &= -\frac{1}{2}e^{-2x} + \frac{1}{2}e^{-x}. \end{aligned}$$

We can substitute into our candidates and obtain

$$\begin{aligned} y_{1p} &= \left(\frac{1}{2}e^x\right) + \left(-\frac{1}{2}e^{-2x} + \frac{1}{2}e^{-x}\right)e^{2x} = e^x - \frac{1}{2}, \\ y_{2p} &= -\left(\frac{1}{2}e^x\right) + \left(-\frac{1}{2}e^{-2x} + \frac{1}{2}e^{-x}\right)e^{2x} = -\frac{1}{2}. \end{aligned}$$

Using the formula  $\vec{y}_p + \vec{y}_h$  we obtain a general solution

$$\begin{aligned} y_1 &= e^x - \frac{1}{2} + a + b e^{2x}, \\ y_2 &= -\frac{1}{2} - a + b e^{2x}, \quad x \in \mathbb{R}. \end{aligned}$$

Next, we will try the guessing method. Scanning the right-hand sides we see a polynomial term appearing twice, each time as just the constant 1, so the basic form of our guess is just a constant polynomial  $A$ . However, the corresponding special number  $\lambda = 0$  matches one of the eigenvalues, so we have to make a correction by one and use a general polynomial of order one in our guess.

We also see an exponential term  $-e^x$ , leading to the basic guess  $A e^x$ . The corresponding special number  $\lambda = 1$  does not match eigenvalues, so there will be no correction.

We therefore form the following guess for solutions:

$$\begin{aligned} y_{1p} &= A e^x + Bx + C, \\ y_{2p} &= D e^x + Ex + F. \end{aligned}$$

We substitute these into the given system:

$$\begin{aligned} [A e^x + Bx + C]' &= (A e^x + Bx + C) + (D e^x + Ex + F) + 1 \\ [D e^x + Ex + F]' &= (A e^x + Bx + C) + (D e^x + Ex + F) + 1 - e^x \\ \implies &(-B - E)x + (B - C - F) - D e^x = 1 \\ &(-B - E)x + (-C + E - F) - A e^x = 1 - e^x. \end{aligned}$$

Comparing both sides we obtain the following system:

$$\begin{aligned} -B - E &= 0 & -B - E &= 0 \\ B - C - F &= 1 & -C + E - F &= 1 \\ -D &= 0 & -A &= -1. \end{aligned}$$

We immediately see that one equation is twice there, which means that this system is undetermined and we are free to choose one variable as we wish. This sometimes happens when solving systems by guessing, in fact it can be expected when we have an eigenvalue match and a resulting correction happening.

However, it is not clear which unknown can be freely chosen, and if we pick the wrong one, we may create an unsolvable system that way. We have to simplify the system before we see how things stand.

We immediately obtain  $D = 0$  and  $A = 1$ , and thus our system reduces to

$$\begin{aligned} B + E &= 0 \\ B - C - F &= 1 \\ -C + E - F &= 1 \end{aligned} \implies \left( \begin{array}{cccc|c} 1 & 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & -1 & 1 \\ 0 & -1 & 1 & -1 & 1 \end{array} \right)$$

We use elimination to reduce the matrix:

$$\left( \begin{array}{cccc|c} 1 & 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & -1 & 1 \\ 0 & -1 & 1 & -1 & 1 \end{array} \right) \sim \left( \begin{array}{cccc|c} 1 & 0 & 1 & 0 & 0 \\ 0 & -1 & -1 & -1 & 1 \\ 0 & -1 & 1 & -1 & 1 \end{array} \right) \sim \left( \begin{array}{cccc|c} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & -1 \\ 0 & 0 & 2 & 0 & 0 \end{array} \right)$$

The third row shows that  $E = 0$  is given, hence also  $B = 0$  by the first equation. Surprisingly, the corrected terms  $Bx$  and  $Ex$  are not needed, although the special number  $\lambda = 0$  did match an eigenvalue. This sometimes happens with systems, but you never know until you try it, so we always have to include general corrected terms when a match occurs.

The shape of the matrix suggests that we can choose  $F$ . We opt for the traditional  $F = 0$ , then  $C = -1$  and we get the particular solution

$$\begin{aligned} y_{1p} &= e^x - 1, \\ y_{2p} &= 0. \end{aligned}$$

We see that  $y_{2p}$  has nothing which is a bit unfair, perhaps we should have chosen something else for  $F$ , but it is too late now, life is tough.

What is more interesting, we obtained a different particular solution compared to our previous attempt using variation. This can happen, there are infinitely many possible solutions after all. This time we get a general solution

$$\begin{aligned} y_1 &= e^x - 1 + a + b e^{2x}, \\ y_2 &= -a + b e^{2x}, \quad x \in \mathbb{R}. \end{aligned}$$

This is actually incorrect. The way we wrote it, it would seem that the homogeneous parts of this and the previous attempt match while particular solutions differ, so they would describe different sets of functions. In fact, the parameters  $a, b$  here are different from those in the previous solution, so properly we should write

$$\begin{aligned} y_1 &= e^x - 1 + \tilde{a} + \tilde{b} e^{2x}, \\ y_2 &= -\tilde{a} + \tilde{b} e^{2x}, \quad x \in \mathbb{R}. \end{aligned}$$

Does this general formula describe the same set of solutions as the previous one? In other words, when we create a specific pair of functions using one description, can we also obtain the same pair from the other description (perhaps using different values of parameters)?

We start by observing that if we use  $a = -\frac{1}{2}$  and  $b = 0$  in the first general solution, we obtain  $y_1 = e^x - 1$  and  $y_2 = 0$ , which is exactly the particular solution we have now. Conversely, the particular solution obtained by variation can be obtained by substituting  $\tilde{a} = \frac{1}{2}$  and  $\tilde{b} = 0$  into the new solution.

This is an inspiration for a general transformation: We claim that if the first formula generates some solution using specific values  $a, b$ , then the new solution creates the same functions when using parameters  $\tilde{a} = a - \frac{1}{2}$ ,  $\tilde{b} = b$ . Indeed, when we substitute these values into the new formula, we obtain exactly the previous version of general solutions. In this way we confirm that these two

formulas describe the same set of solutions.

2. To handle the initial conditions we use the second version of a general solution, because it is fraction-free. We will omit those tildas for simplicity.

$$\begin{aligned} y_1(0) = 2 &\implies e^0 - 1 + a + be^0 = 2 &\implies a + b = 2 \\ y_2(0) = 0 &\implies -a + be^0 = 0 &\implies -a + b = 0. \end{aligned}$$

We conclude that  $a = b = 1$  and the answer to the given question is

$$\begin{aligned} y_1(x) &= e^{2x} + e^x, \\ y_2(x) &= e^{2x} - 1, \quad x \in \mathbb{R}. \end{aligned}$$

If we used the first form of a general solution, we would have to use parameters  $a = \frac{1}{2}$ ,  $b = 1$  to obtain this solution.

△

### 27c. Bonus: Differential elimination

In chapter 20 we talked of differentiation as if it were a mapping. We will push this idea further here.

We start by denoting the differential operator  $f \mapsto f'$  with the letter  $D$ . So  $D[y] = y'$ . It is actually customary to skip the brackets, people would often write just  $Dy$ . We can rewrite differential equations using this notation, for instance

$$y'' - 3y' + 2y = 0 \implies D[D[y]] - 3D[y] + 2y = 0.$$

This is where it gets interesting. The second derivative means that we apply derivative to an outcome of the previous derivative,  $y'' = [y']'$ . In the mapping language, we use the output of the mapping  $D$  as an argument of another edition of  $D$ ,  $D[D[y]]$ . This is a familiar concept, namely the composition of mappings. Normally we would write  $D \circ D$ , but when composing a mapping with itself, we call it a power of this mapping and denote it using the square notation. Therefore  $y''$  can be captured as  $D^2[y]$  or simply  $D^2y$  for short.

Similarly,  $D^3y = y'''$  and so on. We can therefore write

$$y'' - 3y' + 2y = 0 \implies D^2y - 3Dy + 2y = 0.$$

Time for another review When we see  $2\ln(x) + \sin(x)$ , we can interpret it in two ways. We can see two outcomes of functions, namely  $\ln(x)$  and  $\sin(x)$ , combined into a linear combination. Or, we can see it as the function  $2\ln + \sin$  applied to  $x$ . Indeed, when we define the addition and multiplication of functions as objects, we do it by the formula

$$(f + g)(x) = f(x) + g(x), \quad (f \cdot g)(x) = f(x) \cdot g(x).$$

Therefore we can see  $D^2y - 3Dy + 2y$  as the outcome of the mapping  $D^2 - 3D + 2$  being applied to a function  $y$ .

$$y'' - 3y' + 2y = 0 \implies (D^2 - 3D + 2)[y] = 0.$$

Now it gets interesting. If  $D$  was just a number we could write  $D^2 - 3D + 2 = (D - 2)(D - 1)$ . The interesting thing is that this factorization is also valid for the differential operator  $D$  if we interpret the multiplication of mappings in this formula as a composition. In other words, we will understand it as follows:

$$(D - 2)(D - 1)[y] = (D - 2)[(D - 1)[y]].$$

Does it work? Let's see:

$$\begin{aligned} (D - 2)[(D - 1)[y]] &= (D - 2)[Dy - y] = D[y' - y] - 2(y' - y) = [y' - y]' - 2y' + 2y \\ &= y'' - y' - 2y' + 2y = y'' - 3y' + 2y = (D^2 - 3D + 2)[y]. \end{aligned}$$

Moreover, we even have a commutative law in place. For instance, we easily check that

$$(D - 2)(D - 1)[y] = (D - 1)(D - 2)[y].$$

We can express the key property as follows:

- If we interpret compositions of  $D$  as multiplications of  $D$ , then linear combinations of the operator  $D$  behave just like polynomials.

This can be used to treat systems of linear differential equations in a completely novel way. For the start, every such system can be rewritten so that the unknown functions appear on the left.

**Example 27c.a:** We return to our favorite example 26.a:

$$\begin{aligned}y_1' &= 2y_1 + y_2 - 3 \\y_2' &= y_1 + 2y_2 + 3x - 4.\end{aligned}$$

We rewrite it as

$$\begin{aligned}y_1' - 2y_1 - y_2 &= -3 \\y_2' - y_1 - 2y_2 &= 3x - 4.\end{aligned}$$

Using our new notation we can write it also as

$$\begin{aligned}(D - 2)y_1 - y_2 &= -3 \\-y_1 + (D - 2)y_2 &= 3x - 4.\end{aligned}$$

Now for the punchline:

$$\begin{pmatrix} (D - 2) & -1 \\ -1 & D - 2 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} -3 \\ 3x - 4 \end{pmatrix}.$$

△

At this point we may really look at all linear equations, not just those where only one unknown function is allowed to be differentiated. For instance, the system

$$\begin{aligned}y_1'' + y_2' - 2y_1 + 3y_2 &= e^x \\y_1' + y_2' + 4y_1 - y_2 &= 13x\end{aligned}$$

can be written as

$$\begin{pmatrix} (D^2 - 2) & D + 3 \\ D + 4 & D - 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} e^x \\ 13 \end{pmatrix}.$$

The beauty of this notation is that we can now work with this using normal tools from linear algebra, treating individual entries that feature  $D$  as if they were polynomials in  $D$ . For instance, systems can be solved using elimination. However, certain special strategies should be followed, and they correspond to strategies related to polynomial matrices.

When working with a certain column, we compare the candidates for the pivot. Treating the entries as polynomials, we choose for the pivot the one with the lowest degree. Using row operations, it should be possible to make sure that in the new matrix, the entries in the pivot column feature only lower powers of  $D$  than the degree of the pivot. When removing higher powers of  $D$  in a certain entry, we usually start from the highest and work our way down, as far as possible. Sometimes, with smart multiplication of the pivot row, we can remove several powers in one step.

If all the entries below the pivot become zero, then this stage is done, and we also reduce the terms above the pivot as much as possible. Otherwise, we choose for our new pivot the row with the lowest degree in the pivot column and repeat the above process. In this way it should be possible to achieve zeros below the pivot, and have the pivot with the smallest possible degree, which is very desirable.

To illustrate the main points of this type of elimination we will focus on just the pivot column now. This means that we are addressing a certain unknown function in our system. So consider a certain system where  $y_1$  appears as follows:

$$\begin{aligned}y_1'' - y_1 \\y_1'' + 2y_1' + y_1 \\y_1''' + y_1''\end{aligned} \implies \begin{pmatrix} D^2 - 1 \\ D^2 + 2D + 1 \\ D^3 + D^2 \end{pmatrix}.$$

For the pivot we should choose the term with lowest degree, and we have two there. Then it is better to use the simpler one, so in fact the term in the first row is the best candidate for our pivot already. Now we should be able to remove  $D^2$  and also all higher powers from the remaining terms in the column. We do so easily in the second row simply by subtracting the first row from it.

With the third row we first focus on  $D^3$ . We remove it when we subtract the first row “multiplied” by  $D$ . In effect this means that we add another order of derivative to the operator represented by the term  $D^2 - 1$ . Obviously, if we were solving a system, all these operations would be acting in parallel on all other columns. We obtain

$$\begin{pmatrix} D^2 - 1 \\ D^2 + 2D + 1 \\ D^3 + D^2 \end{pmatrix} \sim \begin{pmatrix} D^2 - 1 \\ 2D + 2 \\ D^2 + D \end{pmatrix}.$$

We got rid of  $D^3$  in the third row, and now we focus on  $D^2$  there. This is easy to handle, we just subtract the first row.

$$\begin{pmatrix} D^2 - 1 \\ 2D + 2 \\ D^2 + D \end{pmatrix} \sim \begin{pmatrix} D^2 - 1 \\ 2D + 2 \\ D + 1 \end{pmatrix}.$$

This is as far as our pivot can get us, and we start the whole process again. First we choose a new pivot, the term with the lowest degree, and again there are two candidates. We will choose the one from the third row and then use it to reduce the middle row by subtracting the new pivot row twice.

$$\begin{pmatrix} D^2 - 1 \\ 2D + 2 \\ D^2 + D \end{pmatrix} \sim \begin{pmatrix} D + 1 \\ 2D + 2 \\ D^2 - 1 \end{pmatrix} \sim \begin{pmatrix} D + 1 \\ 0 \\ D^2 - 1 \end{pmatrix}.$$

Now we address the third row. Normally we would first reduce the order by subtracting the first row multiplied by  $D$ , and then handle the first power by a suitable subtraction again. However, we may notice that  $D^2 - 1$  is in fact a multiple (in the world of polynomials in  $D$ ) of the pivot, so we can use the usual approach from elimination. That is, we multiply the pivot row by  $D - 1$  and subtract it from the third row. We obtain

$$\begin{pmatrix} D + 1 \\ 0 \\ D^2 - 1 \end{pmatrix} \sim \begin{pmatrix} D + 1 \\ 0 \\ 0 \end{pmatrix}.$$

Obviously, all these operations would be acting in parallel on all other columns in an actual matrix of a system. In the end we obtain a new system where  $y_1$  appear only in the first equation, namely in the form  $y_1' + y_1$ . We can do nothing more with it, and it is enough.

**Example 27c.b:** We return to our favorite example.

$$\begin{aligned} y_1' = 2y_1 + y_2 - 3 \\ y_2' = y_1 + 2y_2 + 3x - 4 \end{aligned} \implies \begin{aligned} (D - 2)y_1 - y_2 = -3 \\ -y_1 + (D - 2)y_2 = 3x - 4 \end{aligned} \implies \left( \begin{array}{cc|c} D - 2 & -1 & -3 \\ -1 & D - 2 & 3x - 4 \end{array} \right).$$

In the first column, the second entry has a lower degree, so we take it for the pivot.

$$\left( \begin{array}{cc|c} D - 2 & -1 & -3 \\ -1 & D - 2 & 3x - 4 \end{array} \right) \sim \left( \begin{array}{cc|c} -1 & D - 2 & 3x - 4 \\ D - 2 & -1 & -3 \end{array} \right).$$

We multiply the first row by  $D - 2$ , so it becomes  $(-D + 2 \mid (D - 2)^2 \mid (D - 2)[3x - 4])$ . The term in the third column is actually

$$(D - 2)[3x - 4] = [3x - 4]' - 2(3x - 4) = 3 - 6x + 8 = 11 - 6x.$$

We add this to the second row and obtain

$$\left( \begin{array}{cc|c} -1 & D - 2 & 3x - 4 \\ D - 2 & -1 & -3 \end{array} \right) \sim \left( \begin{array}{cc|c} -1 & D - 2 & 3x - 4 \\ 0 & (D - 2)^2 - 1 & 8 - 6x \end{array} \right) \sim \left( \begin{array}{cc|c} -1 & D - 2 & 3x - 4 \\ 0 & D^2 - 4D + 3 & 8 - 6x \end{array} \right).$$

This concludes the first stage, reduction of the first column, and we move to the second column. Since the pivot there does not have a lower degree than the term above it, we cannot do any simplification there, so the elimination stops. We obtained the system

$$\begin{aligned} -y_1 + (D-2)[y_2] &= 3x-4 & -y_1 + y_2' - 2y_2 &= 3x-4 \\ (D^2-4D+3)[y_2] &= -6x+8 & \implies (D^2-4D+3)y_2'' &= 4y_2' + 3y_2 = -6x+8. \end{aligned}$$

This is already solvable. The second equation is just a single linear equation of order two that we can handle. The first equation then represents a formula that allows us to derive  $y_1$  from  $y_2$ .

Solving the second equation we obtain,  $y_2(x) = -2x + a e^x + b e^{3x}$ . We substitute it into the first and get  $y_1(x) = x + 2 - a e^x + b e^{3x}$ . These are not exactly the same formulas as those that we obtained in example 26.a, but they describe the same set of solutions. Indeed, after substituting  $-a$  for  $a$  here the formulas match.

△

Remarkably, we can also use the Cramer rule. For systems of linear algebraic equations we use it in the form

$$x_i = \frac{\det(A_i)}{\det(A)}.$$

For systems of linear differential equations, the form  $\det(A)y_i = \det(A_i)$  is preferable, as we do not quite know how to divide with differential operators. However, there is an unexpected twist there.

**Example 27c.c:** We return to our favorite system

$$\left( \begin{array}{cc|c} D-2 & -1 & -3 \\ -1 & D-2 & 3x-4 \end{array} \right).$$

We apply the formulas, first for  $y_1$ :

$$\begin{aligned} \det \begin{pmatrix} D-2 & -1 \\ -1 & D-2 \end{pmatrix} y_1 &= \det \begin{pmatrix} -3 & -1 \\ 3x-4 & D-2 \end{pmatrix} \\ \implies [(D-2)^2 - 1]y_1 &= (D-2)[-3] - (-1)(3x-4) \\ \implies [D^2 - 4D + 3]y_1 &= [-3]' - 2 \cdot (-3) + (3x-4). \end{aligned}$$

We obtain the equation  $y_1'' - 4y_1' + 3y_1 = 3x + 2$ . By a remarkable coincidence, we arrived at this equation when solving this system by elimination in example 26.a, and we obtained the solution  $y_1(x) = x + 2 + a e^x + b e^{3x}$ .

Now we address the second unknown function.

$$\begin{aligned} \det \begin{pmatrix} D-2 & -1 \\ -1 & D-2 \end{pmatrix} y_2 &= \det \begin{pmatrix} D-2 & -3 \\ -1 & 3x-4 \end{pmatrix} \\ \implies [(D-2)^2 - 1]y_2 &= (D-2)[3x-4] - (-1)(-3). \end{aligned}$$

We obtain the equation  $y_2'' - 4y_2' + 3y_2 = -6x + 8$ . We solved it above, and we know that  $y_2(x) = -2x + c e^x + d e^{3x}$ .

Which brings us to the unexpected twist. We found the functions  $y_1, y_2$  independently, and thus there is no reason to assume any connection between the parameters  $a, b$  in  $y_1$  and  $c, d$  in  $y_2$ . However, there is a relationship (after all, the space of solutions is not four-dimensional), and we find this by substituting our solutions into the original system. We obtain

$$\begin{aligned} [x + 2 + a e^x + b e^{3x}]' &= 2[x + 2 + a e^x + b e^{3x}] + [-2x + c e^x + d e^{3x}] - 3 \\ [-2x + c e^x + d e^{3x}]' &= [x + 2 + a e^x + b e^{3x}] + 2[-2x + c e^x + d e^{3x}] + 3x - 4, \end{aligned}$$

that is,

$$\begin{aligned} a e^x + 3b e^{3x} &= 2a e^x + 2b e^{3x} + c e^x + d e^{3x} & \implies (a+c)e^x + (-b+d)e^{3x} &= 0 \\ c e^x + 3d e^{3x} &= a e^x + b e^{3x} + 2c e^x + 2d e^{3x} & \implies (a+c)e^x + (b-d)e^{3x} &= 0. \end{aligned}$$

Since the functions  $e^x$ ,  $e^{3x}$  are linearly independent, the first equality can be true only if  $a + c = 0$  and  $-b + d = 0$ . Similarly, the second equality demands that  $a + c = 0$  and  $b - d = 0$ , but these are actually the same equations as from the first equality. We conclude that  $d = b$  and  $c = -a$ . We substitute these into the formula for  $y_2$  and obtain the solution

$$\begin{aligned}y_1(x) &= x + 2 + a e^x + b e^{3x}, \\y_2(x) &= -2x - a e^x + b e^{3x}, \quad x \in \mathbb{R}.\end{aligned}$$

This agrees with our previous results.

△

We see the using the Cramer rule, solving an  $n \times n$  system would require solving  $n$  linear differential equations of higher order. This is very likely more work than elimination, where with a bit of luck it is enough to solve just one such equation and the remaining functions can be obtained by substituting into direct formulas. However, this is not always the case, and sometimes we also have to solve more differential equations when using elimination.

The operator approach for elimination is a tool that can be also applied to the problem of reducing a system to just one equation, see chapter 26.

**Example 27c.d:** We return to example 26.b. First we rewrite it properly.

$$\begin{aligned}Dy_1 &= y_1 + 2y_2 + y_3 + 1 & (D-1)y_1 - 2y_2 - y_3 &= 1 \\Dy_2 &= -y_1 + 2y_2 + 2y_3 & \implies y_1 + (D-2)y_2 - 2y_3 &= 0 \\Dy_3 &= 2y_1 + y_2 + y_3 + x & -2y_1 - y_2 + (D-1)y_3 &= x \\& & y_1 + (D-2)y_2 - 2y_3 &= 0 \\& \implies (D-1)y_1 - 2y_2 - y_3 &= 1 \\& & -2y_1 - y_2 + (D-1)y_3 &= x\end{aligned}$$

If we get lucky, an unknown appears in one of the equations without the differential operator  $D$ , then we can use that equation to eliminate this unknown from the other equations. In this particular case we decided to use the second equation with isolated  $y_1$ , so we moved it to the top.

Informally, we could express  $y_1$  from the first equation and substitute into others, here we will show a formal elimination approach. We add double of the first row to the third, and apply  $D-1$  to the first row before we subtract it from the second. The first row then looks like this:

$$(D-1)y_1 + (D-1)(D-2)y_2 - 2(D-1)y_3 = 0.$$

We obtain

$$\begin{aligned}y_1 + (D-2)y_2 - 2y_3 &= 0 \\[-(D-1)(D-2) - 2]y_2 + [2(D-1) - 1]y_3 &= 1 \\[(2D-4) - 1]y_2 + [(D-1) - 4]y_3 &= x \\y_1 + (D-2)y_2 - 2y_3 &= 0 \\ \implies (-D^2 + 3D - 4)y_2 + (2D - 3)y_3 &= 1 \\(2D - 5)y_2 + (D - 5)y_3 &= x\end{aligned}$$

Now we have a problem, because none of the equations features an unknown without the operator  $D$ , which stopped us when attempting an intuitive evaluation. However, this time we look at it as a general elimination. We choose a pivot, and the third row looks more appealing (just one derivative instead of the second order).

$$\begin{aligned}y_1 + (D-2)y_2 - 2y_3 &= 0 \\(2D - 5)y_2 + (D - 5)y_3 &= x \\(-D^2 + 3D - 4)y_2 + (2D - 3)y_3 &= 1\end{aligned}$$

If  $D^2 - 3D + 4$  had  $2D - 5$  as a factor, we would easily get rid of the corresponding term by subtracting a suitable multiple of the second row from the third. Unfortunately, this is not true. We will therefore multiply the third row by  $2D - 5$ , and then add the second row multiplied by  $D^2 - 3D + 4$ . This second row then becomes

$$(2D^3 - 11D^2 + 23D - 20)y_2 + (D^3 - 8D^2 + 19D - 20)y_3 = (D^2 - 3D + 4)x = 4x - 3$$

Here we go,

$$\begin{aligned} & y_1 + (D - 2)y_2 - 2y_3 = 0 \\ & (2D - 5)y_2 + (D - 5)y_3 = x \\ & (2D - 5)(-D^2 + 3D - 4)y_2 + (2D - 5)(2D - 3)y_3 = 1 \\ & \qquad \qquad \qquad y_1 + (D - 2)y_2 - 2y_3 = 0 \\ \implies & \qquad \qquad \qquad (2D - 5)y_2 + (D - 5)y_3 = x \\ & (-2D^3 + 11D^2 - 23D + 20)y_2 + (4D^2 - 16D + 15)y_3 = -5 \\ & \qquad \qquad \qquad y_1 + (D - 2)y_2 - 2y_3 = 0 \\ \implies & (2D - 5)y_2 + (D - 5)y_3 = x \\ & (D^3 - 4D^2 + 3D - 5)y_3 = 4x - 8 \end{aligned}$$

The last row is the equation we expect after elimination: It only features one unknown function and has order three:

$$y_3''' - 4y_3'' + 3y_3' - 5y_3 = 4x - 8.$$

The first row offers a reduction formula that will determine  $y_1$  once we find the other two:

$$y_1 = -y_2' + 2y_2 + 2y_3.$$

In the second row we would expect a reduction formula for  $y_2$ , and we do get one:

$$2y_2' - 5y_2 = x - y_3' + 5y_3.$$

Unfortunately, it has the form of a differential equation, but we cannot do any better than this.

Obviously, elimination has its problems even using differential calculus.

### 27c.1 Differential calculus and homogeneous equations

Assume that we were given some equation of the form  $L[y] = 0$  with constant coefficient. Since the left-hand side is linear and has constant coefficients, we should be able to write it as  $L[y] = p(D)[y]$ , where  $p(D)$  is some polynomial with the differential operator  $D$ .

For instance, given the equation  $y'' - 4y' + 3y = 0$ , we can write it as  $(D^2 - 4D + 3)[y] = 0$ .

We can factor the polynomial into linear factors and obtain

$$(D - 3)(D - 1)[y] = 0.$$

We easily observe that if some function  $y$  satisfies  $(D - 1)[y] = 0$ , then it also satisfies  $(D - 3)(D - 1)[y] = 0$ . Since  $(D - 3)(D - 1) = (D - 1)(D - 3)$  also for differential operators, similar argument tells us that solutions of the equation  $(D - 3)[y] = 0$  solve the given equation. Remarkably, this also works the other way around, every solution of the equation  $(D - 3)(D - 1)[y] = 0$  must solve one of the simpler equations.

This is the key to our favorite method of solving homogeneous equations. Given some equation of the form  $p(D)[y] = 0$ , if the polynomial  $p(D)$  factors into linear factors, that is, if the polynomial  $p(\lambda)$  has only real roots, then we obtain all solutions of this equation simply by solving equations corresponding to the linear factors. And since equations of the form  $(D - a)[y] = 0$  have the obvious solution  $e^{ax}$ , we see why we build our fundamental systems out of exponentials.

### 27c.2 Differential calculus and the guessing method

The differential calculus offers a way to transform nonhomogeneous equations into homogeneous. To be of any use, we only care about the case when the original equation and the new are both

linear.

Assume that we were given some equation of the form  $L[y] = b(x)$  with constant coefficient, so it is of the form  $p(D)[y] = b(x)$  for some polynomial  $p(D)$  with the differential operator  $D$ .

For instance, given the equation  $y'' - 4y' + 3y = e^{2x}$ , we can write it as  $(D^2 - 4D + 3)[y] = e^{2x}$ .

Assume further that we are able to find some differential operation that can turn the right-hand side  $b(x)$  into zero, and to be of any use, assume that this operator is also some polynomial in  $D$ , we can call it  $q(D)$ .

For instance, we know that  $e^{2x}$  solves  $y' - 2y = 0$ , that is,  $(D - 2)[y] = 0$ , which tells us that  $(D - 2)[e^{2x}] = 0$ . In this case therefore  $q(D) = D - 2$ .

Now we apply this  $q(D)$  to the original equation  $p(D)[y] = b(x)$  and obtain a new equation  $q(D)p(D)[y] = 0$ . This new equation is homogeneous, and it is still linear with constant coefficients. Moreover, every solution of the original equation is also a solution of the new equation. This means that we can actually find the solutions of the original nonhomogeneous problems among solutions of the new homogeneous problem.

If  $p$  and  $q$  have distinct roots, and therefore decompose into distinct linear factors, we can easily separate solutions from the original homogeneous equation (these solve  $p(D)[y] = 0$ ) from the solutions that are new, they must come from the factors of  $q(D)$ . Thus we look for our particular solutions among the solutions of the equation  $q(D)[y] = 0$ , which in our particular example means among solutions of  $y' - 2y = 0$ . This leads to the guess  $A e^{2x}$ .

Things get more problematic when  $p(D)$  and  $q(D)$  share linear factors, and careful analysis shows the need for a corrective factor.

This approach works for any right-hand side for which we can find a linear differential operator that turns it into zero, and when we make a survey, we end up with the usual suspects: polynomials, exponentials, sines and cosines, and products of the three.

### 31. Finding eigenvalues and eigenvectors

First we recall the definitions of key notions.

**Definition 31.1.**

Consider an  $n \times n$  matrix  $A$ .

A number  $\lambda$  is called an **eigenvalue** of  $A$  if there is a non-zero vector  $\vec{x}$  such that  $A\vec{x} = \lambda\vec{x}$ .

Vectors  $\vec{x}$  with this property are then called **eigenvectors** of  $A$  associated with (corresponding to) the eigenvalue  $\lambda$ .

The relationship is symmetric, we may also say that  $\lambda$  is the eigenvalue associated with an eigenvector  $\vec{x}$ .

What do we need to do if we want to find eigenvalues and eigenvectors? Unfortunately, there is no explicit formulas for them, the standard answer goes as follows. First we find the **characteristic polynomial**  $p(\lambda) = \det(A - \lambda E_n)$ , then we solve the equation  $p(\lambda) = 0$  to find eigenvalues. Finally (and assuming that all eigenspaces are one-dimensional), for each eigenvalue  $\lambda$  we determine its eigenvector as a non-zero solution of the system  $(A - \lambda E_n)\vec{x} = \vec{0}$ .

How does this theory work in practice with an  $n \times n$  matrix? We find the polynomial  $p(\lambda)$  by determining its coefficients, which means finding  $n!$  products and sorting them to form  $n + 1$  coefficients. That's a lot of work. Then we need to find all solutions of the corresponding equation, which leads us to techniques outlined in chapter 19. And, finally, we have to solve  $n$  times a system of  $n$  equations, which is also a lot of work, see chapter 23. It is not surprising that this procedure is used mainly when doing calculations by hand for tiny matrices. How about large matrices?

There are shortcuts. We could try to determine the polynomial by matrix elimination, but it would have to be a symbolic elimination, working with expressions featuring  $\lambda$ , which is not exactly very convenient. Even if we use elimination for all the tasks outlined above, we still face roughly  $n^4$  operations just to handle matrices, plus the problem of finding  $n$  roots.

It is no surprise that people looked for other ways to find eigenvalues. We will show elementary yet fairly powerful (and therefore popular) method and some of its extensions. Before we get to them, we have to address one important topic.

The above approach first finds eigenvalues and then finds corresponding eigenvectors. At the heart of the methods that we will shortly see is an opposite process: First we find eigenvectors, then their eigenvalues. Now given an eigenvector  $\vec{x}$ , how does one find the eigenvalue  $\lambda$  that it belongs to?

In the world of theory the answer is simple. The right pair  $\lambda, \vec{x}$  satisfies  $A\vec{x} = \lambda\vec{x}$ , which offers a very simple method. We simply pick some non-zero entry  $x_i$  in the vector  $\vec{x}$ , then we look at the  $i$ th coordinate  $y_i$  in the vector  $A\vec{x}$ , and due to the equality  $A\vec{x} = \lambda\vec{x}$  we must have  $y_i = \lambda x_i$ , that is,  $\lambda = \frac{y_i}{x_i}$ .

However, in practical calculation this is not so simple, because we in fact do not have an eigenvector, but its approximation. The above procedure then would not produce an eigenvalue, but some number which is perhaps close, but the procedure has some weak spots. In particular, if we use different coordinates  $x_i$ , we are likely to end up with different candidates for  $\lambda$ . In the end, another approach proved more fruitful, and it is based on the following question: If we take some vector  $\vec{x}$ , which number  $\lambda$  makes it the closest to being an eigenvector, that is, which number  $\lambda$  minimizes the function  $t \mapsto \|A\vec{x} - \lambda\vec{x}\|$ ?

Which brings us to another question: What is  $\| \cdot \|$ ? Well, by  $\|\vec{y}\|$  we mean the Euclidean norm of the vector  $\vec{y}$ , which is the familiar formula

$$\|\vec{y}\| = \sqrt{\sum_{k=1}^n |y_k|^2}.$$

We will see later that there are other norms, other ways to measure magnitude of vectors, but for now we will be happy with this one. Now we answer the main question.

**Theorem 31.2.**

Let  $A$  be an  $n \times n$  matrix and  $\vec{x} \in \mathbb{R}^n$ . The function

$$t \mapsto \|A\vec{x} - \lambda\vec{x}\|$$

is minimized by the value  $\lambda = \frac{\vec{x}^T A \vec{x}}{\vec{x}^T \vec{x}}$ .

Some readers may recognize the expression in the numerator as the bilinear form determined by the matrix  $A$ , while the expression in the denominator is simply  $\|\vec{x}\|^2$ .

The formula above makes a good sense even in the world of precise calculations.

**Fact 31.3.**

If  $\vec{x}$  is an eigenvector of a matrix  $A$  associated with eigenvalue  $\lambda$ , then  $\frac{\vec{x}^T A \vec{x}}{\vec{x}^T \vec{x}} = \lambda$ .

This useful formula deserves a name.

**Definition 31.4.**

For an  $n \times n$  matrix  $A$  and a vector  $\vec{x} \in \mathbb{R}^n$  we define their **Rayleigh quotient** by the formula

$$\frac{\vec{x}^T A \vec{x}}{\vec{x}^T \vec{x}}.$$

It has many other interesting properties, but we will not need them here.

### 31a. Power method

We will now assume that a given  $n \times n$  matrix is diagonalizable, that is, there exists a basis of  $\mathbb{R}^n$  composed of its eigenvectors. We will also assume that one eigenvalue dominates the others, that is, we are able to order eigenvalues so that  $|\lambda_1| > |\lambda_i|$  for  $i \geq 2$ . Then it is easy to show that if we start with some vector  $\vec{x}_0$  that is not perpendicular to the eigenspace of  $\lambda_1$ , then we have

$$\frac{1}{\lambda_1^k} A^k \vec{x}_0 \rightarrow \vec{v}_1.$$

This is the basis for the starter numerical method for finding eigenvectors and, consequently, eigenvalues.

**Algorithm 31a.1.**

⟨power method for finding the largest eigenvalue and an associated eigenvector⟩

Given: an  $n \times n$  matrix  $A$ , tolerance  $\varepsilon > 0$ , arbitrary initial vector  $\vec{x}_0$ .

If complex eigenvalues are needed, choose a vector with non-zero imaginary part in each component.

**0.** Set  $k = 0$ .

**1.** Compute

$$\vec{x}_{k+1} = \frac{A\vec{x}_k}{\|A\vec{x}_k\|_\infty},$$

If  $\|\vec{x}_{k+1} - \vec{x}_k\|_\infty \geq \varepsilon$ , increase  $k$  by one and go back to step **1**.

△

This algorithm can be improved to continually update our best estimate for  $\lambda_1$ .

**Algorithm 31a.2.**

⟨power method for finding the largest eigenvalue and an associated eigenvector⟩

Given: an  $n \times n$  matrix  $A$ , tolerance  $\varepsilon > 0$ , arbitrary initial vector  $\vec{x}_0$ .

If complex eigenvalues are needed, choose a vector with non-zero imaginary part in each component.

**0.** Set  $k = 0$ , let  $l_0 = \frac{\vec{x}_0^* A \vec{x}_0}{\vec{x}_0^* \vec{x}_0}$ .

**1.** Compute

$$\vec{x}_{k+1} = \frac{|l_k| \cdot A \vec{x}_k}{l_k \cdot \|A \vec{x}_k\|_\infty}, \quad l_{k+1} = \frac{\vec{x}_{k+1}^* A \vec{x}_{k+1}}{\vec{x}_{k+1}^* \vec{x}_{k+1}}.$$

If  $\|\vec{x}_{k+1} - \vec{x}_k\|_\infty \geq \varepsilon$ , increase  $k$  by one and go back to step **1**.

Alternative: Use Euclidean norm, that is,

$$\vec{x}_{k+1} = \frac{|\lambda_k| \cdot A \vec{x}_k}{\lambda_k \cdot \|A \vec{x}_k\|_2}, \quad l_{k+1} = \frac{\vec{x}_{k+1}^* A \vec{x}_{k+1}}{\vec{x}_{k+1}^* \vec{x}_{k+1}}.$$

△

This method works rather well.

**Theorem 31a.3.**

Let  $A$  be an  $n \times n$  matrix with eigenvalues  $\lambda_j$ , where  $|\lambda_1| > |\lambda_j|$  for  $j \geq 2$ . Let  $M = \max_{j \geq 2} |\lambda_j|$ .

For  $\vec{x}_0$  let  $\{l_k\}$ ,  $\{\vec{x}_k\}$  be sequences generated by the power method. Then  $\{l_k\}$  converges to  $\lambda_1$  and  $\{\vec{x}_k\}$  converges to some eigenvector  $\vec{v}$  associated with  $\lambda_1$ .

Moreover,  $|\lambda_1 - l_k| = O\left(\left[\frac{M}{|\lambda_1|}\right]^{2k}\right)$  and  $\|\vec{v}_1 - \vec{x}_k\| = O\left(\left[\frac{M}{|\lambda_1|}\right]^k\right)$ .

It is natural to ask what about the other, smaller eigenvalues. But before we do it, it should be noted that in many applications we are quite happy with knowing just the largest eigenvalue. This is related to an interesting notion.

**Definition 31a.4.**

We define the **spectral radius** of a square matrix  $A$  as

$$\rho(A) = \max_j (|\lambda_j|),$$

where the maximum runs through all eigenvalues  $\lambda_j$  of  $A$  (including possible complex ones).

In some applications, the spectral radius serves as a measure of impact of the given matrix. We see that the power method yields exactly that information, so it is more useful than it looks. Still, it does have its problems. One is the case when the largest eigenvalue has higher multiplicity. We will not show a cure for this, but it should be noted that in many applications this cannot happen due to underlying physical principles of the problem at hand. The power iteration also fails when there are two candidates for the largest eigenvalue (for instance  $-1$  and  $1$ ). There is an approach that can handle this problem, in fact it will be a byproduct of our search for other eigenvalues. It goes like this.

**Algorithm 31a.5.**

⟨inverse power method for finding eigenvalue and an associated eigenvector⟩

Given: an  $n \times n$  matrix  $A$ , arbitrary real number  $\mu$  that is not an eigenvalue of  $A$ , tolerance  $\varepsilon > 0$ , arbitrary non-zero initiation vector  $\vec{x}_0$ .

If complex eigenvalues are needed, choose a vector with non-zero imaginary part in each component.

1. Compute  $B = (A - \mu E_n)^{-1}$ .
  2. Using power iteration, find the largest eigenvalue  $\lambda_B$  of the matrix  $B$  and an associated eigenvector  $\vec{v}$ .
- Output:** The number  $\mu + \frac{1}{\lambda_B}$  is the eigenvalue of  $A$  closest to  $\mu$  with eigenvector  $\vec{v}$ .
- △

Does it really work? Validity of this approach is based on the following facts from linear algebra.

**Fact 31a.6.**

Let  $A$  be a matrix and  $\lambda$  its eigenvalue with eigenvector  $\vec{v}$ . Then the following are true.

- (i) The matrix  $A^{-1}$  has eigenvalue  $\frac{1}{\lambda}$  with eigenvector  $\vec{v}$ .
- (ii) For any  $\mu \in \mathbb{R}$ , the matrix  $A - \mu E_n$  has eigenvalue  $\lambda - \mu$  with eigenvector  $\vec{v}$ .

Let's see. The inverse power method found the eigenvalue  $\lambda$  that is the largest among eigenvalues of  $(A - \mu E_n)^{-1}$ , that is, the one that is furthest from the origin. Then  $\frac{1}{\lambda}$  is the eigenvalue of  $A - \mu E_n$  that is closest to the origin. Consequently,  $\frac{1}{\lambda} + \mu$  must be an eigenvalue of  $A = (A - \mu E_n) + \mu E_n$ . If we want  $\frac{1}{\lambda}$  to be as close to possible to zero, the number  $\frac{1}{\lambda} + \mu$  must be as close as possible to  $\mu$ .

How does it work in practice? We start with the plain power method, it provides us with the largest eigenvalue. If this fails, then very likely there are more candidates for this title. Then it is enough to check on the matrix  $B = A - \mu E_n$ , where  $\mu$  is a preferably small number. This shifts eigenvalues a bit so that the two candidates for the largest eigenvalue of  $A$  are no longer eigenvalues of the same magnitude for  $B$ . With a bit of luck, the power method applied to  $B$  yields some  $\lambda_B$ , then  $\lambda = \mu + \lambda_B$  is an eigenvalue of  $A$ . The opposite shift  $A + \mu E_n$  should then yield the other candidate.

Then it is time to hunt for the other eigenvalues. If we apply the power iteration to the matrix  $A^{-1}$ , the algorithm above should yield the smallest eigenvalue. This leaves us with two ranges where all the other eigenvalues should be, and we can chose  $\mu$  from this range to get at them.

This is a very nice method that works also with complex eigenvalues. Does it have any disadvantages? Theoretically, there is a gap, because for this to work we need all eigenvalues to be simple. However, in many practical problems it happens that eigenvalues of higher multiplicity are ruled out by the nature of the studied process.

Unfortunately, there is a more serious problem, the need to determine the inverse matrix. This is a serious time eater for larger matrices. We will need to dedicate a chapter to calculations like this to see what is happening. We will see that there is really no way to make this task simple, so for larger matrices we will also want a different approach.

Before we do that, one parting shot. In many practical problems we do not really need all eigenvalues, we just need to make sure that eigenvalues are not too large and knowledge of the largest one is enough to confirm it. Then the power iterations works well.

## 31b. Deflation

If we want an eigenvalue that is not the largest one, we need to get  $\lambda_1$  out of the picture. An obvious idea is to stick with vectors whose decompositions do not use  $\vec{v}_1$ , then  $\lambda_1$  does not appear in formulas and power iteration leads us to the second largest eigenvalue. This process can be repeated as necessary.

There is one problem with this approach. While theoretically it should be easy to steer clear of  $\lambda_1$ , computations on computers involve errors and the  $\vec{v}_1$  component is bound to creep into our calculations sooner or later, so we end up with  $\lambda_1$ . In the end. The solution to this problem is to keep removing unwanted components at every step of the iteration using projection.

**Algorithm 31b.1.**

⟨deflation methoda for finding eigenvalue and an associated eigenvector⟩

Given: an  $n \times n$  matrix  $A$  and its eigenvalues  $\lambda_1$  through  $\lambda_N$  with eigenvectors  $\vec{v}_1$  through  $\vec{v}_N$  satisfying  $\|v_i\| = 1$ , tolerance  $\varepsilon > 0$ , arbitrary initial vector  $\vec{x}_0$ .

If complex eigenvalues are needed, choose a vector with non-zero imaginary part in each component.

**0.** Set  $k = 0$ .

**1.** Let  $\vec{y} = \vec{x}_k - \sum_{j=1}^N \lambda_j (\vec{x}_k \odot \vec{v}_j) \vec{v}_j$ . Compute  $\vec{x}_{k+1} = \frac{A\vec{y}}{\|A\vec{y}\|}$ ,  $l_{k+1} = \vec{x}_{k+1}^* A \vec{x}_{k+1}$ .

If  $\|\vec{x}_{k+1} - \vec{x}_k\|_\infty \geq \varepsilon$ , increase  $k$  by one and go back to step **1**.

Then  $\lambda_k \rightarrow \lambda_{N+1}$ ,  $\vec{x}_k \rightarrow \vec{v}_{N+1}$ .

△

## 34. Transformations and their applications to ODEs

### Definition 34.1.

Let  $\{a_k\}_{k=0}^{\infty} = \{a_0, a_1, a_2, \dots\}$  be a sequence of real (complex) numbers.

The symbol  $\sum_{k=0}^{\infty} a_k = a_0 + a_1 + a_2 + a_3 + \dots$  is called a series.

We say that a given series **converges** (to a number  $A$ ), denoted  $\sum_{k=0}^{\infty} a_k = A$ , if

$$\lim_{N \rightarrow \infty} \left( \sum_{k=0}^N a_k - A \right) = 0.$$

We say that the given series **konverguje** if it converges to some number  $A$ . Otherwise we say that the series **diverges**.

We say that the given series  $\sum_{k=0}^{\infty} a_k$  **converges absolutely** if  $\sum_{k=0}^{\infty} |a_k|$  converges.

### Definition 34.2.

We define the sum of two series  $\sum_{k=0}^{\infty} a_k + \sum_{k=0}^{\infty} b_k$  as the series  $\sum_{k=0}^{\infty} (a_k + b_k)$ .

We define multiplication of a series by a number  $c \cdot \sum_{k=0}^{\infty} a_k$  as the series  $\sum_{k=0}^{\infty} (ca_k)$ .

### Theorem 34.3.

Let  $\sum_{k=0}^{\infty} a_k = A$  and  $\sum_{k=0}^{\infty} b_k = B$  be convergent series.

Then also the series  $\sum_{k=0}^{\infty} (a_k + b_k) = A + B$  and  $\sum_{k=0}^{\infty} (ca_k) = cA$  converge.

### Theorem 34.4.

Let  $N > 0$ , consider a series  $\sum_{k=0}^{\infty} a_k$ .

$\sum_{k=0}^{\infty} a_k$  converges if and only if the series  $\sum_{k=N}^{\infty} a_k$  converges. Then we also have

$$\sum_{k=0}^{\infty} a_k = \sum_{k=0}^{N-1} a_k + \sum_{k=N}^{\infty} a_k.$$

**Definition 34.5.**

Consider series  $\sum_{k=0}^{\infty} a_k$ ,  $\sum_{k=0}^{\infty} b_k$ . We define their **Cauchy product**  $\left(\sum_{k=0}^{\infty} a_k\right) \cdot \left(\sum_{k=0}^{\infty} b_k\right)$  as the series  $\sum_{k=0}^{\infty} c_k$ , where  $\{c_k\} = \{a_k\} * \{b_k\}$ , that is,

$$c_0 = a_0 b_0,$$

$$c_1 = a_0 b_1 + a_1 b_0,$$

$$c_2 = a_0 b_2 + a_1 b_1 + a_2 b_0,$$

$$\vdots$$

$$c_n = \sum_{k=0}^n a_k b_{n-k}.$$

**Theorem 34.6.**

Let  $\sum_{k=0}^{\infty} a_k$  be a series. If it converges but not absolutely, then the following are true:

- It is possible to choose signs  $\varepsilon_k = \pm 1$  so that the series  $\sum \varepsilon_k a_k$  diverges.
- There exists a subset  $M \subset \mathbb{N}_0$  such that  $\sum_{k \in M} a_k$  diverges.
- For arbitrary  $C$  including  $\pm\infty$  we can permute the order of terms in the series so that  $\sum a_{\pi(k)} = C$ .

**Theorem 34.7.**

Let  $\sum a_k$  be a series.

If the series  $\sum |a_k|$  converges, then also  $\sum a_k$  converges and  $\left| \sum_{k=0}^{\infty} a_k \right| \leq \sum_{k=0}^{\infty} |a_k|$ .

**Theorem 34.8.**

Let  $\sum_{k=0}^{\infty} a_k$  be a series. If  $\sum_{k=0}^{\infty} a_k = A$  converges absolutely, then the following are true:

- For any convergent series  $\sum_{k=0}^{\infty} b_k = B$  also their Cauchy product converges and

$$\left(\sum_{k=0}^{\infty} a_k\right) \left(\sum_{k=0}^{\infty} b_k\right) = AB.$$

- For every choice of signs  $\varepsilon_k = \pm 1$  the series  $\sum \varepsilon_k a_k$  converges.
- For every subset  $M \subset \mathbb{N}_0$  the series  $\sum_{k \in M} a_k$  converges and denoting  $M' = \mathbb{N}_0 - M$  we have

$$\sum_{k \in M} a_k + \sum_{k \in M'} a_k = A.$$

- For any permutation of terms in the series we have  $\sum_{k=0}^{\infty} a_{\pi(k)} = A$ .

**Definition 34.9.**

Let  $a \in \mathbb{R}$ .

By a **power series with center  $a$**  we mean any series of the form  $\sum_{k=0}^{\infty} a_k(x-a)^k$ , where  $a_k \in \mathbb{R}$  and  $x$  is a variable (a parameter).

**Theorem 34.10.**

Let  $\sum_{k=0}^{\infty} a_k(x-a)^k$  be a power series.

There exists a number  $r \in \mathbb{R}_0^+ \cup \{\infty\}$  such that the series converges absolutely for  $|x-a| < r$  and diverges for  $|x-a| > r$ .

This number is called the **radius of convergence** of this series.

**Fact 34.11.**

Let  $f$  be a function such that there is a power series with center  $a$  and  $r > 0$  so that  $f(x) = \sum_{k=0}^{\infty} a_k(x-a)^k$  on  $U_r(a)$ . Then for every  $k \in \mathbb{N}$  we also have

$$a_k = \frac{f^{(k)}(a)}{k!}.$$

**Definition 34.12.**

Let a function  $f$  have derivatives of all orders at a point  $a$ .

We define its **Taylor series** with center  $a$  as

$$\sum_{k=0}^{\infty} \frac{f^{(k)}(a)}{k!} (x-a)^k.$$

Finding such a series is called **expanding the given function into a power/Taylor series** (with center  $a$ ).

**Theorem 34.13.**

Let a function  $f$  have derivatives of all orders on some neighborhood  $U_r(a)$  with  $r > 0$ . If there exists  $M > 0$  such that  $|f^{(k)}(x)| \leq M$  for all  $k \in \mathbb{N}_0$  and  $x \in U_r(a)$ , then  $f(x) = \sum_{k=0}^{\infty} \frac{f^{(k)}(a)}{k!} (x-a)^k$  on  $x \in U_r(a)$ .

**Fact 34.14.**

$$\begin{aligned} \frac{1}{1-x} &= \sum_{k=0}^{\infty} x^k = 1 + x + x^2 + x^3 + x^4 + \dots, \quad x \in (-1, 1); \\ e^x &= \sum_{k=0}^{\infty} \frac{x^k}{k!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots, \quad x \in \mathbb{R}; \\ \sin(x) &= \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!} = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots, \quad x \in \mathbb{R}; \\ \cos(x) &= \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{(2k)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots, \quad x \in \mathbb{R}; \\ \ln(1+x) &= \sum_{k=1}^{\infty} (-1)^{k+1} \frac{x^k}{k} = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots, \quad x \in (-1, 1). \end{aligned}$$

**Theorem 34.15.**

Let  $a \in \mathbb{R}$ , assume that power series  $\sum_{k=0}^{\infty} a_k(x-a)^k = f(x)$ ,  $\sum_{k=0}^{\infty} b_k(x-a)^k = g(x)$  have radii of convergence  $r_f$  and  $r_g$ .

(i) For all  $a, b \in \mathbb{R}$  we have

$$af(x) + bg(x) = a \left( \sum_{k=0}^{\infty} a_k(x-a)^k \right) + b \left( \sum_{k=0}^{\infty} b_k(x-a)^k \right) = \sum_{k=0}^{\infty} (aa_k + bb_k)(x-a)^k$$

and this series has radius of convergence  $r = \min(r_f, r_g)$ .

(ii) We have

$$f(x) \cdot g(x) = \left( \sum_{k=0}^{\infty} a_k(x-a)^k \right) \cdot \left( \sum_{k=0}^{\infty} b_k(x-a)^k \right) = \sum_{k=0}^{\infty} \left( \sum_{i=0}^k a_i b_{k-i} \right) (x-a)^k$$

and this series has radius of convergence  $r = \min(r_f, r_g)$ .

**Theorem 34.16.**

Let a power series  $\sum_{k=0}^{\infty} a_k(x-a)^k = f(x)$  have radius of convergence  $r > 0$ . Then the following are true:

(i) For every  $c \in \mathbb{R}$  we have  $f(x-c) = \sum_{k=0}^{\infty} a_k((x-c)-a)^k = \sum_{k=0}^{\infty} a_k(x-(a+c))^k$ .

(ii) For every  $n \in \mathbb{N}$  we have  $(x-a)^n f(x) = \sum_{k=0}^{\infty} a_k(x-a)^{k+n} = \sum_{k=n}^{\infty} a_{k-n}(x-a)^k$ .

(iii) If  $\lim_{x \rightarrow a} \left( \frac{f(x)}{x-a} \right)$  converges, then  $\frac{1}{(x-a)} f(x) = \sum_{k=1}^{\infty} a_k(x-a)^{k-1} = \sum_{k=0}^{\infty} a_{k+1}(x-a)^k$ .

All these series have radius of convergence  $r_f$ .

**Theorem 34.17.**

Let a power series  $\sum_{k=0}^{\infty} a_k(x-a)^k = f(x)$  have radius of convergence  $r > 0$ . Then the following are true:

(i) The function  $f$  is continuous.

(ii) The function  $f$  is differentiable on  $U_r(a)$  and on this set we have

$$f'(x) = \sum_{k=1}^{\infty} k a_k (x-a)^{k-1}.$$

(iii) The function  $f$  has an antiderivative on  $U_r(a)$  and

$$\int f(x) dx = \sum_{k=0}^{\infty} \frac{a_k}{k+1} (x-a)^{k+1} + C.$$

(iv) The function  $f$  has derivatives of all orders on  $U_r(a)$  and for every  $n \in \mathbb{N}$  we have

$$f^{(n)}(x) = \sum_{k=n}^{\infty} k(k-1) \cdots (k-n+1) a_k (x-a)^{k-n} = \sum_{k=n}^{\infty} \frac{k!}{(k-n)!} a_k (x-a)^{k-n}.$$

**Algorithm 34.18.**

$\langle$ principle of transformation $\rangle$

Consider a set  $A$  with an operation  $\circ_A$  and a set  $B$  with an operation  $\circ_B$ . Let  $T$  be a 1-1 mapping that satisfies

$$T(x \circ_A y) = T(x) \circ_B T(y)$$

for all  $x, y \in A$ . Then instead of evaluating  $x \circ_A y$  we can use this procedure:

1. We transport the problem to the world  $B$ :  $T(x), T(y)$ .
2. We solve the problem in the world  $B$ :  $T(x) \circ_B T(y)$ .
3. We move the result back to the world  $A$ :  $T^{-1}(T(x) \circ_B T(y))$ .

$\triangle$

**Fact 34.19.**

Let  $a \in \mathbb{R}$ , let  $V$  be the space of functions that can be expanded into a power series with center  $a$ . For  $f$  define a mapping  $T$  by the condition  $T(f) = \{a_k\}_{k=0}^{\infty}$  if  $f(x) = \sum_{k=0}^{\infty} a_k(x-a)^k$  on some  $U_r(a)$ .

Take any  $f, g \in V$  and assume that  $T(f) = \{a_k\}_{k=0}^{\infty}$  and  $T(g) = \{b_k\}_{k=0}^{\infty}$ . Then the following are true:

- (i)  $T(\alpha f + \beta g) = \{\alpha a_k + \beta b_k\}_{k=0}^{\infty}$  for all  $\alpha, \beta \in \mathbb{R}$ .
- (ii)  $T((x-a)f) = \{0, a_0, a_1, a_2, \dots\}$ .
- (iii) If  $f(a) = 0$ , then  $T(\frac{1}{x-a}f) = \{a_{k+1}\}_{k=0}^{\infty} = \{a_1, a_2, a_3, \dots\}$ .
- (iv)  $T(f') = \{(k+1)a_{k+1}\}_{k=0}^{\infty} = \{a_1, 2a_2, 3a_3, \dots\}$ .

**Definition 34.20.**

Let  $f$  be a function defined on an interval  $I = \langle a, a+T \rangle$  for some  $a \in \mathbb{R}$ ,  $T > 0$ . Its **periodic extension** is defined as the function

$$f(t) = f(t - kT) \quad \text{for } t \in \langle a + kT, a + (k+1)T \rangle.$$

**Theorem 34.21.**

Let  $f$  be a function that is  $T$ -periodic. Let  $\omega = \frac{2\pi}{T}$ .  
If the series

$$\frac{a_0}{2} + \sum_{k=1}^{\infty} [a_k \cos(k\omega t) + b_k \sin(k\omega t)]$$

converges to  $f(x)$  on  $\mathbb{R}$  uniformly, then necessarily

$$a_k = \frac{2}{T} \int_0^T f(t) \cos(k\omega t) dt \text{ pro } k \in \mathbb{N}_0,$$

$$b_k = \frac{2}{T} \int_0^T f(t) \sin(k\omega t) dt \text{ pro } k \in \mathbb{N}.$$

**Definition 34.22.**

Let  $f$  be a function integrable on an interval  $\langle a, a + T \rangle$ .

We define its **Fourier series** as

$$\frac{a_0}{2} + \sum_{k=1}^{\infty} [a_k \cos(k\omega t) + b_k \sin(k\omega t)],$$

where  $\omega = \frac{2\pi}{T}$  and

$$a_k = \frac{2}{T} \int_0^T f(t) \cos(k\omega t) dt \text{ pro } k \in \mathbb{N}_0,$$

$$b_k = \frac{2}{T} \int_0^T f(t) \sin(k\omega t) dt \text{ pro } k \in \mathbb{N}.$$

We write

$$f \sim \frac{a_0}{2} + \sum_{k=1}^{\infty} [a_k \cos(k\omega t) + b_k \sin(k\omega t)].$$

**Theorem 34.23.**

(Jordan criterion)

Let  $f$  be a function that is piecewise continuous on some interval  $I$  of length  $T$ , assume that it has derivative  $f'$  that is piecewise continuous on  $I$ .

Consider its periodic extension, call it  $f$  again for simplicity.

Let  $f \sim \frac{a_0}{2} + \sum_{k=1}^{\infty} [a_k \cos(k\omega t) + b_k \sin(k\omega t)]$ . Then for every  $t \in \mathbb{R}$  we have

$$\frac{a_0}{2} + \sum_{k=1}^{\infty} [a_k \cos(k\omega t) + b_k \sin(k\omega t)] = \frac{1}{2} \left[ \lim_{x \rightarrow t^-} (f(x)) + \lim_{x \rightarrow t^+} (f(x)) \right].$$

If, moreover,  $f$  is continuous on  $\mathbb{R}$ , then  $\frac{a_0}{2} + \sum_{k=1}^{\infty} [a_k \cos(k\omega t) + b_k \sin(k\omega t)] = f$  and the convergence of this series is uniform on  $\mathbb{R}$ .

**Fourier series in amplitude-phase form:**

$$f \sim \frac{a_0}{2} + \sum_{k=1}^{\infty} A_k \sin(k\omega t + \varphi_k).$$

**Fourier series in complex form:**

$$f \sim \sum_{k=-\infty}^{\infty} c_k e^{ik\omega t},$$

where

$$c_k = \frac{1}{T} \int_a^{a+T} f(t) e^{-ik\omega t} dt.$$

**Definition 34.24.**

Let  $f(t)$  be a function integrable on  $\mathbb{R}$ . Its **Fourier transform**  $\mathcal{F}[f](\omega)$  is defined by the formula

$$\mathcal{F}[f] : \omega \mapsto \int_{-\infty}^{\infty} f(t) e^{-i\omega t} dt.$$

Often we also write  $\mathcal{F}[f] = \hat{f}$ .

**Theorem 34.25.**

Let  $f$  be a function integrable on  $\mathbb{R}$  that is continuous with possible exception of finitely many jump discontinuities. If we redefine  $f$  as  $f(x) = \frac{1}{2}(f(x^-) + f(x^+))$  at these points, then

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \hat{f}(\omega) e^{i\omega t} d\omega.$$

Fourier transform:

**Dictionary:**

$$\mathcal{F}[1] = 2\pi\delta(\omega);$$

$$\mathcal{F}[H(t)] = \pi\delta(\omega) + \frac{1}{i\omega};$$

$$\mathcal{F}[e^{i\omega_0 t}] = 2\pi\delta(\omega - \omega_0);$$

$$\mathcal{F}[e^{i\omega_0 t} H(t)] = \frac{1}{\omega_0 + i\omega};$$

$$\mathcal{F}[\sin(\omega_0 t)] = \frac{\pi}{2i} [\delta(\omega - \omega_0) - \delta(\omega + \omega_0)];$$

$$\mathcal{F}[\delta(t)] = 1;$$

$$\mathcal{F}[\text{sgn}(t)] = \frac{2}{i\omega};$$

$$\mathcal{F}[e^{-\omega_0|t|}] = 2\omega_0\omega_0^2 + \omega^2; \quad (\omega_0 > 0)$$

$$\mathcal{F}[t e^{i\omega_0 t} H(t)] = \frac{1}{(\omega_0 + i\omega)^2}; \quad (\omega_0 > 0)$$

$$\mathcal{F}[\cos(\omega_0 t)] = \frac{\pi}{2i} [\delta(\omega - \omega_0) + \delta(\omega + \omega_0)].$$

**Grammar:**

$$\mathcal{F}[\alpha f + \beta g] = \alpha\mathcal{F}[f] + \beta\mathcal{F}[g];$$

$$\mathcal{F}[f(t - a)] = e^{-ia\omega} \hat{f}(\omega);$$

$$\mathcal{F}[t f(t)] = i\hat{f}'(\omega);$$

$$\mathcal{F}[e^{iat} f(t)] = \hat{f}(\omega - a);$$

$$\mathcal{F}[f'(t)] = i\omega \hat{f}(\omega).$$

here  $\hat{f} = \mathcal{F}[f]$

**Definition 34.26.**

For  $f: \langle 0, \infty \rangle \mapsto \mathbb{R}$  we define its **Laplace transform**  $\mathcal{L}\{f(t)\}$  by the formula

$$\mathcal{L}\{f(t)\} : s \mapsto \int_0^{\infty} f(t)e^{-st} dt,$$

assuming that this converges for at least one  $s \in \mathbb{R}$ .

Notation:  $\mathcal{L}\{f(t)\}(s)$ ,  $\mathcal{L}\{f\}$ ,  $F(s)$ , alternatively  $f(t) \hat{=} F(s)$ .

**Definition 34.27.**

**Heaviside function** is defined as

$$H(t) = \begin{cases} 1, & t \geq 0; \\ 0, & t < 0. \end{cases}$$

**Definition 34.28.**

$$\mathcal{L}_0 = \left\{ f(t) : [0, \infty) \mapsto \mathbb{R}; f \text{ piecewise continuous and } |f(t)| \leq C e^{\alpha t} \text{ for some } C, \alpha > 0 \right\}.$$

**Fact 34.29.** (dictionary)

- $\mathcal{L}\{e^{\alpha t} H(t)\} = \frac{1}{s-\alpha}$ ,  $s > \alpha$  for all  $\alpha \in \mathbb{R}$ ;
- $\mathcal{L}\{t^n H(t)\} = \frac{n!}{s^{n+1}}$ ,  $s > 0$  for all  $n \in \mathbb{N}_0$ ;
- $\mathcal{L}\{\sin(\omega t) H(t)\} = \frac{\omega}{s^2 + \omega^2}$ ,  $s \in \mathbb{R}$  for all  $\omega \in \mathbb{R}$ ;
- $\mathcal{L}\{\cos(\omega t) H(t)\} = \frac{s}{s^2 + \omega^2}$ ,  $s \in \mathbb{R}$  for all  $\omega \in \mathbb{R}$ .

**Theorem 34.30.** (grammar)

Let  $f \in \mathcal{L}_0$ .

(i) (linearity)  $\mathcal{L}\{af(t) + bg(t)\} = a\mathcal{L}\{f(t)\} + b\mathcal{L}\{g(t)\}$  for every  $g \in \mathcal{L}_0$  and all  $a, b \in \mathbb{R}$ .

(ii) (change of scale)  $\mathcal{L}\{f(at)\} = \frac{1}{a}\mathcal{L}\{f(t)\}|_{s/a}$  for every  $a > 0$ ;

(iii) (shift in image)  $\mathcal{L}\{e^{at} f(t)\} = \mathcal{L}\{f(t)\}|_{s-a}$  for every  $a \in \mathbb{R}$ ;

(iv) (derivative of image)  $\mathcal{L}\{t^n f(t)\} = (-1)^n \frac{d^n}{ds^n} \mathcal{L}\{f(t)\}$  for every  $n \in \mathbb{N}$ ;

(v) (derivative of preimage) If  $f^{(n)} \in \mathcal{L}_0$ , then  $\mathcal{L}\{f^{(n)}(t)\} = s^n \mathcal{L}\{f(t)\}(s) - s^{n-1} f(0^+) - s^{n-2} f'(0^+) - \dots - s f^{(n-2)}(0^+) - f^{(n-1)}(0^+)$ ;

(vi) (integral of preimage)  $\mathcal{L}\{\int_0^t f(u) du\} = \frac{1}{s} \mathcal{L}\{f(t)\}$ .

(vii) (integral of image) If  $\lim_{t \rightarrow 0^+} \left(\frac{f(t)}{t}\right)$  converges, then

$$\mathcal{L}\left\{\frac{1}{t} f(t)\right\} = \int_s^{\infty} \mathcal{L}\{f(t)\}(q) dq.$$

**Theorem 34.31.**

Assume that for some  $f, g \in \mathcal{L}_0$  we have  $\mathcal{L}\{f\} = \mathcal{L}\{g\}$  on some  $\langle s_0, \infty \rangle$ . Then  $f = g$  up to at most countable set of isolated points.

If, moreover,  $f$  and  $g$  are continuous from the right everywhere, then  $f = g$ .

**Theorem 34.32.**

If  $F(s)$  is a rational function whose numerator has smaller degree than the denominator, then the inverse  $\mathcal{L}^{-1}\{F(s)\}$  exists and can be found using partial fractions decomposition.

**Fact 34.33.**

$$\mathcal{L}^{-1}\left\{\frac{1}{s-\alpha}\right\} = e^{\alpha t},$$

$$\mathcal{L}^{-1}\left\{\frac{1}{s^n}\right\} = \frac{1}{(n-1)!}t^{n-1},$$

$$\mathcal{L}^{-1}\left\{\frac{\omega}{s^2+\omega^2}\right\} = \sin(\omega t),$$

$$\mathcal{L}^{-1}\left\{\frac{s}{s^2+\omega^2}\right\} = \cos(\omega t).$$

**Theorem 34.34.**

(1)  $\mathcal{L}^{-1}$  is linear;

$$(2) \mathcal{L}^{-1}\{F(s-a)\} = e^{at}\mathcal{L}^{-1}\{F(s)\}; \quad (2) F(s-a) \hat{=} e^{at}f(t);$$

$$(3) \mathcal{L}^{-1}\{F(as)\} = \frac{1}{a}\mathcal{L}^{-1}\{F(s)\}\Big|_{t/a}; \quad (3) F(as) \hat{=} \frac{1}{a}f\left(\frac{t}{a}\right);$$

$$(4) \mathcal{L}^{-1}\{sF(s)\} = [\mathcal{L}^{-1}\{F(s)\}]' + \mathcal{L}^{-1}\{F(s)\}(0^+); \quad (4) sF(s) \hat{=} f'(t) + f(0^+);$$

$$(5) \mathcal{L}^{-1}\{F'(s)\} = -t\mathcal{L}^{-1}\{F(s)\}. \quad (5) F'(s) \hat{=} -tf(t).$$

**Fact 34.35.**

Let  $f$  be a function defined at least on an interval  $\langle a, b \rangle$ .

Then the function  $f(t)[H(t-a) - H(t-b)]$  has values

$$f(t)[H(t-a) - H(t-b)] = \begin{cases} f(t), & t \in \langle a, b \rangle; \\ 0, & \text{elsewhere.} \end{cases}$$

**Theorem 34.36.**

Let  $f \in \mathcal{L}_0$ ,  $f \hat{=} F$ . Then for every  $a > 0$  we have

$$\mathcal{L}\{f(t-a)H(t-a)\} = e^{-as}\mathcal{L}\{f(t)H(t)\};$$

$$\mathcal{L}^{-1}\{e^{-as}F(s)\} = f(t-a) \cdot H(t-a).$$

$$\mathcal{L}\{f(t)H(t-a)\} = e^{-as}\mathcal{L}\{f(t+a)H(t)\}.$$

**Theorem 34.37.**

Let  $f$  be a function that is  $T$ -periodic on  $\langle 0, \infty \rangle$ . Denote one period as

$$f_T(t) = f(t)[H(t) - H(t-T)]. \quad \text{Then } \mathcal{L}\{f(t)\} = \frac{\mathcal{L}\{f_T(t)\}}{1 - e^{-sT}}.$$

$$\begin{aligned}
\mathcal{L}\{e^{\alpha t}\} &= \frac{1}{s - \alpha} & \mathcal{L}\{e^{\alpha t} f(t)\} &= \mathcal{L}\{f\}(s - a) \\
\mathcal{L}\{\sin(\omega t)\} &= \frac{\omega}{s^2 + \omega^2} & \mathcal{L}\{t f(t)\} &= -[\mathcal{L}\{f\}(s)]' \\
\mathcal{L}\{\cos(\omega t)\} &= \frac{s}{s^2 + \omega^2} & \mathcal{L}\{f'(t)\} &= s\mathcal{L}\{f\}(s) - f(0^+) \\
\mathcal{L}\{1\} &= \frac{1}{s} & \mathcal{L}\{f''(t)\} &= s^2\mathcal{L}\{f\}(s) - sf(0^+) - f'(0^+) \\
\mathcal{L}\{t\} &= \frac{1}{s^2} & \mathcal{L}\left\{\int_0^t f(u) du\right\} &= \frac{1}{s}\mathcal{L}\{f\}(s) \\
\mathcal{L}\{t^n\} &= \frac{n!}{s^{n+1}} & \mathcal{L}\{f(t) \cdot H(t - a)\} &= e^{-as}\mathcal{L}\{f(t + a)H(t)\} \\
& & \mathcal{L}^{-1}\{e^{-as}F(s)\} &= \mathcal{L}^{-1}\{F(s)\} \cdot H(t - a)
\end{aligned}$$

Assume:  $f(t) \hat{=} F(s)$

$$\begin{aligned}
e^{\alpha t} &\hat{=} \frac{1}{s - \alpha} & e^{\alpha t} f(t) &\hat{=} F(s - a) \\
\sin(\omega t) &\hat{=} \frac{\omega}{s^2 + \omega^2} & t f(t) &\hat{=} -F'(s) \\
\cos(\omega t) &\hat{=} \frac{s}{s^2 + \omega^2} & f'(t) &\hat{=} sF(s) - f(0^+) \\
1 &\hat{=} \frac{1}{s} & f''(t) &\hat{=} s^2F(s) - sf(0^+) - f'(0^+) \\
t &\hat{=} \frac{1}{s^2} & \int_0^t f(u) du &\hat{=} \frac{1}{s}F(s) \\
t^n &\hat{=} \frac{n!}{s^{n+1}} & f(t - a) \cdot H(t - a) &\hat{=} e^{-as}F(s) \\
& & \mathcal{L}\{f(t) \cdot H(t - a)\} &= e^{-as}\mathcal{L}\{f(t + a)H(t)\} \\
& & \mathcal{L}^{-1}\{e^{-as}F(s)\} &= f(t - a) \cdot H(t - a)
\end{aligned}$$

### 38. Taylor expansion, asymptotic growth

One of the more important tools in analysis is the **Taylor polynomial** or **Taylor expansion**. Recall that if a function  $f$  has some derivatives on a neighborhood of a given point  $a$ , then we often use approximation

$$\begin{aligned} f(x) &\approx f(a) + f'(a)(x - a) + \frac{1}{2!}f''(a)(x - a)^2 + \cdots + \frac{1}{n!}f^{(n)}(a)(x - a)^n \\ &= \sum_{k=0}^n \frac{1}{k!}f^{(k)}(a) \cdot (x - a)^k. \end{aligned}$$

This is the traditional form, in this book we will find more useful the other popular norm that is obtained by introducing the step size  $h = x - a$ .

$$\begin{aligned} f(a + h) &\approx f(a) + f'(a)h + \frac{1}{2!}f''(a)h^2 + \cdots + \frac{1}{n!}f^{(n)}(a)h^n \\ &\approx \sum_{k=0}^n \frac{1}{k!}f^{(k)}(a)h^k \end{aligned}$$

For nice functions we hope that the polynomial yields a good approximation of the function. The difference between the value  $f(a + h)$  and the Taylor polynomial (called the remainder) can be expressed in several ways and one of the more useful is the “Lagrange form of remainder”. We know that if  $f$  is  $n + 1$ -times continuously differentiable on the interval between  $a$  and  $a + h$ , then there must be some number  $\xi$  in it such that  $\frac{1}{(n+1)!}f^{(n+1)}(\xi)h^{n+1}$  is exactly the difference between  $f(a + h)$  and the Taylor polynomial of degree  $n$ . That is, we have

$$f(a + h) = \sum_{k=0}^n \frac{1}{k!}f^{(k)}(a)h^k + \frac{1}{(n + 1)!}f^{(n+1)}(\xi)h^{n+1}.$$

Surprisingly, even for some very nice functions the remainder need not decrease to zero as we increase the degree of the Taylor polynomial. However, there are known conditions that already guarantee what we want: That by increasing  $n$  to infinity the error goes to zero. For instance, this is true when the function has derivatives of all orders and they share a common bound on the interval between  $a$  and  $a + h$ . Pretty much all elementary functions satisfy this.

Then one can introduce the notion of **Taylor series** and we can “expand” the given function around the center  $a$ :

$$f(a + h) \approx \sum_{k=0}^{\infty} \frac{1}{k!}f^{(k)}(a)h^k.$$

It is important that the reader feel comfortable with Taylor expansions, as they will be one of the more useful concepts in this book.

Working with “infinite polynomials” has its drawbacks, so we often decide to ignore the tail of this series, keeping only first few terms. However, if we want to do it properly we need to understand asymptotic behaviour.

#### 38a. Asymptotic growth

The motivation here is to compare how various expressions behave when the variable is taken “to the extreme”. As a motivation we recall powers. Readers undoubtedly know that powers with higher exponents grow incomparably faster than those with smaller exponents. For instance, the familiar graphs of a line and a parabola leave little doubt that as we substitute larger and larger numbers, then  $x^2$  will totally outstrip  $x$  as they race to infinity.

This domination is so total that the latter becomes negligible compared to the former. For instance, try to take your calculator and substitute  $x = 999999999$  into the expression  $x^2 + 150x$  and then into  $x^2$ . The two answers will be almost the same, the linear term makes next to no difference, even though we tried to help  $x$  by including it many times.

On the other hand, while the expression  $2x$  definitely grows faster to infinity than  $x$ , it is not

the case of total dominance. We can see it in a way similar to the above: If we compare values of  $2x + x$  and of  $2x$ , we see that they never become almost the same, no matter how large number we substitute into them. In fact, in many applications we do not worry too much when an expression is multiplied by a constant, we only care about substantial change in behaviour.

There is a way to see this difference. How do we compare things? Sometimes by subtracting, but this is good for determining fine differences, it does not work well for large differences. For those it is better to divide. Given two expressions  $f$  and  $g$ , the ratio  $\frac{f}{g}$  tells us how they compare.

If we take  $\frac{x^2}{x}$  and let  $x \rightarrow \infty$  (mathematically we express this using a limit), we obtain infinity. Informally,  $x^2$  grows infinitely times faster than  $x$ . Note that we obtain the same information by noting that  $\frac{x}{x^2} \rightarrow 0$  as  $x \rightarrow \infty$ . Similarly, by noting that  $\frac{13x^3+23x^2}{2x^4} \rightarrow 0$  as  $x \rightarrow \infty$  we conclude that the expression  $2x^4$  totally dominates  $13x^3 + 23x^2$  around infinity.

On the other hand,  $\frac{4x^2+13x}{2x^2} \rightarrow 2$  as  $x \rightarrow \infty$ , which shows that the upper expression may be larger, but it does not dominate the expression  $2x^2$ , they grow comparably fast. Incidentally, we also see that the part  $13x$  has no influence on the outcome of the limit, which shows that it can be safely ignored for large  $x$ .

It is very likely that the reader is already familiar with this type of reasoning, as it is quite useful in certain applications. In some situations we might like to say that the expression  $3x^4 - 2x^3 + 17x$  grows like  $3x^4$  when  $x \rightarrow \infty$ , and sometimes we would simply say that it grows like  $x^4$  as  $x \rightarrow \infty$ , in case when the constant does not seem important in our application. When we do such a reasoning, we say that we determine the **asymptotic rate of growth** for the given expression. We will now introduce mathematical concepts that allow us to express such thoughts in a precise manner and work with them efficiently.

**Definition 38a.1.**

Consider a number  $a \in \mathbb{R}$ , or  $a = \pm\infty$ , let  $f, g$  be functions defined on some (reduced) neighborhood of  $a$ .

We say that “ $f$  is  $o(g)$  as  $x \rightarrow a$ ” if  $\lim_{x \rightarrow a} \left( \frac{|f(x)|}{|g(x)|} \right) = 0$ .

We say that “ $f$  is  $O(g)$  as  $x \rightarrow a$ ” if there exist some constant  $C$  and a reduced neighborhood  $P$  of  $a$  such that  $|f| \leq C|g|$  on  $P$ .

The first notion serves for expressing total domination. For instance,  $x^2$  is  $o(x^3)$  as  $x \rightarrow \infty$ . The second notion is used when we want to show that a given expression does not dominate another one.

**Example 38a.a:** Consider  $f(x) = 5x^7 - 135x$ . Since  $5x^7 - 135x \leq 5x^7$ , we can say that  $f$  is  $O(5x^7)$ . We could also say that  $f$  is  $O(5x^8)$ . This notation also allows us to ignore the multiplicative constant at  $x^7$ , for instance, we can say that  $f$  is  $O(2x^7)$ . Indeed, we choose  $C = 3$  and see that

$$5x^7 - 135x \leq 3 \cdot 2x^7 \text{ for } x \in (0, \infty),$$

exactly as the definition requires.

We could also say that  $f$  is  $O(13x^{13})$  as  $x \rightarrow \infty$ .

The correct feeling is that  $O$  is a bit like an inequality  $f \leq g$ , so it should also include the cases of total dominance (little  $o$ ), when the inequality  $f < g$  is “very sharp”, and it is a bit fuzzy, because it does not care about multiplicative constants.

While this all is true, in practice we use it in a very specific way, we try to isolate the most dominant trend in the given expression. In our case we get the best information by stating that  $5x^7 - 135x$  is  $O(x^7)$  as  $x \rightarrow \infty$ .

The fact that we truly captured the asymptotic growth of the given expression can be seen from the fact that it also works the other way around:  $x^7 = O(5x^7 - 135x)$  as  $x \rightarrow \infty$ . Also some of the

other estimates above work this way, for instance  $2x^7 = O(5x^7 - 135x)$  as  $x \rightarrow \infty$ . Put simply,  $x^7$  is the essential part, it is as if we had two “fuzzy inequalities”  $f \leq g$  and  $g \leq f$ , these two expressions behave essentially the same.

On the other hand,  $5x^8$  does not capture the behaviour of  $f$ . True, we have that  $f$  is  $O(5x^8)$ , but that’s just one “inequality” and it is definitely not true that  $5x^8$  is  $O(f)$  as  $x \rightarrow \infty$ , because no matter what constant  $C$  we use to speed it up, the power  $5x^8$  will always outgrow the expression  $5Cx^7 - 135Cx$  if we use large enough  $x$ .

△

There are other notions that work with asymptotic comparison, but these two will do for us. Note that there is no accepted shorthand notation for the two notions. Some authors use  $f \in o(g), x \rightarrow a$ , which makes perfect mathematical sense if we interpret  $o(g)$  as the set of all functions that are dominated by  $g$  around  $a$ . However, most authors (and in particular people who actually use it practically: engineers, physicists etc.) prefer a different notation:  $f = o(g), x \rightarrow a$ , similarly for  $O$ . This notation is much harder to justify as correct mathematics, but it is very convenient and everybody understands it. Which is the reason why I will be using it in this book, although—as a mathematician—I suffer a bit while doing so.

Before we move on, one quick observation. The reader undoubtedly noticed a connection between situations of total dominance and situations when some parts of an expression can be safely ignored. Now we have a language to express it precisely, we also use the opportunity to confirm the hierarchy between the two o’s.

**Fact 38a.2.**

Assume that  $f, g$  are functions defined on a neighborhood of  $a$ . Then the following are true:

- (i) If  $f = o(g)$  as  $x \rightarrow a$ , then  $f = O(g)$  as  $x \rightarrow a$ .
- (ii) If  $f = o(g)$  as  $x \rightarrow a$ , then  $g + f = O(g)$  as  $x \rightarrow a$ .

We looked at powers at infinity as an inspiration, because reader is (or should be) familiar with such reasoning. We will also make use of it in some later chapters. However, we will be more often interested in behaviour of positive powers near zero. They all go to zero there and the question is how fast. The good news is that there are the same patterns as around infinity, just the other way around.

**Fact 38a.3.**

If  $b > a \geq 0$  and  $\beta \in \mathbb{R}$ , then  $\beta h^b = o(h^a)$  as  $h \rightarrow 0$ .

If  $b \geq a \geq 0$  and  $\beta \in \mathbb{R}$ , then  $\beta h^b = O(h^a)$  as  $h \rightarrow 0$ .

The picture on the right shows the difference between  $10h^2$  and  $h$ . You can see that we tried to help the quadratic power by enlarging it ten times, but to no avail, once we got close to zero, it died on us fairly quickly. The thing to remember is that *around zero, higher powers are negligible*.

Here’s a practical take. Imagine that we are asking what is the error of some approximation with step  $h$  and we determine several sources, which then add up. We want to express the essential behaviour of this error. Around infinity we would look for the highest power. However, around zero it is the other way around, so we look for the lowest power.

**Example 38a.b:**

Consider the expression  $3h^2 - 7h^3 + 13h^5$ . Because we have  $7h^3 = o(h^2)$  and  $13h^5 = o(h^2)$  as  $h \rightarrow 0$ , we can conclude that

$$3h^2 - 7h^3 + 13h^5 = O(h^2) \text{ as } h \rightarrow 0.$$

Note that if this was an estimate of some error in calculations, then such an upper estimate is very useful. While it does not tell us much about how such an error looks like, it guarantees that it decreases to zero as  $h \rightarrow 0^+$ . Moreover, it disappears roughly as fast as  $h^7$ , which is a pretty good speed. In numerical mathematics we do this often: We derive an upper estimate for an error, and in the process we ignore higher powers of  $h$ , until we arrive at the key power, the order of the error (here it is 7). And we want this order to be as high as possible, because the higher the order, the faster the error diminishes as we make  $h$  smaller.

Sometimes we want to have a more precise information about the expression, for instance when we know that we will use it in calculations later on. A good way to capture the essence is like this:

$$3h^2 - 7h^3 + 13h^5 = 3h^2 + O(h^3) \text{ as } h \rightarrow 0.$$

Intuitively, this tells us that for small values of  $h$  the expression is pretty much the same as  $2h^2$ , because its other parts are bounded (more or less) by  $h^3$ , which is negligible compared to  $h^2$ .

△

This notation is very useful, because not only does it allow us to capture the substance of expressions, it can be also used very efficiently in calculations.

### Example 38a.c:

Consider the following three expressions:

$$f(h) = 3h^4 + 5h^6 - 4h^7$$

$$g(h) = h^4 + 2h^5 - 4h^6$$

$$h(h) = h^5 + 3h^6 - 2h^7$$

We know that  $f = O(h^4)$ ,  $g = O(h^4)$ , and  $h = O(h^5)$  as  $h \rightarrow 0$ . What can we say about the expression  $f + g + h$ ? First we look at the exact calculation:

$$\begin{aligned} f - g + h &= [3h^4 + 5h^6 - 4h^7] - [h^4 + 2h^5 - 4h^6] + [h^5 + 3h^6 - 2h^7] \\ &= 2h^4 - h^5 + 12h^6 - 6h^7 = O(h^4) \text{ as } h \rightarrow 0. \end{aligned}$$

Now compare this to the following computation:

$$f - g + h = O(h^4) - O(h^4) + O(h^5) = O(h^4) \text{ as } h \rightarrow 0.$$

Does it make sense? When several expressions meet, it is enough to compare their dominant terms and choose the most dominant of them. The example also shows why  $O(h^4) - O(h^4)$  does not give zero: Each  $O$ -form represents certain expression that is dominated by some multiple of  $h^4$  and there is no guarantee that the number of  $h^4$  in these two expressions match. With the absence of more detailed information we have to accept the worst case scenario that there will be some  $h^4$  left (which is also the most likely case).

Now look at this:

$$\begin{aligned} f \cdot h &= [3h^4 + 5h^6 - 4h^7] \cdot [h^5 + 3h^6 - 2h^7] \\ &= 3h^9 + 9h^{10} - h^{11} + 11h^{12} - 22h^{13} + 8h^{14} = O(h^9) \text{ as } h \rightarrow 0. \end{aligned}$$

Or,

$$f \cdot h = O(h^4) \cdot O(h^5) = O(h^9) \text{ as } h \rightarrow 0.$$

If we only care about estimates, we can simplify our life by calculating with  $O$ -forms instead of actual expressions. It is not difficult, the above examples illustrate key formulas. Before we show them officially, we look at one more example. We will make a small change in  $g$ .

$$f(h) = 3h^4 + 5h^6 - 4h^7$$

$$g(h) = 3h^4 + 2h^5 - 4h^6$$

$$h(h) = h^5 + 3h^6 - 2h^7$$

The orders of growth at 0 remain the same, therefore we still have

$$f - g + h = O(h^4) - O(h^4) + O(h^5) = O(h^4) \text{ as } h \rightarrow 0.$$

This answer (upper estimate by  $h^4$ ) is true, but it is no longer the optimal answer. Indeed,

$$\begin{aligned} f - g + h &= [3h^4 + 5h^6 - 4h^7] - [3h^4 + 2h^5 - 4h^6] + [h^5 + 3h^6 - 2h^7] \\ &= -h^5 + 12h^6 - 6h^7 = O(h^5) \text{ as } h \rightarrow 0. \end{aligned}$$

We see that this error can indeed be estimated from above by  $h^4$ , but there is a better estimate available. We can find it using the  $O$ -notation if we decide to keep more information:

$$f - g + h = [3h^4 + O(h^5)] - [3h^4 + O(h^5)] + [h^5 + O(h^6)] = h^5 + O(h^5) - O(h^5) + O(h^6) = O(h^5) \text{ as } h \rightarrow 0.$$

Our calculations now confirm that  $h^5$  is indeed the dominant term in the expression and the estimate  $O(h^5)$  is optimal.

Now imagine that even the power  $h^5$  in our calculation disappeared. Then we would not know whether  $O(h^5)$  is optimal or not. If we cared, we would repeat the calculation, this time keeping track of all powers up to  $h^6$  and hiding only the higher ones in  $O$ -forms. There is no rule that would advise us which powers to keep in our calculations. Usually we try as little as possible (the smallest one), and if we find out that it was not good enough, we try to keep more. Experience helps a lot.

△

Now we show the rules for calculating with  $O$ -forms.

**Fact 38a.4.**

- (i) For  $a > 0$  and  $\alpha, \beta \in \mathbb{R}$ :  $\alpha O(h^a) \pm \beta O(h^a) = O(h^a)$  as  $h \rightarrow 0$ .
- (ii) For  $b \geq a \geq 0$  and  $\alpha, \beta \in \mathbb{R}$ :  $\alpha O(h^a) \pm \beta O(h^b) = O(h^a)$  as  $h \rightarrow 0$ .
- (iii) For  $a, b \geq 0$ :  $O(h^a) \cdot O(h^b) = O(h^{a+b})$  as  $h \rightarrow 0$
- (iv) For  $b \geq a \geq 0$ :  $\frac{1}{h^a} O(h^b) = O(h^{b-a})$  as  $h \rightarrow 0$ .

They are not really difficult. If you can't recall a suitable rule, just imagine some particular expression in place of an  $O$ -form and common sense should lead to an answer. For instance, if you are not sure about  $h^2 \cdot O(h^4)$ , just imagine

$$h^2 \cdot [ah^4 + bh^5 + ch^6 + \dots] = ah^6 + bh^6 + ch^6 + \dots = O(h^6) \text{ as } h \rightarrow 0.$$

In numerical mathematics we traditionally only consider  $h > 0$ . Then we will talk about asymptotic growth for  $h \rightarrow 0^+$ , that is, as  $h$  tends to zero from the right. As usual, one-sided considerations work the same as in the case of  $h \rightarrow 0$ .

## 38b. Asymptotic equality

In this section we look at a more precise way to capture behaviour of functions at infinity.

**Definition 38b.1.**

Let  $f, g$  be functions defined on some neighborhood of infinity. We say that  $f, g$  are **asymptotically equal**, denoted  $f \sim g$ , if

$$\lim_{x \rightarrow \infty} \left( \frac{f(x)}{g(x)} \right) = 1.$$

We use this notion in situations when we have a formula and we want offer a simpler alternative expressions that can replace it for large values of  $x$ .

**Example 38b.a:** We claim that  $(2x - 1)^2 + \sqrt{x} \sim 4x^2$ . Indeed,

$$\lim_{x \rightarrow \infty} \left( \frac{(2x - 1)^2 + \sqrt{x}}{4x^2} \right) = \lim_{x \rightarrow \infty} \left( \left(1 - \frac{1}{2x}\right)^2 + \frac{1}{4x\sqrt{x}} \right) = 1.$$

My calculator cannot tell the difference between these two expressions for  $x$  starting from  $10^{12}$ . Maple in its default configuration thinks these expressions are the same for  $x = 10^{10}$ .

△

We obtain the simpler formula in the good old way by ignoring less important terms. We know the hierarchy from calculus: Exponentials beat powers, these in turn beat logarithms, and among powers higher exponents rule. Before we start exploring this notion, let's ask a natural question: Is there any connection to the previous section on asymptotic growth?

**Fact 38b.2.**

Let  $f, g$  be functions defined on some neighborhood of infinity. The following are equivalent:

- (i)  $f \sim g$ ;
- (ii)  $f - g = o(f)$  as  $x \rightarrow \infty$ ;
- (iii)  $f - g = o(g)$  as  $x \rightarrow \infty$ .

Proofs are straightforward, for instance the implication (i)  $\implies$  (ii) requires proving that

$$\lim_{x \rightarrow \infty} \left( \frac{f(x) - g(x)}{f(x)} \right) = \lim_{x \rightarrow \infty} \left( 1 - \frac{g(x)}{f(x)} \right) = 1 - 1 = 0.$$

In some fields people also use the notation  $f = \Theta(g)$ . If you happen to know it then you most likely also see that  $f \sim g$  implies  $f = \Theta(g)$ .

The Fact shows that the difference between  $f$  and  $g$  is negligible if  $f \sim g$ , but only compared to these functions, it does not mean that it goes to zero. For instance, in example 38b.a the difference between the two expressions is  $-4x + 1 + \sqrt{x}$ , this definitely does not go to zero at infinity.

We can interpret the relation  $f \sim g$  as follows. We have a quantity  $f(x)$  and we approximate it using quantity  $g(x)$ . We have just seen that the absolute error need not disappear as  $x \rightarrow \infty$ . However, the relative error does tend to zero. In fact, this is just another reading of part (ii) in the above fact.

On the other hand, knowing that  $f - g \rightarrow 0$  at infinity does not guarantee that  $f \sim g$ , here the trouble comes when  $f$  and  $g$  get close to zero. If we prevent it, we do have some information.

**Fact 38b.3.**

Let  $f, g$  be functions defined on some neighborhood of infinity. Assume that there is some  $A > 0$  so that  $|f| \geq A$  and  $|g| \geq A$  on that neighborhood.

If  $f - g \rightarrow 0$  at infinity, then  $f \sim g$ .

**Proof:** We will use the fact from calculus that if  $F \rightarrow 0$  as  $x \rightarrow a$  and  $G$  is bounded on a neighborhood of  $a$ , then  $FG \rightarrow 0$  as  $x \rightarrow a$ . In our case  $\left| \frac{1}{g(x)} \right| \leq \frac{1}{A}$ . Thus

$$\begin{aligned} \lim_{x \rightarrow \infty} \left( \frac{f(x)}{g(x)} \right) &= \lim_{x \rightarrow \infty} \left( \frac{f(x)}{g(x)} - 1 \right) + 1 = \lim_{x \rightarrow \infty} \left( (f(x) - g(x)) \frac{1}{g(x)} \right) + 1 \\ &= 0 + 1 = 1, \end{aligned}$$

just like we needed. □

What can we say about properties of the relation  $\sim$ ? As a relation it is an equivalence.

**Fact 38b.4.**

Let  $f, g, h$  be functions defined on some neighborhood of infinity. The following are true.

- (i)  $f \sim f$ .
- (ii) If  $f \sim g$ , then  $g \sim f$ .
- (iii) If  $f \sim g$  and  $g \sim h$ , then  $f \sim h$ .

We need not worry about dividing by zero. Since  $\frac{f}{g} \rightarrow 1$ , there must be a neighborhood of infinity where the ratio is positive, so none of the functions can be zero there.

This notion also has some nice algebraic properties.

**Fact 38b.5.**

Let  $f, g$  be functions defined on some neighborhood of infinity such that  $f \sim g$ . Then the following are true.

- (i)  $f^k \sim g^k$  for any  $k \in \mathbb{R}$ .
- (ii)  $\ln(f) \sim \ln(g)$  if  $f, g$  have a positive limit (including infinity) not equal to 1 at infinity.
- (iii) If also  $u \sim v$  for some functions  $u, v$ , then  $f \cdot u \sim g \cdot v$  and  $\frac{f}{u} \sim \frac{g}{v}$  if the ratios make sense on some neighborhood of infinity.

**Proof:** Most proofs are routine.

(ii): Since both functions have positive limits, they must be positive on some neighborhood of infinity and thus the logarithms make sense. By our assumption  $\frac{f}{g} \rightarrow 1$ , hence  $\ln\left(\frac{f}{g}\right) \rightarrow \ln(1)$ , that is,  $\ln(f) - \ln(g) \rightarrow 0$ . The additional assumption also stipulates existence of some limit  $L > 0$  of function  $g$  at infinity, where  $L \neq 1$ . We are ready.

$$\begin{aligned} \lim_{x \rightarrow \infty} \left( \frac{\ln(f)}{\ln(g)} \right) &= \lim_{x \rightarrow \infty} \left( \frac{\ln(f)}{\ln(g)} - 1 \right) + 1 = \lim_{x \rightarrow \infty} \left( \frac{\ln(f) - \ln(g)}{\ln(g)} \right) + 1 \\ &= \frac{\lim_{x \rightarrow \infty} (\ln(f) - \ln(g))}{\lim_{x \rightarrow \infty} (\ln(g))} + 1 = \frac{0}{\ln(L)} + 1 = 0 + 1 = 1, \end{aligned}$$

since  $\ln(L) \neq 0$ . □

Note that we definitely need some condition in (ii) to prevent trouble in that ratio of limits. This is not just a question of proof that could be perhaps improved. Consider functions  $f(x) = 1 + \frac{1}{x^2}$  and  $g(x) = 1 + \frac{1}{x}$  that satisfy  $f \sim g$ . However, using l'Hospital's rule we obtain

$$\begin{aligned} \lim_{x \rightarrow \infty} \left( \frac{\ln(f(x))}{\ln(g(x))} \right) &= \lim_{x \rightarrow \infty} \left( \frac{g(x)}{f(x)} \frac{f'(x)}{g'(x)} \right) \\ &= \lim_{x \rightarrow \infty} \left( \frac{g(x)}{f(x)} \right) \cdot \lim_{x \rightarrow \infty} \left( \frac{-2x^{-3}}{-x^{-2}} \right) = 1 \cdot \lim_{x \rightarrow \infty} \left( \frac{2}{x} \right) = 0. \end{aligned}$$

There might be other conditions that would guarantee validity of (ii), but the situation is not simple and the condition we used works in most practical problems.

It is interesting to note what does not work. Concerning (iii), there are two popular operations missing and for a good reason. When we add functions  $f + u$ , it may happen that their dominant terms cancel, the same then happens with  $g + v$  and suddenly we are comparing terms over which we have no control. For a simple example take  $f(x) = x^2 + x$ ,  $g(x) = x^2 + 1$ , and  $u(x) = v(x) = -x^2$ .

It is also not true that  $f \sim g$  implies  $e^f \sim e^g$ , the functions  $f, g$  from the previous paragraph serve as a counter-example. It is also not true that from  $f \sim g$  one would get  $f' \sim g'$ . Graphically, the function  $f$  may have similar values as  $g$ , but it may oscillate about it with smaller and smaller

amplitudes. This inspires the counterexample:  $f(x) = x$  and  $g(x) = x + \sin(x)$  satisfy  $f \sim g$ , but  $f'(x) = 1$  is not asymptotically equal to  $g'(x) = \cos(x)$ .

The notion of asymptotic equality has other uses than just providing reasonably nice and reasonably precise approximations. There is one application known by all who passed their calculus course. It is the Limit comparison theorem for integrals: If  $f \sim g$ , then  $\int_{\infty}^{\infty} f(x) dx$  converges if and only if  $\int_{\infty}^{\infty} g(x) dx$  converges.

### 39. Appendix B: Proof of Picard's theorem

In this chapter we will prove the Picard theorem. It is interesting, because it uses notions and ideas that we used in an unrelated chapter.

We start by setting up the situation.

We are given an ODE of the form  $y' = f(x, y)$  and an initial condition  $y(x_0) = y_0$ . We want to prove that there exists some solution  $u(x)$  of this problem. The Picard theorem guarantees existence of this solution only on some neighborhood of  $x_0$ , so it is a local result, but it also guarantees its uniqueness on this neighborhood.

In order to reach this conclusion we need help from some assumptions. To this end we assume that the initial data  $(x_0, y_0)$  are inside some open rectangle  $I \times J$  on which the given equations is well behaved, namely  $f$  is continuous there and it is also Lipschitz in the second variable. In order to use these assumptions we will need some quantitative information.

1. We know that continuous functions are bounded on bounded finite sets. Thus we may assume (perhaps by making  $I \times J$  a bit smaller) that there is some  $M > 0$  so that  $|f(x, y)| \leq M$  on  $I \times J$ .

2. The Lipschitz assumption means that there is some  $K > 0$  so that

$$|f(x, y_1) - f(x, y_2)| \leq K \cdot |y_2 - y_1|$$

whenever  $x, y_1, y_2 \in I \times J$ .

So this is our basic setup. To show that there is some local solution we will use the fixed point approach. We used it before to solve problems of the form  $\varphi(x) = x$ . Solutions are called fixed points and sometimes we were able to obtain them using iteration  $x_{n+1} = \varphi(x_n)$ .

A key to success was the Banach fixed point theorem: If  $\varphi: \mathcal{M} \mapsto \mathcal{M}$  is a contraction, that for any initial choice  $x_0 \in \mathcal{M}$  the iteration  $x_{n+1} = \varphi(x_n)$  converges to some fixed point  $x_f$  and this point is unique in  $\mathcal{M}$ .

We can easily rewrite the given equation into a fixed-point form:

$$y = f(x, y) - y' + y.$$

This suggests that we look at the mapping  $\varphi(y) = f(x, y) - y' + y$ . If we find some fixed point of this mapping, we found a solution of the given ODE. However, this brings us to the first complication: The given initial value problem also requires that  $y(x_0) = y_0$  and quite obviously, our mapping is not able to guarantee this.

There is not good way to fix this problem, but it is useful to follow one blind alley for a while. Let's go.

Our setup would also have to include some set  $\mathcal{M}$  on which  $\varphi$  would act. An interesting idea is to incorporate the given initial condition into the definition of  $\mathcal{M}$ . For instance, we could try to define  $\mathcal{M}$  as the set of all functions  $g$  that are defined and differentiable on some neighborhood of  $x_0$  and satisfy  $g(x_0) = y_0$ .

This set is obviously non-empty, because the constant function  $g(x) = y_0$  belongs there. If we find a solution to our ODE in this set, it will automatically satisfy the initial condition and we are done. However, this set does not work well with our setup, because we cannot guarantee that  $\varphi$  goes from  $\mathcal{M}$  to  $\mathcal{M}$ . There are two fatal failures.

First, if we take some  $g$  from  $\mathcal{M}$ , then it is differentiable, but its derivative need not be, so  $\varphi(g) = f(x, g(x)) - g'(x) + g(x)$  need not be differentiable either. We could try to be more restrictive and require that functions in  $\mathcal{M}$  have to have derivatives of all orders, but this could easily rule out the solution we are looking for.

Even if we fixed this, there is another problem. When we take  $g \in \mathcal{M}$  then there is no guarantee that  $\varphi(g)(x_0) = y_0$  and there is no way to fix this.

**Example 39.a:** Consider the initial value problem  $y' = xy + x^2$ ,  $y(0) = 13$ .

Using the above approach we would rewrite it as  $y = xy + x^2 - y' + y$ . Thus we are interested in the mapping  $\varphi$  that maps a function  $g$  into the function  $\varphi(g): x \mapsto xg(x) + x^2 - g'(x) + g(x)$ . We also focus on the set  $\mathcal{M}$  of all differentiable functions on neighborhoods of  $x_0 = 0$  that satisfy  $g(0) = 13$ .

Obviously,  $g(x) = e^{5x} + 12$  is one such function. Then

$$\varphi(g) = x(e^{5x} + 12) + x^2 - 5e^{5x} + e^{5x} + 12 = (x - 4)e^{5x} + 12x + x^2 + 12.$$

This function is differentiable on a neighborhood of 0, but  $\varphi(g)(0) = 8$ , so  $\varphi(g) \notin \mathcal{M}$ .

△

So this was a total failure, but in fact the basic approach is sound. We just have to change the given equation into a fixed-point problem in a different way. Hopefully, the reader also got used a bit to the idea that we can treat functions as algebraic objects, collect them in sets, and use them as arguments (and images) for mappings.

So let's start with the given equation again. Since the name of the variable in the equation does not matter, we can write it as  $y'(t) = f(t, y(t))$ . Now we integrate on both sides over an interval between  $x_0$  and  $x$ , where  $x$  can be also smaller than  $x_0$ :

$$\int_{x_0}^x y'(t) dt = \int_{x_0}^x f(t, y(t)) dt.$$

On the left we can use the fundamental theorem of calculus, obtaining  $y(x) - y(x_0)$ , which allows us to build the initial condition into our equation:

$$y(x) - y_0 = \int_{x_0}^x f(t, y(t)) dt \iff y(x) = y_0 + \int_{x_0}^x f(t, y(t)) dt$$

We transformed the given differential equation into an integral equation that also includes the initial condition, that is, this is an equivalent form of the given initial value problem. Indeed, our deduction process shows that any solution  $y(x)$  of the given IVP must also solve this new equation.

Conversely, if some function  $y(x)$  solves the new integral equation, then by differentiating on both sides we immediately see that it must also solve the given differential equation, and moreover,

$$y(x_0) = y_0 + \int_{x_0}^{x_0} f(t, y(t)) dt = y_0 + 0 = y_0.$$

We thus have the first simplification, we only have to worry about one equation. We also notice that this new version already has the form of a fixed-point problem, because we have the function  $y(x)$  isolated on the left and some expression that involves  $y$  also on the right. Thus we are inspired to consider the following mapping that acts on functions  $g$ :

$$\varphi(g) : x \mapsto y_0 + \int_{x_0}^x f(t, y(t)) dt$$

**Example 39.b:** Let's revisit our previous example. Integrating and incorporating the initial condition we arrive at the form

$$y(x) = 13 + \int_0^x ty(t) + t^2 dt.$$

The resulting mapping  $\varphi$  acts on functions  $g$  according to formula

$$\varphi(g) : x \mapsto 13 + \int_0^x tg(t) + t^2 dt.$$

For instance, if  $g(x) = x^2$ , then

$$\varphi(g)(x) = 13 + \int_0^x t \cdot t^2 + t^2 dt = 13 + \int_0^x t^3 + t^2 dt = 13 + \left[ \frac{1}{4}t^4 + \frac{1}{3}t^3 \right]_{t=0}^{t=x} = 13 + \frac{1}{4}x^4 + \frac{1}{3}x^3.$$

So the mapping  $\varphi$  sends  $x^2$  to  $13 + \frac{1}{4}x^4 + \frac{1}{3}x^3$ .

If we try  $g(x) = e^{x^2}$ , we obtain

$$\varphi(g)(x) = 13 + \int_0^x t e^{t^2} + t^2 dt = 13 + \frac{1}{2}e^{x^2} + \frac{1}{3}x^3.$$

As we observed earlier, the image  $\varphi(g)$  always automatically satisfies the given initial condition, even if  $g$  itself does not.

△

Let's review our situation. We arrived at a mapping  $\varphi$  with the property that every fixed point of it solves the given initial value problem and vice versa. Now we just need to show that such a fixed point exists. To show that this idea has a chance to work we look at another example.

**Example 39.c:** Consider the initial value problem  $y' = y$ ,  $y(0) = 1$ .

This is an exponential differential equation, so we easily guess the right answer (or solve by separation), but let's pretend that we do not know.

The integral form is  $y(x) = 1 + \int_0^x y(t) dt$ , and we consider the mapping  $\varphi$  given by the formula

$$\varphi(x)(x) = 1 + \int_0^x g(t) dt.$$

We were finding fixed points by iteration, let's give it a try. We need an initial function, and it makes sense to start with something very simple, but it would be nice if it also satisfies the initial condition. One obvious candidate is the constant function  $y_0(x) = 1$ .

Then

$$y_1(x) = \varphi(y_0)(x) = 1 + \int_0^x 1 dt = 1 + x,$$

$$y_2(x) = \varphi(y_1)(x) = 1 + \int_0^x 1 + t dt = 1 + x + \frac{1}{2}x^2,$$

$$y_3(x) = \varphi(y_2)(x) = 1 + \int_0^x 1 + t + \frac{1}{2}t^2 dt = 1 + x + \frac{1}{2}x^2 + \frac{1}{6}x^3,$$

$$y_4(x) = \varphi(y_3)(x) = 1 + \int_0^x 1 + t + \frac{1}{2}t^2 + \frac{1}{6}t^3 dt = 1 + x + \frac{1}{2}x^2 + \frac{1}{3!}x^3 + \frac{1}{4!}x^4.$$

It is easy to show by induction that  $y_n(x) = \sum_{k=0}^n \frac{x^k}{k!}$ . These functions converge (in what sense? good question) to a well-known function  $\sum_{k=0}^{\infty} \frac{x^k}{k!} = e^x$ , which, indeed, is the solution to the given

initial value problem.

We see that the iteration  $y_{n+1} = \varphi(y_n)$  lead to a fixed point  $y_f(x) = e^x$ .

△

This example nicely illustrates that the notion of a fixed point and iteration also work in other, more general settings, which is exactly what we need. In particular, we will appeal to the Banach contraction theorem.

In order to do so, we have to determine the set on which we work, and show that our mapping is a contraction.

**Set:** This is a technical moment. We will consider continuous functions that satisfy the initial condition, but we will only take those that exists in a certain rectangle centered on  $(x_0, y_0)$ . Precisely, we take certain  $\varepsilon > 0$  and  $\delta > 0$  and consider the set

$$\mathcal{F} = \{g : U_\varepsilon(x_0) \mapsto U_\delta(y_0) : g \text{ continuous on } U_\varepsilon(x_0) \text{ and } g(x_0) = y_0\}.$$

This set is obviously non-empty, because it includes the constant function  $g(x) = y_0$ . Where do we get  $\varepsilon$  and  $\delta$ ? First, we fix them so that this rectangle allows us to use our main assumptions about  $I \times J$ . Thus for  $\delta$  we have one and only requirement:  $U_\delta(y_0) \subseteq J$ . Since  $(x_0, y_0)$  was taken from the interior of  $I \times J$ , we can surely find such  $\delta > 0$ .

For  $\varepsilon$  we need more. The first requirement is similar, we need to make  $\varepsilon$  small enough so that  $U_\varepsilon(x_0) \subseteq I$ . But we also need to add two more conditions:  $\varepsilon < \frac{1}{M}$  and  $\varepsilon < \frac{\delta}{L}$ , where the constants  $M, K$  come from our setup above (from the assumptions of the Picard theorem) and  $\delta$  was already fixed in the previous paragraph. Note that all these three requirements have the form of an upper bound, so we can definitely find  $\varepsilon > 0$  so small that it satisfies all of them.

We will see why we need these two conditions later, for now we know that we restricted ourselves to a certain rectangle. We are interested only in functions that exist in it and pass through its center.

We conclude this part by proving that our mapping  $\varphi$  actually goes from  $\mathcal{F}$  to  $\mathcal{F}$ . To this end, take some  $g \in \mathcal{F}$ .

First, since  $g$  is continuous on  $U_\varepsilon(x_0)$  and has values in  $J$  where  $f$  is continuous with respect to its second variable, then also  $f(x, g(c))$  is continuous on  $U_\varepsilon(x_0)$ , therefore it makes sense to integrate between  $x_0$  and  $x$  for any  $x \in U_\varepsilon(x_0)$ . This means that  $\varphi(g)$  actually exists and is defined on  $U_\varepsilon(x_0)$  as well.

Moreover, by the Fundamental Theorem of Calculus, the resulting function is continuous, and as we observed before, satisfies  $\varphi(g)(x_0) = y_0$ . Thus almost all conditions from the definition of  $\mathcal{F}$  are true, it remains to show that  $\varphi(g)$  has values in  $U_\delta(y_0)$ . To this end, we take any  $x \in U_\varepsilon(x_0)$  and estimate:

$$|\varphi(g)(x) - y_0| = \left| y_0 + \int_{x_0}^x f(t, g(t)) dt - y_0 \right| = \left| \int_{x_0}^x f(t, g(t)) dt \right|.$$

We observed above that one of the assumptions of the Picard theorem gives us boundedness by  $M$  on  $I \times J$ . We integrate over some subinterval of  $U_\varepsilon(x_0) \subseteq I$ , so  $t$  come from  $I$  as needed, and we took  $g$  from  $\mathcal{F}$ , which means that its values  $g(t)$  are in  $U_\delta(y_0) \subseteq J$ . Thus we have  $|f(t, g(t))| \leq M$ , therefore we can estimate

$$|\varphi(g)(x) - y_0| \leq M \cdot |x - x_0| < M \cdot \varepsilon.$$

Now we see where one of the requirements on  $\varepsilon$  came from, we needed  $\varepsilon$  so small that  $M\varepsilon < \delta$ . By this assumption,  $|\varphi(g)(x) - y_0| < \delta$ , so values of  $\varphi(g)$  are indeed in  $U_\delta(y_0)$ .

We proved that  $\varphi$  goes from  $\mathcal{F}$  to  $\mathcal{F}$  and it makes sense to look for a fixed point of this mapping in  $\mathcal{F}$ .

**Contraction:** In order to have a contraction we first need some way to measure distances between objects. The Banach contraction theorem works in general for metric spaces, this shows

that we would have to cover some extra theory here. Let's just play it by the ear.

There are several popular ways to measure distance between functions and we will go with the simplest. Given two functions  $g, h$ , the question "How far are they" has a natural answer, we look at their graphs and see where the graphs are most apart. We apply it to our setting, for  $g, h \in \mathcal{F}$  we define

$$\|g - h\|_\infty = \max\{|g(x) - h(x)| : x \in U_\varepsilon(x_0)\}.$$

As the notation suggests, this actually comes from a notion of a norm. Indeed, the definition  $\|g\|_\infty = \max |g(x)|$  defines a norm in many linear spaces of functions. However, we are not concerned with this now, because our set  $\mathcal{F}$  is not a linear space.

One can prove that our notion of distance satisfies all conditions that are needed for a metric space, so the Banach contraction theorem can be used. Thus our last step is to show that  $\varphi$  is a contraction on  $\mathcal{F}$  with respect to our distance.