Obyčejné diferenciální rovnice numericky

Je dána rovnice $y'(x) = F(x, y(x))$ s počáteční podmínkou $y(x_0) = y_0$ (dohromady se tomu říká úloha, přesněji počáteční úloha) pro interval $I = \langle x_0, x_0 + T\rangle$. Snažíme se najít přibližné řešení. Interval I rozdělíme na n stejných částí s uzlovými body $x_0, x_1, \dots, x_n$, kde $x_i = x_0 + ih$ pro $h = \frac{T}{n} > 0$. Funkci chceme získat jako tabulku hodnot y_i v bodech x_i, přičemž samozřejmě chceme, aby pro skutečné řešení $y(x)$ byly odchylky $y(x_i) - y_i$ co nejmenší.

Poznamenejme, že ačkoliv je číslo h odvozené, tak se o něm mluví častěji než o n, říká se mu **krok metody**. Je na něm totiž vidět, jak daleko jsou uzlové body od sebe, což samozřejmě výrazně ovlivní přesnost výpočtu, zatímco z čísla n to (bez znalosti délky intervalu T) přímo nevidíme. Typicky se očekává, že s menším krokem metody dostáváme lepší výsledky, někdy se jde limitně $h \to 0$. Při všech těchto úvahách se ovšem bere jako samozřejmost (i když se to neříká), že se omezujeme čistě na ta h, která lze získat ve tvaru $\frac{T}{n}$.

Jak se vlastně pozná kvalita metody? Ukážeme si nejčastější dvě kritéria.

• Metoda se nazývá **konvergentní**, jestliže se při zvyšování n blížíme s body y_i ke skutečnému řešení. Matematicky přesně:

Uvažujme určitou úlohu $y' = F(x, y(x))$, $y(x_0) = y_0$, nechť $y(x)$ je její přesné řešení. Pro každé n najdeme pomocí zkoumané metody řešení $\{y_i\}$ a jeho maximální chybu $E_n = \max\limits_i |y(x_i) - y_i|$. Metoda je konvergentní, pokud pro všechny úlohy s Lipschitzovskou F platí $E_n \to 0$.

Všechny metody, které zde probereme, jsou konvergentní.

• Metoda je **stabilní**, pokud chyby, které se při výpočtech objeví, neposiluje. Chyby se objeví jistě, už proto, že nehledáme řešení přesné, ale přibližné, dalším zdrojem jsou chyby zaokrouhlovací související s floating point algebrou, a metody často využívají již provedených výpočtů k dalším krokům, takže je velký potenciál na to, aby i malá chyba na začátku běhu metody způsobila výraznou odchylku vypočteného řešení od toho správného. Metody jsou k tomuto různě náchylné.

Existují různé způsoby, jak toto měřit, jedna rozšířená se dívá na reakci metody na konkrétní úlohu. Vezme se úloha, která má omezené řešení v tom smyslu, že existuje $M > 0$ takové, že když pro libovolné T najdeme řešení y na $\langle x_0, x_0 + T\rangle$, pak $|y(x)| \leq M$ pro $x \in \langle x_0, x_0 + T\rangle$. Na tuto úlohu aplikujeme hodnocenou metodu, čímž vznikne konkrétní výpočetní schéma. Toto schéma se nazve stabilní, pokud pro všechna n (neboli všechna h) jsou i obdržená řešení omezená (ve stejném smyslu, univerzální omezující konstanta pro všechna T).

Stabilita je dosti vzácná. Často dostáváme podmíněnou stabilitu, kdy metoda dává omezená řešení jen pro určité hodnoty kroku h. Stabilita se dá také kvalitativně porovnávat, do tohoto pokročilého tématu zde nepůjdeme. Důležité pro nás je, že je známo, které metody jsou stabilnější (ve smyslu že pro hodně úloh vedou na stabilní schéma) a které méně.

Zde si představíme nejzákladnější iterační metody (a na závěr mírně nakoukneme jinam). Princip iteračních metod je, že řešení $\{y_i\}$ konstruhují postupně, začnou s y_0 a postupně dopočítávají y_1, y_2 atd., přičemž se využívá předchozích spočítaných hodnot. V typickém případě existuje konstanta s taková, že se vždy používá právě s předchozích výsledků, tedy na spočítání y_{i+1} se použije $y_{i-s+1}, y_{y-s+2}, \dots, y_i$. V takovém případě mluvíme o metodách s-krokových.

Obecně jsou vícekrokové metody lepší, ale potřebují ke své iniciaci znát startovací hodnoty $y_0, \dots, y_{s-1}$, což bývá problém. Proto jsou velmi populární metody jednokrokové, kdy y_{i+1} počítáme čistě ze znalosti y_i, k nastartování pak stačí znát y_0, což je dáno jako počáteční podmínka pro úlohu. Někdy se používá kombinace, jednokrokovými metodami se odhadnou $y_0, \dots, y_{s-1}$ a dále se jede vícekrokovou metodou.

Pro úplnost dodejme, že existují i metody alternativní, jednu si nastíníme na závěr.

Je zjevné, že už první spočítaná hodnota y_1, která jistě není přesně rovna $y(x_1)$, vnáší do postupu chybu, a tento vlastně chybný údaj se pak stane východiskem pro odhad y_2, který už zase sám o sobě není přesný, a navíc ještě využívá ne zcela přesná vstupní data, a tak se to dál nabaluje, takže právě iterační metody jsou velice náchylné k nestabilitě.

• Uvažujeme úlohu a výpočetní schéma pro konkrétní n, čemuž odpovídá krok h, vzniknou tak body x_i. Zvolíme nějaké i z rozsahu $s, \dots, n$. Spočítáme si y_i, ale s použitím vstupních dat ze skutečného řešení y, tedy namísto $y_{i-s}, \dots, y_{i-1}$ použijeme $y(x_{i-s}), \dots, y(x_{i-1})$. Rozdíl $d_i(h) = y(x_i) - y_i$ pak představuje skutečnou chybu metody v i-tém kroku, bez vlivu předchozích odchylek.

Metoda je **konzistentní**, jestliže pro všechny řešené úlohy platí, že $\lim_{h \to 0^+} \left(\frac{d_i(h)}{h}\right) = 0$ pro všechna i. Dá se ukázat, že konvergentní metody jsou konsistentní, ale ne naopak.

Tento pojem umožňuje měřit kvalitu konvergence. Řekneme, že metoda **je řádu** p, popř. že má řád p, jestliže $d_i(h) = \Theta(h^{p+1})$ pro $h \to 0$.

Proč je změna o jedničku? Globální pohled na metodu získáme tak, že si odhadneme celkovou chybu po provedení většího počtu kroků. Protože největší možný počet kroků je $n = \frac{T}{h}$ a v jednom kroku uděláme chybu $\sim Ch^{p+1}$, vychází, že u metody řádu p je globální chyba přibližně h^p. Někdy se také hodí úvaha opačná, chceme-li globální chybu ε, tak na jeden krok připadá zhruba $\frac{\varepsilon}{n} = c \cdot h\varepsilon$.

1a. Eulerova metoda (Euler method)

Řešíme rovnici $y' = F(x, y)$. Chceme vymyslet indukční krok, který nás dovede z y_i do y_{i+1}. Nejjednodušší je si funkci na okolí x_i aproximovat pomocí tečny:

$$y(x) \sim y(x_i) + y'(x_i) \cdot (x - x_i) \sim y_i + F(x_i, y_i) \cdot (x - x_i).$$

Použijeme to v bodě $x = x_{i+1}$ a dostáváme přirozený odhad pro y_{i+1}.

Algoritmus (Eulerova metoda (Euler formula) pro počáteční úlohu $y' = F(x, y)$)**.**
Zadána rovnice $y' = F(x, y)$ na $\langle x_0, x_0 + T\rangle$, počáteční podmínka y_0 a $n \in I\!N$, odtud $h = \frac{T}{n}$.
0. y_0 je ze zadání.
1. Pro $i = 0, \ldots, n - 1$ definujeme $y_{i+1} = y_i + hF(x_i, y_i)$.
△

Věta.
Uvažujme úlohu $y' = F(x, y)$ na $I = \langle x_0, x_0 + T\rangle$ s počáteční podmínkou $y(0) = y_0$. Jestliže je F Lipschitzovská a úloha má řešení z $C^2(I)$, pak Eulerova metoda konverguje a je řádu 1.

Důkaz: Jaká je chyba aproximace v i-tém kroku? Namísto tečny použijeme Taylorův rozvoj s Lagrangeovým tvarem zbytku:

$$y(x_{i+1}) = y(x_i) + y'(x_i)h + \tfrac{1}{2}y''(\xi)h^2 = y_i + hF(x_i, y_i) + \tfrac{1}{2}y''(\xi)h^2 = y_{i+1} + \tfrac{1}{2}y''(\xi)h^2.$$

Vidíme, že $d_{i+1}(h) = \frac{1}{2}y''(\xi)h^2$, kde ξ je nějaké číslo z intervalu I. Protože je y'' spojitá, je na I omezena nějakou konstantou M_2, platí tedy $e_i = |d_i(h)| \le \frac{1}{2}M_2h^2$. Tím je potvrzen řád metody.

Důkaz konvergence je náročnější a vynecháme jej.

□

Implicitní Eulerova metoda.

Zajímavý nápad je dívat se zpět, tedy hledat y_{i+1} tak, aby to při zpětném pohledu dobře ladilo s rovnicí.

$$y_{i+1} = y_i + hF(x_{i+1}, y_{i+1}).$$

Toto je implicitní Eulerova metoda, která má rovněž řád jedna.

Ukáže se, že z hlediska rychlosti konvergence a lokální chyby je implicitní Eulerova metoda v zásadě stejná jako Eulerova, jenže na rozdíl od ní je implicitní, což je docela velký problém. V některých případech lze řešit $y_{i+1} = y_i + hF(x_{i+1}, y_{i+1})$ pro y_{i+1} analyticky, většinou nezbývá než tuto rovnici řešit numericky, například metodou iterační, kdy se často bere $y_{i+1}^{(0)} = e_i$ a $y_{i+1}^{(k+1)} = y_i + hF(x_{i+1}, y_{i+1}^{(k)})$.

Eulerovy metody se nedoporučují pro přesnější výpočty, protože pak by vyžadovaly velmi malá h, kde ale zase výsledek zašumí chyby výpočtu. Navíc mívá často problémy se stabilitou. Je dobrá na rychlé přiblížení, popřípadě jako startovací metoda pro vícekrokové metody. Není stabilní, nicméně je často podmíněně stabilní.

1b. Přístup přes integrály

Víme, že pro diferencovatelné funkce máme $y(b) = y(a) + \int_a^b y'(t)\,dt$. Pokud toto aplikujeme na jeden úsek v našem schématu pro řešení rovnice, dostáváme

$$y(x_{i+1}) = y(x_i) + \int_{x_i}^{x_{i+1}} y'(t)\,dt = y_i + \int_{x_i}^{x_i+h} F(t, y(t))\,dt.$$

Integrály jsme se již naučili odhadovat, dostaneme tak odhad y_{i+1}. Podle toho, jak kvalitní metody pro integrál použijeme, dostáváme různě kvalitní metody pro řešení diferenciálních rovnic. Pokud totiž přesnou hodnotu integrálu $I = \int_a^b y'(t)\,dt$ nahradíme odhadem I_n, pak bude lokální chyba kroku metody rovna

$$d_i(h) = y(x_{i+1}) - y_{i+1} = \Big[y(x_i) + \int_{x_i}^{x_{i+1}} y'(t)\,dt\Big] - [y(x_i) + I_n] = I - I_n.$$

Máme-li odhad chyby ve tvaru $c(b-a)h^p$, tak díky $b-a = h$ dostáváme $d_i(h) = ch^{p+1}$, takže řád odhadu chyby, který platí pro metody aproximace integrálu, nám přímo dá řád vzniklé metody pro řešení ODR.

Nápad 1. Použijeme metodu levých obdélníků: $y_{i+1} = y_i + hF(x_i, y_i)$. Toto je Eulerova metoda.

Nápad 2. Použijeme metodu pravých obdélníků: $y_{i+1} = y_i + hF(x_{i+1}, y_{i+1})$. Toto je implicitní Eulerova metoda.

Nápad 3. Použijeme metodu obdélníkovou používající hodnoty ve středu panelů (která je řádu dva): $y_{i+1} = y_i + hF(x_i + \frac{1}{2}h, y(x_i + \frac{1}{2}h))$. Dává to metodu druhého řádu, problém je, že $y(x_i + \frac{1}{2}h)$ nemáme v našem výpisu hodnot $\{y_i\}$. Trik: Odhadneme si to podobně jako u Eulerovy metody, tedy myslíme si, že $y(x_i + \frac{1}{2}h) \sim y_i + \frac{1}{2}hF(x_i, y_i)$. Dostáváme pak schéma

$$y_{i+1} = y_i + hF\big(x_i + \tfrac{1}{2}h, y_i + \tfrac{1}{2}hF(x_i, y_i)\big).$$

Uděláme si oficiální algoritmus, ve kterém si každý krok rozložíme do fází.

Algoritmus („modified Euler formula" či „improved polygon" pro počáteční úlohu $y' = F(x, y)$)**.**
Zadána rovnice $y' = F(x, y)$ na $\langle x_0, x_0 + T\rangle$, počáteční podmínka y_0 a $n \in I\!N$, odtud $h = \frac{T}{n}$.
0. y_0 je ze zadání.
1. Pro $i = 0, \dots, n-1$:
a) Odhadneme $y'(x_i)$: $k_1 = F(x_i, y_i)$, odhadneme $y(x_i + \frac{1}{2}h)$: $y^*_{i+1/2} = y_i + \frac{1}{2}k_1 h$.
b) Odhadneme $y'(x_i + \frac{1}{2}h)$: $k_2 = F(x_i + \frac{1}{2}h, y^*_{i+1/2})$.
c) Spočítáme $y_{i+1} = y_i + hk_2$.
$\triangle$

Tato metoda je druhého řádu a z našich „nápadů" má nejlepší stabilitu. Patří mezi populárnější metody.

Nápad 4. Použijeme lichoběžníkovou metodu: $y_{i+1} = y_i + \frac{1}{2}h[F(x_i, y_i) + F(x_i, y_{i+1})]$. Hodnotu y_{i+1} si nejprve odhadneme pomocí Eulera.
Pro $i = 0, \dots, n-1$ tedy spočítáme $y'(x_i)$ jako $k_1 = F(x_i, y_i)$ a odhadneme y_{i+1} Eulerovým vzorcem: $y^*_{i+1} = y_i + k_1 h$.
Pak znovu odhadneme $y_{i+1} = y_i + \frac{1}{2}h[F(x_i, y_i) + F(x_{i+1}, y^*_{i+1})]$.

Tato metoda patří do zajímavé rodinky metod zvaných „predictor-corrector", protože jsme nejprve odhadli y_{i+1} a pak tento odhad zpřesnili. Je 2. řádu a chová se také dobře ohledně stability.

Výraz $F(x_{i+1}, y^*_{i+1})$ vlastně odhaduje $y'(x_{i+1})$. Vzniká tím jakýsi kompromis mezi dopřednou a zpětnou Eulerovou metodou.

Algoritmus („Heun formula“ či „improved Euler formula“ pro počáteční úlohu $y' = F(x,y)$).
Zadána rovnice $y' = F(x,y)$ na $\langle x_0, x_0+T\rangle$, počáteční podmínka y_0 a $n \in \mathbb{N}$, odtud $h = \frac{T}{n}$.
0. y_0 je ze zadání.
1. Pro $i = 0, \dots, n-1$:
a) Odhadneme $y'(x_i)$: $k_1 = F(x_i, y_i)$.
b) Odhadneme y_{i+1}: $y^*_{i+1} = y_i + k_1 h$, pak odhadneme směrnici $y'(x_{i+1})$: $k_2 = F(x_{i+1}, y^*_{i+1})$.
c) Spočítáme $y_{i+1} = y_i + h \cdot \frac{1}{2}(k_1 + k_2)$.
$\triangle$

1c. Metody Rungeho a Kutta

Vzorec $y_{i+1} = y_i + h \cdot \frac{1}{2}(k_1 + k_2)$ lze přepsat jako $y_{i+1} = y_i + h \cdot k$, kde $k = \frac{1}{2}(k_1 + k_2)$ je pokus o odhadnutí správné směrnice tím, že si ji „osaháme“ v různých místech intervalu $\langle x_i, x_{i+1}\rangle$ a pak to nějak zprůměrujeme. Právě tato myšlenka spojuje kategorii zvanou Runge-Kuttovy metody, my se zde podíváme na metody explicitní, tedy proveditelné přímým výpočtem.

Myšlenka: Vytvoříme si N odhadů směrnice v různých místech intervalu $\langle x_i, x_{i+1}\rangle$.

$$\begin{aligned}
k_1 &= F(x_i, y_i)\\
k_2 &= F(x_i + c_2 h, y_i + a_{21}k_1)\\
k_3 &= F(x_i + c_3 h, y_i + a_{31}k_1 + a_{32}k_2)\\
&\ \vdots\\
k_N &= F(x_i + c_N h, y_i + a_{N1}k_1 + a_{N2}k_2 + \dots + a_{N,N-1}k_{N-1})
\end{aligned}$$

Připouštíme možnost, že se do téhož místa vrátíme vícekrát (predictor-corrector), abychom první nástřel hodnoty $y'(x)$ opravili. Výsledné odhady směrnic $k_1, \dots, k_N$ pak zprůměrujeme $k = \sum\limits_{j=1}^{N} w_j k_j$ a odhadujeme $y_{i+1} = y_i + h \cdot k$.

Často se pro názornost bod $x_i + c_j h$ značí jako x_{i+c_j}, třeba $x_{i+2/3} = x_i + \frac{2}{3}h$ (jsme ve dvou třetinách intervalu mezi x_i a x_{i+1}), odpovídající odhad pro y pak značíme $y^*_{i+2/3} = y_i + a_{j1}k_1 + \dots + a_{j,j-1}k_{j-1}$, popřípadě $y^{**}_{i+2/3}$, pokud se do $x_{i+2/3}$ vrátíme a odhad zpřesníme.

Schéma se dá zachytit pomocí konstant c_j, $a_{j,l}$ a w_j. Tradičně se používa tzv. Butcherovo schéma (Butcher tableau)

$$\begin{array}{c|ccccc}
0 & & & & & \\
c_2 & a_{21} & & & & \\
c_3 & a_{31} & a_{32} & & & \\
\vdots & \vdots & & \ddots & & \\
c_N & a_{N1} & a_{N2} & \cdots & a_{N,N-1} & \\
\hline
 & w_1 & w_2 & \cdots & w_{N-1} & w_N
\end{array}$$

Konstanty N (počet kroků), c_j (nody), a_{jl} (Runge-Kutta matice) a w_j (váhy) se volí tak, aby při hledání řádu metody zmizelo co nejvíce členů v Taylorově rozvoji pro lokální chybu. Další omezení vyplývají z požadavku, aby metoda správně fungovala. Protože průměrujeme, chceme $\sum\limits_{j=1}^{N} w_j = 1$. Dá se dokázat, že vzniklá metoda bude konsistentní, pokud pro $j = 2, \dots, N$ bude splněno $\sum\limits_{l=1}^{j-1} a_{j,l} = c_j$.

Vždy dokážeme schéma nastavit tak, že N dává řád metody. Když tedy mluvíme o RK-metodě řádu N, tak to má dva významy, jednak řád metody dle obecné definice výše, a druhak počet kroků vykonaných k odhadu k. U správně vytvořené metody tyto dva významy souhlasí.

RK metody prvního řádu jdou rovnou jedním krokem, dostáváme tak Eulerovy metody (klasickou i implicitní).

RK metody druhého řádu se nejčastěji používají dvě, viděli jsme je již v sekci 1b jako nápady 3 a 4. Metoda z nápadu 3 se obvykle značí zkratkou **RK2**, což naznačuje, že je z těch dvou populárnější. Maple ovšem jako `rk2` vidí nápad 4 neboli Heunův vzorec.

Čím vyšší řád chceme, tím více musíme v každém cyklu počítat. Velice populárním kompromisem je řád 4, jmenovitě jedno konkrétní schéma zvané **RK4**. Získá se úpravou výpočtu vycházejícího ze Simpsonovy metody.

Algoritmus (RK4 pro počáteční úlohu $y' = F(x, y)$).
Zadána rovnice $y' = F(x, y)$ na $\langle x_0, x_0 + T\rangle$, počáteční podmínka y_0 a $n \in I\!N$, odtud $h = \frac{T}{n}$.
0. y_0 je ze zadání.
1. Pro $i = 0, \dots, n-1$:
a) Odhadneme $y'(x_i)$: $k_1 = F(x_i, y_i)$.
b) Odhadneme $y(x_i + \frac{1}{2}h)$: $y^*_{i+1/2} = y_i + \frac{1}{2}k_1 h$, a pak směrnici $y'(x_i + \frac{1}{2}h)$: $k_2 = F(x_i + \frac{1}{2}h, y_{i+1/2}{}^*)$.
c) Znovu (a lépe?) odhadneme $y(x_i + \frac{1}{2}h)$: $y^{**}_{i+1/2} = y_i + \frac{1}{2}k_2 h$ a pak $y'(x_i + \frac{1}{2}h)$: $k_3 = F(x_i + \frac{1}{2}h, y^{**}_{i+1/2})$.
d) odhadneme $y(x_i + h)$: $y^*_{i+1} = y_i + k_3 h$ a směrnici $y'(x_{i+1})$: $k_4 = F(x_{i+1}, y^*_{i+1})$.
e) Spočítáme $y_{i+1} = y_i + \frac{1}{6}h[k_1 + 2k_2 + 2k_3 + k_4]$.
$\triangle$

Jak už bylo řečeno, je to metoda 4. řádu, s vynikajícími vlastnostmi ohledně stability. Je to asi nejčastěji používaná Runge-Kuttova metoda.

Poznámka:
Existují další typy Runge-Kuttových metod. Takzvané implicitní nemají matici A dolní trojúhelníkovou, ale čtvercovou, takže počítají $k_j = F(x_i + c_j h, y_i + a_{j1}k_1 + a_{j2}k_2 + \dots + a_{jN}k_N)$ pro všechna $j = 1, \dots, N$. Vznikají tak implicitní rovnice, které je třeba při každém kroku metody vyřešit. Praxe ukazuje, že obecně mají implicitní RK metody větší rozsah pro h, kdy jsou stabilní, takže se aplikují zejména u problematických rovnic. Dá se také říci, že pro daný krok a síť bývají přesnější, ale platíme za to nutností řešit ty implicitní rovnice, což se často dělá iterací pro hledání pevného bodu. Pak je zase třeba zajistit konvergenci, často tak dostáváme omezení na velmi malá h.
Adaptivní RK metody si zase hrají s délkou kroku. Snaží se jej průběžně měnit tak, aby nikde nebyl zbytečně malý, když je rovnice příznivá, a v problémových partiích se naopak síť zahustí. Jeden příklad si ukážeme níže.
$\triangle$

1d. Odhad chyby

Dokážeme poznat z běhu metody, jakou asi děláme chybu? Jedna možnost je použít (podobně jako u integrování) Richardsonovu extrapolaci, ta však vyžaduje přepočítávání pro dvě různé sítě. Podíváme se na ni níže. Jiná zajímavá alternativa je použít pro stejnou síť dvě různé metody. Porovnáním odhadneme lokální chybu jednoho kroku (viz určování řádu), pomocí ní pak dokážeme zhruba odhadnout, jak jsme na tom globálně.
Mějme tedy síť $x_0 < x_1 < \dots < x_N$ a dvě metody, první vytvoří odhady $\{y_i\}$ a druhá $\{z_i\}$. Zvolme nějaký bod x_i a podívejme se na lokální chybu té první.

$$d_{i+1}(h) = y(x_{i+1}) - y_{i+1} = y(x_{i+1}) - z_{i+1} + z_{i-1} - y_{i+1}.$$

Předpokládejme, že první metoda je řádu p a druhá řádu $q > p$. Pak je výraz nalevo řádu h^{p+1}, zatímco napravo je $y(x_{i+1}) - z_{i+1}$ řádu vyššího, tudíž jej lze zanedbat. Dostáváme závěr, že $z_{i-1} - y_{i+1}$ musí být řádu h^{p+1} a je to odhad lokální chyby d_i. Globální chyba je pak přibližně rovna $d_i \cdot n = \frac{1}{h}d_i$, můžeme ji tedy odhadnout výrazem $\frac{1}{h}\max|z_{i+1} - y_{i+1}|$.
Docela dost jsme toho zanedbávali, ale v praxi to funguje relativně uspokojivě. Ještě zajímavější je následující úvaha. V našich předchozích odhadech jsme jistě dělali chyby. Pokud by ovšem byly systematické, tak se dají obdobné chyby očekávat i při jiných krocích, lze tedy poznatky jednoho kroku vztáhout konzistentně na další. To znamená, že naše výsledky můžeme použít k predikci chyby při dalším běhu.
Předpokládejme tedy, že máme odhad globální chybu $E_h = \frac{1}{h}\max|z_{i+1} - y_{i+1}| \sim Kh^p$. Pokud tutéž metodu použijeme s krokem velikosti sh (kde $s > 0$), dostáváme chybu

$$E_{sh} \sim K(sh)^p = s^p K h^p \sim s^p \frac{1}{h}\max|z_{i+1} - y_{i+1}|.$$

Odtud dostáváme následující závěr.

Fakt.
Mějme síť $x_0 < x_1 < \cdots < x_N$ s krokem $h > 0$ a odhady řešení $\{y_i\}$ vytvořený hlavní metodou řádu p a $\{z_i\}$ vytvořený kontrolní metodou řádu vyššího než p. Pak jsou hodnoty $|z_{i+1} - y_{i+1}|$ rozumným odhadem lokální chyby hlavní metody.
Jestliže chceme odhad řešení s globální chybou $\varepsilon > 0$, pak je vhodné zkusit hlavní metodu s krokem sh, kde

$$s = \Big(\frac{h\varepsilon}{\max|z_{i+1} - y_{i+1}|}\Big)^{1/p}.$$

Typicky se používá kontrolní metoda (error estimator) řádu $p+1$. Lze tak například použít dopředného Eulera a jako kontrolu Heunův vzorec. Pro lepší výsledky je ale dobré začít metodou vyššího řádu. Pak se stává nevýhodou nutnost počítat pro každou z nich více hodnot k_j. Fehlberg přišel se zajímavým nápadem na RK metodu 4. řádu, která používá tytéž k_j jako jistá RK metoda řádu 5.

$$\begin{array}{c|cccccc}
0 & & & & & & \\
\frac{1}{4} & \frac{1}{4} & & & & & \\
\frac{3}{8} & \frac{3}{32} & \frac{9}{32} & & & & \\
\frac{12}{13} & \frac{1932}{2197} & -\frac{7200}{2197} & \frac{7296}{2197} & & & \\
1 & \frac{439}{216} & -8 & \frac{3680}{513} & -\frac{845}{4104} & & \\
\frac{1}{2} & -\frac{8}{27} & 2 & -\frac{3544}{2565} & \frac{1859}{4104} & -\frac{11}{40} & \\
\hline
 & \frac{25}{216} & 0 & \frac{1408}{2565} & \frac{2197}{4104} & -\frac{1}{5} & \\
 & \frac{16}{135} & 0 & \frac{6656}{12825} & \frac{28561}{56430} & -\frac{9}{50} & \frac{2}{55}
\end{array}$$

Protože výsledek ve Faktu je i lokální, je tato dvojice metod ideální jako nástroj k adaptivní metodě, která pružně přizpůsobuje velikost kroku.

Algoritmus (RKF45, adaptivní metoda Runge-Kutta-Fehlberga pro počáteční úlohu $y' = F(x,y)$)**.**
Zadána rovnice $y' = F(x,y)$ na $\langle x_0, x_0+T\rangle$, počáteční podmínka y_0, počáteční krok $h_0 > 0$ a požadovaná přesnost $\varepsilon > 0$.
0. y_0 je ze zadání.
1. Pro $i = 0, \dots, n-1$:
a) Spočítáme $k_1 = F(x_i, y_i)$.
b) Spočítáme $k_2 = F\big(x_i + \frac{1}{4}h_i, y_i + \frac{1}{4}k_1\big)$.
c) Spočítáme $k_3 = F\big(x_i + \frac{3}{8}h_i, y_i + \frac{3}{32}k_1 + \frac{9}{32}k_2\big)$.
d) Spočítáme $k_4 = F\big(x_i + \frac{12}{13}h_i, y_i + \frac{1932}{2197}k_1 - \frac{7200}{2197}k_2 + \frac{7296}{2197}k_3\big)$.
e) Spočítáme $k_5 = F\big(x_i + h_i, y_i + \frac{439}{216}k_1 - 8k_2 + \frac{3680}{513}k_3 - \frac{845}{4104}k_4\big)$.
f) Spočítáme $k_6 = F\big(x_i + \frac{1}{2}h_i, y_i - \frac{8}{27}k_1 + 2k_2 - \frac{3544}{2565}k_3 + \frac{1859}{4104}k_4 - \frac{11}{40}k_5\big)$.
g) Odhadneme $y_{i+1} = y_i + h_i\big(\frac{25}{216}k_1 + \frac{1408}{2565}k_3 + \frac{2197}{4104}k_4 - \frac{1}{5}k_5\big)$
a $z_{i+1} = y_i + h_i\big(\frac{16}{135}k_1 + \frac{6656}{12825}k_3 + \frac{28561}{56430}k_4 - \frac{9}{50}k_5 + \frac{2}{55}k_6\big)$.
Jestliže $\frac{1}{h_i}|z_{i+1} - y_{i+1}| > \varepsilon$, určíme $s = \big(\frac{h\varepsilon}{|z_{i+1}-y_{i+1}|}\big)^{1/4}$ a začneme opět od a) s krokem $h_i = sh_i$.
Jinak nastavíme $x_{i+1} = x_i + h_i$, $h_{i+1} = h_i$ a pokračujeme v cyklu, tedy zvýšíme i o jedničku atd.
$\triangle$

Na konci algoritmu je třeba se trefit do koncové hodnoty $x_0 + T$, v průběhu je při opravách h_i také obvykle dobré hlídat, aby se h nedostalo z rozumných mezí $\langle h_{\min}, h_{\max}\rangle$.
Tento algoritmus patří k nejpoužívanějším.

2. Richardsonova extrapolace

Pokud u metody (ne nutně jednokrokové, dokonce ani nemusí být iterační) známe odhad její chyby, můžeme se pokusit vykrátit první člen tohoto odhadu pomocí dvou běhů metody.
Mějme tedy metodu, o které je známo, že má chybu $d_i(h) \sim ch^{p+1} + O(h^{q+1})$, kde $q > p$.

Uvažujme dva její běhy, jeden s krokem h dává $\{y_{h,i}\}$, druhý s krokem $\frac{1}{2}h$ dává $\{y_{h/2,j}\}$. Je snadné si rozmyslet, že body $\{x_i\}$ jsou podmnožinou bodů $\{x_j\}$ a hodnota funkce y v bodě $x_{h,i} = x_{h/2,2i}$ je aproximována čísly $y_{h,i}$ a $y_{h/2,2i}$.

Podle našeho předpokladu (a s přimhouřením oka) dostáváme pro bod $x = x_{h,i} = x_{h/2,2i}$ odhady $y(x) - y_{h,i} = Ch^{p+1} + O(h^{q+1})$ a $y(x) - y_{h/2,2i} = C(h/2)^{p+1} + O((h/2)^{q+1})$, odtud pak úpravou a odečtením

$$\left.\begin{aligned} y(x) - y_{h,i} &= Ch^{p+1} + O(h^{q+1}) \\ 2^{p+1}y(x) - 2^{p+1}y_{h/2,2i} &= Ch^{p+1} + O(h^{q+1}) \end{aligned}\right\} \implies (2^{p+1}-1)y(x) - 2^{p+1}y_{h/2,2i} + y_{h,i} + O(h^{q+1})$$

$$\implies y(x) = \frac{2^{p+1}y_{h/2,2i} - y_{h,i}}{2^{p+1}-1} + O(h^{q+1}).$$

Zdá se tedy, že získáváme odhad pro $y(x)$ s chybou řádu q.

Ve skutečnosti jsme samozřejmě neměli ve výrazech přesnou rovnost, takže nelze říct, že by člen Ch^{p+1} úplně zmizel, ale v praxi Richardsonova extrapolace funguje dost dobře.

Podobně jako u integrálů, i zde lze odvodit i přibližný odhad chyby pro řešení $y_{h/2,2i}$,

$$y(x_{2i}) - y_{h/2,2i} \sim \frac{y_{h/2,2i} - y_{h,i}}{2^{p+1}-1}.$$

Toto lze použít při testování, zda jsme již spokojeni s výsledkem.

3. Soustavy rovnic, rovnice vyššího řádu

Uvažujme soustavu rovnic

$$\begin{aligned} y_1' &= F_1(x, y_1, y_2, \dots, y_m) \\ y_2' &= F_2(x, y_1, y_2, \dots, y_m) \\ &\vdots \\ y_m' &= F_m(x, y_1, y_2, \dots, y_m) \end{aligned}$$

na intervalu $\langle x_0, x_0+T\rangle$. Cílem je pro určitou síť bodů $x_0 < x_1 < \cdots < x_n = x_0+T$ vytvořit odhady pro jednotlivé funkce, tedy pro každou funkci y_j z přesného řešení najít konečnou posloupnost $\{(y_j)_i\}_{i=1}^n$, která bude y_j co nejblíže.

Metody, které jsme probrali výše, lze relativně snadno upravit na tuto situaci. Například Eulerova metoda by fungovala takto:

1. Hodnoty $(y_j)_0$ získáme z dané počáteční podmínky $y_1(x_0), \cdots, y_m(x_0)$.

2. Máme-li $(y_1)_i, \cdots, (y_m)_i$ pro jisté $i < n$, tak hodnoty v bodě x_{i+1} odhadneme vzorci

$$\begin{aligned} (y_1)_{i+1} &= (y_1)_i + F_1(x_i, (y_1)_i, (y_2)_i, \dots, (y_m)_i) \\ &\vdots \\ (y_m)_{i+1} &= (y_m)_i + F_m(x_i, (y_1)_i, (y_2)_i, \dots, (y_m)_i) \end{aligned}$$

Věci se chovají vcelku dle očekávání, například lokální chyba je $O(h^2)$, jde tedy o metodu řádu 1.

Nyní uvažujme rovnici $y^{(m)} = F(x, y, y', \dots, y^{(m-1)})$ řádu m. Ze znalosti m-té derivace se ke znalosti rychlosti růstu funkce přechází komplikovaněji, proto bývá jednodušší použít standardní převod rovnice na soustavu. Zavedeme funkce $y_1 = y$, $y_2 = y'$, $y_2 = y''$ atd., obecně $y_j = y^{(j-1)}$ pro $j = 1, \dots, m$, dostáváme pak soustavu

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= y_3 \\ &\vdots \\ y_{m-1}' &= y_m \\ y_m' &= F(x, y_1, y_2, \dots, y_{m-1}) \end{aligned}$$

Je zjevné, že každé řešení původní úlohy vede na řešení soustavy, naopak vyřešením soustavy dostáváme mimo jiné funkci y_1 splňující původní rovnici. Soustavu řešíme oblíbenou metodou.

4. Metoda konečných diferencí (finite difference method, FDM)

Pro případ jedné proměnné jsme již odvodili, jak odhadovat derivaci.

$$
\begin{aligned}
f'(a) &= \frac{f(a+h)-f(a)}{h} + O(h), && \text{dopředná diference}\\
f'(a) &= \frac{f(a)-f(a-h)}{h} + O(h), && \text{zpětná diference}\\
f'(a) &= \frac{f(a+h)-f(a-h)}{2h} + O(h^2), && \text{symetrická diference}\\
f''(a) &= \frac{f(a+h)+f(a-h)-2f(a)}{h^2} + O(h^2).
\end{aligned}
$$

Aplikovat tyto vzorce na funkci y danou hodnotami na síti je pak snadné, například

$$y'(x_i) \sim \frac{y_{i+1}-y_{i-1}}{2h}, \qquad y''(x_i) \sim \frac{y_{i+1}+y_{i-1}-2y_i}{h^2}.$$

Je-li dána rovnice $y' = F(x, y(x))$ a pravidelná síť $x_0 < x_1 < \cdots < x_n$ s krokem h, pak lze v konkrétním bodě x_i aplikovat na levou stranu odhad derivace.

Dopředná diference vede na rovnici $\frac{y_{i+1}-y_i}{h} = F(x_i, y_i)$, což po vyřešení pro y_{i+1} okamžitě vede na Eulerovu metodu. Zpětná diference pak vede na rovnici $\frac{y_i-y_{i-1}}{h} = F(x_i, y_i)$, po posunu indexu to dává $y_{i+1} - y_i = hF(x_{i+1}, y_{i+1})$ a čtenář v tom jistě poznává implicitní Eulerovu metodu.

Symetrická diference vede po dosazení na rovnici $y_{i+1} - y_{i-1} = 2hF(x_i, y_i)$ neboli

$$y_{i+1} = y_{i-1} + 2hF(x_i, y_i),$$

což je dvoukroková metoda 2. řádu. Těm jsme se tu nevěnovali, poznamenejme tedy že konverguje výrazně rychleji než předchozí dvě. Je explicitní, protože v okamžiku výpočtu y_{i+1} bychom již měli znát y_{i-1} a y_i. To je splněno ve všech případech kromě $i = 1$. Známe y_0, ale potřebujeme i y_1. Tento problém je společný všem vícekrokovým metodám, je třeba je nastartovat. V tomto případě bychom y_1 odhadli nějakou jednokrokovou metodou, nejlépe druhého řádu (tedy asi RK2), abychom nevnášeli hned na začátku chybu horšího řádu, než je u hlavní metody.

Na druhou stranu všechny vícekrokové metody mají oproti Runge-Kuttovým metodám jednu výhodu v tom, že hodnotu $F(x_i, y_i)$ počítáme jednou a pak využíváme v dalších krocích, zatímco RK musí pro každý krok vyhodnocovat F vícekrát, to může být pro rychlost důležitým faktorem.

Výhodou přístupu přes diference je, že zabere i u rovnic vyššího řádu. Uvažujme například rovnici $y'' + x = y \cdot y'$. Ta není separabilní ani lineární, tudíž na ni nezaberou základní analytické metody, a nelze ji převést na soustavu lineárních diferenciálních rovnic prvního řádu.

Použijeme-li diference, dostáváme

$$\frac{y_{i+1}+y_{i-1}-2y_i}{h^2} + x_i = y_i \cdot \frac{y_{i+1}-y_{i-1}}{2h} \implies 2y_{i+1} - hy_iy_{i+1} = 4y_i - hy_iy_{i-1} - 2y_{i-1} - x_i 2h^2,$$

což je explicitní dvoukroková metoda.

Metoda konečných diferencí zazáří nejvíce při řešení parciálních diferenciálních rovnic, kde se pracuje s vícerozměrnou sítí bodů a je třeba aproximovat parciální derivace pomocí sousedních hodnot. Právě pro PDR jsou také typické okrajové úlohy, které si s FDM vyloženě notují. Vzniklé matice mívají pravidelnou strukturu (například pětidiagonální) a jsou známy a studovány efektivní metody k jejich zpracování.

Pro úplnost ještě dodejme, že zvolená síť závisí na oblasti, na které problém řešíme. Pokud je to čtverec či hranol, pak se hodí pravidelná čtvercová síť. Pro jiné tvary se často přechází například k síti z trojúhelníků, což vyžaduje sofistikovanější způsoby odhadu pro parciální derivace. Je to velmi zajímavá a zároveň extrémně užitečná (simulace, předpověď počasí) oblast matematiky, jsou jí věnovány rozsáhlé knihovny tlustých knih.