Chyby a jejich šíření (Error propagation)

Při numerických výpočtech je mnoho zdrojů chyb. Nejčastější jsou následující:

- chyby ve vstupních údajích,
- chyby metody,
- chyby výpočtu.

Podíváme se na ně blíže.

Chyba na vstupu: Zde je tradiční hovořit o „nejistotě“ (uncertainity), protože každé měření funguje jen s určitou přesností, která se odvíjí od použité metody. Typický výsledek měření pak není 13.7 cm, ale třeba 13.7 ± 0.15 cm. Toto v numerických výpočtech neovlivníme, je ovšem třeba si položit otázku, jak na takovou chybu náš algoritmus zareaguje.

Chyba metody: Málokdy máme k dispozici výpočet, který dává výsledek přesně (jako je třeba vzorec pro kořeny kvadratické rovnice). Většinou je k dispozici algoritmus, který nás ke správné odpovědi dovede libovolně blízko, ale přesný výsledek bychom dostali jen při nekonečně dlouhém výpočtu (např. při hledání kořenů polynomů vyššího stupně). Tento typ chyby se často specifikuje přesněji, třeba diskretizační chyba (discretization error) je označení pro situaci, kdy si spojitou situaci „nakouskujeme“ na konečný počet jednodušších (a přibližných) situací (viz numerický výpočet integrálu), „truncation error“ se zase používá v situacích, kdy namísto přesného vyjádření nekonečným vzorcem používáme jeho konečnou část (např. aproximace pomocí Taylorova vzorce).

Jde o chybu, která vznikne, i když jsou všechny kroky výpočtu provedeny zcela přesně. U každé metody se tuto chybu snažíme odhadnout a samozřejmě minimalizovat. Budeme se tomu věnovat v příslušných kapitolách.

Chyba výpočtu: Obvykle jí říkáme zaokrouhlovací chyba (round-off error, truncation error).

Počítače neumí pracovat s přesnými reálnými čísly, výpočty ve „floating point“ aritmetice musí výsledky ořezávat tak, aby se vešly do rozsahu rezervovaného pro číslo. Ani této chybě se nelze vyhnout.

Práce ve floating point má jeden zajímavý důsledek: počítač „nepozná nulu“. Jakmile vyjde ve výpočtu číslo menší než nejmenší možná floating point hodnota, počítač z něj udělá nulu. Je tu i problém opačný. Řekněme, že počítači vyjde po složitém výpočtu 10^{-13}. Je to v zásadě správný výsledek, nebo měla vyjít nula a tohle je způsobeno drobnými chybkami ve výpočtu? Neexistuje způsob, jak na tuto otázku spolehlivě odpovědět.

Při reálném běhu algoritmu se všechny tyto vlivy slévají, přičemž nejde o prosté sečtení, může to být dokonce horší (hyperaditivita). Chyba vzniklá v průběhu výpočtu totiž může další průběh vykolejit tak, že například chyba metody vzroste víc, než by jinak vzrostla. Chyby se tak slévají a snaží se průběh výpočtu svést na zcestí, nebezpečí je akutní zejména u algoritmů iteračních, kdy je opravdu šance chyby hromadit.

Odhadnout, o kolik se výsledek liší od správného, tedy není snadné. Abychom se v situaci dokázali vyznat, zkoumáme jednotlivé vlivy odděleně. Při hodnocení algoritmů si tedy obvykle klademe tyto otázky:

- Jaká je chyba metody, tedy chyba vzniklá při přesných datech a přesném výpočtu?
- Jak náchylný je algoritmus na vznik zaokrouhlovacích chyb?
- Jak se algoritmus zachová při chybě vstupních dat? Tomuto se říká „numerická stabilita“ algoritmu a vypovídá o tom, jak se v algoritmu „šíří chyba“. Toto zkoumání se dělá za předpokladu, že výpočet jako takový již probíhá přesně.

V praxi se někdy mluví o stabilitě metody a zahrnuje se do toho i náchylnost k zaokrouhlovací chybě, jinými slovy když už tedy akceptujeme chybu metody a očekáváme určitý (nepřesný) výsledek, tak jaká je při práci ve floating point aritmetice šance, že se výsledek od toho očekávaného výrazněji liší.

Zde se podíváme na základní fakta, tedy okolnosti vzniku zaokrouhlovací chyby a šíření chyby při základních operacích.

Definice.
Nechť x je číslo a $\hat{x}$ jeho odhad.
Pak definujeme **absolutní chybu** $E_x = x - \hat{x}$ a **relativní chybu** $\varepsilon_x = \frac{|E_x|}{|x|}$.
Libovolné číslo e_x splňující $|E_x| \leq e_x$ se nazývá **odhad** chyby.

Absolutní chyba je typická třeba v oblasti strojírenství. Při hodnocení věrohodnosti výpočtů je zásadní chyba relativní. Oba pojmy mají zásadní nedostatek v tom, že se nedají zjistit. Když se totiž rozhodneme spočítat odhad $\hat{x}$ nějaké hodnoty x, tak to určitě děláme právě proto, že vlastně x zjistit neumíme (nebo nechceme), tudíž ani neumíme spočítat rozdíl $x - \hat{x}$. Proto v praxi pracujeme s odhadem chyby e_x, který u vstupních dat vyplývá ze znalostí postupů, kterými jsme k údajům došli, u výsledků pak ze zkoumání toho nejpesimističtějšího scénáře.

Z odhadu chyby pak získáme odhad relativní chyby $\varepsilon_x = \frac{e_x}{|x|}$. Zde je ovšem podobný problém, protože x pořád neznáme. Umíme spočítat číslo $\frac{e_x}{|\hat{x}|}$ a můžeme doufat, že se od hledané relativní chyby (či jejího odhadu) moc neliší, ale jistotu nemáme. Nic lepšího ale v praxi není.

1. Počítání ve floating point

Jde o de facto jediný rozumný formát pro vědecké či technické výpočty a pro numerickou matematiku je naprosto zásadní schopnost počítače rychle s těmito čísly počítat, rychlost počítače se pak udává ve „flops“ neboli floating point operations per second. Moderní PC mívá rychlost pár giga flops.

Uvažujme situaci, kdy pracujeme-li s čísly o **základu** β („base“, zde budeme pracovat s $\beta = 10$) na p platných míst („precision“). Číslo p tedy určuje, s jakou přesností pracujeme. Typická vědecká kalkulačka má p v rozmezí $11 - -13$, u počítačů se nejčastěji používá přesnost daná standardem IEEE, což je $p = 24$ (ale při $\beta = 2$, což dává cca 7 dekadických desetinných míst) a $p = 53$ (opět v binárním tvaru, cca 16 dekadických desetinných míst). Jsou i další (přesnější) formáty.

V takové situaci jsou čísla uchovávána ve tvaru $d_1.d_2d_2 \cdots d_p \times \beta^e$, kde číslice $d_1, \cdots, d_p$ jsou tzv. **platná čísla** („significant digits“), jde o celá čísla mezi 0 a $\beta - 1$, výjimkou je d_1, které nesmí být nula. Exponent e má rovněž daný rozsah, ale do relativní přesnosti výpočtu přímo nezasahuje. Číslo $d_1.d_2d_2 \cdots d_p$ jako takové se pak nazývá **mantissa** („mantissa“ či častěji „significand“).

Pokud dostaneme obecné reálné číslo, je třeba jej do tohoto tvaru převést, což často zahrnuje ztrátu informace.

Fakt.
Při počítání ve formátu floating point se základem β s přesností na p míst je maximální relativní chyba rovna $\frac{1}{2}\beta \times \beta^{-p}$.

Bohužel, toto platí jen v případě, že máme svobodu si číslo upravit do základního tvaru $d_1.d_2d_2 \cdots d_p \times \beta^e$. Existují situace, kdy ji nemáme? Ano, v případě, že začneme s čísly pracovat. Jmenovitě pokud čísla sčítáme či odčítáme, pak je musíme převést na shodný exponent. Tím ovšem může klidně nastat situace, že celé dané číslo ztratíme a máme $\varepsilon = 1$ neboli stoprocentní chybu. Obecně se dá čekat při sečtení/odečtení N čísel chyba až $N\varepsilon$. Proti tomuto neexistuje přímá obrana, prostě se to musí ohlídat.

Trik 1: Než čísla sečteme, seřadíme je podle velikosti a začneme sčítat od nejmenších.

Trik 2: Předpokládejme, že máme sečíst čísla $x_1, \ldots, x_N$ (která třeba chodí postupně). Existuje zajímavý algoritmus zvaný „Kahan summation formula“:

```
c:=0; s:=x_1;
for j=2 to N do
    x:=x_j-c; y:=s+x; c:=(y-s)-x;
    s:=y;
```

Každopádně každý numerický matematik ví, že je velice nebezpečné sčítat/odčítat čísla, která se liší v řádu, čím více, tím je situace horší.

Formát floating point má ještě další dva významné rysy, a to je fakt, že existuje nejmenší vyjadřitelné kladné číslo a největší vyjadřitelné kladné číslo. Pokud nás výpočet zavede k číslu, které není nula, ale nevejde se do tohoto rozsahu, nastává průšvih, tzv. chyba podtečení či chyba přetečení („underflow error“, „overflow error“). I na to je třeba si dávat pozor a organizovat výpočty chytře.

2. Šíření chyb

Základem je vědět, co se s chybou na vstupu stane při základních algebraických operacích.

Fakt.
Uvažujme reálná čísla $x, y > 0$ a jejich odhady $\hat{x}, \hat{y} > 0$. Pak platí:
(i) $|E_{ax}| = |a| \cdot |E_x|, \qquad \varepsilon_{ax} = \varepsilon_x;$
(ii) $|E_{x+y}| \le |E_x| + |E_y|, \qquad \varepsilon_{x+y} \le \max(\varepsilon_x, \varepsilon_y);$
(iii) $|E_{x-y}| \le |E_x| + |E_y|, \qquad \varepsilon_{x-y} \le \max(\varepsilon_x, \varepsilon_y)\frac{x+y}{|x-y|};$
(iv) $|E_{1/x}| \le \frac{|E_x|}{x\hat{x}}, \qquad \varepsilon_{1/x} \le \frac{x}{\hat{x}}\varepsilon_x \le \frac{1}{1-\varepsilon_x}\varepsilon_x \le (1+2\varepsilon_x)\varepsilon_x$ (pro $\varepsilon_x < 1$);
(v) $|E_{x\cdot y}| \le y \cdot |E_x| + \hat{x} \cdot |E_y|$ nebo $|E_{x\cdot y}| \le y \cdot |E_x| + x \cdot |E_y| + |E_x E_y|$,
$\varepsilon_{x\cdot y} \le \varepsilon_x + \varepsilon_y \frac{\hat{x}}{x} \le \varepsilon_x + (1+\varepsilon_x)\varepsilon_y;$
(vi) $|E_{x/y}| \le \frac{1}{y}\big(|E_x| + |E_y|\frac{\hat{x}}{\hat{y}}\big)$ nebo $|E_{x/y}| \le \frac{1}{\hat{y}}\big(|E_x| + |E_y|\frac{x}{y}\big)$,
$\varepsilon_{x/y} \le \varepsilon_x + \varepsilon_y \frac{\hat{x}}{x}\frac{y}{\hat{y}} \le \varepsilon_x + \frac{1+\varepsilon_x}{1-\varepsilon_y}\varepsilon_y.$

Důkaz: (i): $E_{ax} = ax - a\hat{x} = a(x-\hat{x}) = aE_x$. Potom také $\varepsilon_{ax} = \varepsilon_x$.
(ii): $E_{x+y} = (x+y) - (\hat{x}+\hat{y}) = (x-\hat{x}) + (y-\hat{y}) = E_x + E_y$. Potom také (díky $x, y > 0$)

$$\begin{aligned}\varepsilon_{x+y} = \frac{|E_x+E_y|}{x+y} &\le \frac{|E_x|+|E_y|}{x+y} = \frac{|E_x|}{x}\frac{x}{x+y} + \frac{|E_y|}{y}\frac{y}{x+y} = \varepsilon_x\frac{x}{x+y} + \varepsilon_y\frac{y}{x+y} \\ &\le \max(\varepsilon_x,\varepsilon_y)\frac{x}{x+y} + \max(\varepsilon_x,\varepsilon_y)\frac{y}{x+y} = \max(\varepsilon_x,\varepsilon_y)\frac{x+y}{x+y}.\end{aligned}$$

(iv): $E_{1/x} = \frac{1}{x} - \frac{1}{\hat{x}} = \frac{\hat{x}-x}{x\hat{x}} = \frac{-E_x}{x\hat{x}}$. Pak $\varepsilon_{1/x} = \frac{|E_{1/x}|}{1/x} = \frac{1}{\hat{x}}|E_x| = \frac{x}{\hat{x}}\varepsilon_x$.
Dále $\frac{x}{\hat{x}} = \frac{x}{x-E_x} = \frac{1}{1-E_x/x} \le \frac{1}{1-|E_x|/x} = \frac{1}{1-\varepsilon_x}$.
Pro $\varepsilon < 1$ pak obecně platí $\frac{1}{1-\varepsilon} \le 1 + 2\varepsilon$.
(v): Druhý odhad pro chybu dostaneme výpočtem
$E_{x\cdot y} = xy - \hat{x}\hat{y} = xy - (x-E_x)(y-E_y) = xy - y(x-E_x) + E_y(x-E_x) = yE_x + xE_y - E_xE_y$.
První odhad získáme pomocí
$E_{x\cdot y} = xy - \hat{x}(y-E_y) = xy - \hat{x}y + \hat{x}E_y = xy - (x-E_x)y + \hat{x}E_y = yE_x + \hat{x}E_y$.
Odtud pak $\varepsilon_{x\cdot y} = \frac{|E_{xy}|}{xy} \le \frac{y|E_x|+|E_y|\hat{x}}{xy} = \frac{|E_x|}{x} + \frac{|E_y|\hat{x}}{xy} = \varepsilon_x + \varepsilon_y\frac{\hat{x}}{x}$.
Alternativa je pomocí $\frac{\hat{x}}{x} = \frac{x-E_x}{x} \le 1 + \frac{|E_x|}{x} = 1 + \varepsilon_x$.

□

Z pravidla pro násobení hned indukcí odvodíme, že pro mocninu x^n platí $|E_{x^n}| \le n\max(x,\hat{x})^n|E_x|$ a proto $\varepsilon_{x^n} \le n\big(\frac{\max(x,\hat{x})}{x}\big)^n\varepsilon_x$.
Ukázali jsme vzorce, kde na vstupu a výstupu máme stejný druh chyby. Podle toho, jaké aplikaci bude výpočet sloužit, se ale můžeme setkat i s případy smíšenými, třeba na vstupu máme chybu absolutní a rádi bychom chybu relativní. Úvahy jsou pak obdobné.

Výsledky jsou to pěkné, ale jednak v praxi nezjistitelné (viz diskuse výše) a navíc občas komplikovanější, než bychom rádi. Při praktických výpočtech se proto obvykle přechází k jednodušším odhadům chyby, které již nejsou zaručené, ale v praxi dávají užitečnou informaci. Dostaneme se k nim z přesných odhadů pomocí trochy optimismu. Pokud doufáme, že odhad výsledku je již velmi věrný, pak lze odhadovat například $\frac{\hat{x}}{x} \sim 1$, čímž se přesné odhady silně zjednoduší. Je také možné použít úvahu, že pokud jsou

odhady velmi věrné, pak je ε_x velice blízké nule, v takovém případě je ε_x^2 zanedbatelné ve srovnání s ε_x. Odtud pak například dostaneme, že $(1+2\varepsilon_x)\varepsilon_x = \varepsilon_x + 2{\varepsilon_x}^2$ je vlastně přibližně rovno ε_x.

Aplikujme toto na na situaci, kdy hlavním zdrojem chyby je zaokrouhlování ve „floating point“ aritmetice, jinými slovy je to chyba relativní a pro vstupní data je vždy stejná, označme ji ε. Vzorečky výše pak vypadají následovně:

- $\varepsilon_{ax+y} = 2\varepsilon$,
- $\varepsilon_{x-y} = \varepsilon\frac{x+y}{|x-y|}$,
- $\varepsilon_{1/x} = \varepsilon_x$ a $\varepsilon_{x^n} = n\varepsilon$,
- $e_{xy} = \varepsilon_{x/y} = 2\varepsilon$.

Výrazně nebezpečné je evidentně odčítání, pokud jsou ta čísla blízká.

Shrneme-li naše úvahy o aritmetických operacích, tak násobení a dělení je vcelku bezpečné ohledně šíření chyby, pokud tedy nedojde k podtečení či přetečení výsledku. Vysoce rizikové je sčítání/odčítání čísel odlišných řádů a odčítání podobných čísel.

Chyba při výpočtu funkce.

Někdy dokážeme odvodit výsledek pomocí speciálních vlastností dotyčné funkce.

Příklad: Jak se šíří chyba při výpočtu exponenciály? Nejprve odhad chyby:

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

Je ovšem možné také počítat

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

odtud

$$\varepsilon_{e^x} = (1 - e^{-E_x}) = (1 - e^{-x\varepsilon_x}).$$

Je-li relativní chyba pevně dána (například přesností výpočtu), tak pro vzrůstající x je $\varepsilon_{e^x} \to 1$, což je velice nepříjemné.

△

Obecně lze k odhadu použít Taylorův polynom v bodě x. Máme

$$f(x+h) = f(x) + f'(x)h + \tfrac{1}{2}f''(x)h^2 + \tfrac{1}{3!}f'''(x)h^3 + \cdots$$

Pokud toto aplikujeme na $x + h = x - E_x = \hat{x}$, dostáváme

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

Pokud je E_x velmi malé, lze odhadovat $E_{f(x)} \sim f'(x)E_x$ a následně $\varepsilon_{f(x)} \sim \frac{f'(x)\cdot x}{f(x)}\varepsilon_x$.

Například pro dosazování do sinu máme (v inženýrském zaokrouhlení) $|E_{\sin(x)}| \leq |\cos(x)| \cdot |E_x|$ a následně $\varepsilon_{\sin(x)} \leq \left|\cos(x)\frac{x}{\sin(x)}\right|\varepsilon_x$. Pro velmi malá x je $\cos(x)\frac{x}{\sin(x)} \sim 1$, takže to není tak špatné.

△

Poznámka:

Tradičně se měření interpretují tak, že jde vlastně o náhodné veličiny, udávaný rozsah $x \pm E_x$ znamená, že E_x je standardní odchylka σ příslušné náhodné veličiny. Pokud jsou veličiny $x, y, \ldots$ nezávislé a chyby E mají náhodné rozdělení, pak ten nejpesimičtější scénář výše přichází jen zřídka, častěji se chyby navzájem alespoň částečně vyruší. Má smysl se ptát, jakou chybu lze očekávat (ve smyslu pravděpodobnostním, tedy jaká chyba nastává ve velké většině případů).

Pro běžné operace lze použít standardní pravděpodobnostní teorii pro výpočet odchylky u součtu, rozdílu atd. náhodných veličin, dojde se tak k následujícímu:

- Pravděpodobné E pro $ax \pm y$ je $\sqrt{a^2{E_x}^2 + {E_y}^2}$.
- Pravděpodobné ε pro xy a x/y je $\sqrt{{\varepsilon_x}^2 + {\varepsilon_y}^2}$.
- Pro funkci $f(x, y, z, \ldots)$ pak je pravděpodobná chyba rovna

$$E = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 \cdot {E_x}^2 + \left(\frac{\partial f}{\partial y}\right)^2 \cdot {E_y}^2 + \left(\frac{\partial f}{\partial z}\right)^2 \cdot {E_z}^2 + \cdots}.$$

△

3. Podmíněnost úlohy

Při řešení dané úlohy konkrétním výpočtem (metodou) je chování vstupní chyby ovlivněno používanými výpočty, opět budeme předpokládat, že probíhají přesně. Někdy se stane, že dotyčný komplexní výpočet vstupní chyby nemnoží či dokonce utlumuje, pak říkáme, že je to „stabilní metoda“. Existují ovšem také metody nestabilní, u kterých při nepříznivých vstupních datech dojde k explozivnímu nárustu chyby. Při zkoumání stability metody určitě přijdou vhod odhady šíření chyb odvozené výše.

U některých úloh nezáleží na konkrétní implementaci, jakýkoliv výpočet (metoda řešení) nedokáže obecně zabránit nárustu chyby, pokud máme smůlu na vstupní data. O nepříznivém typu vstupních dat (konkrétní matice, konkrétní rovnice atd.) Říkáme, že je „špatně podmíněný“, mluvíme tedy například o špatně podmíněné matici. Naopak rádi máme objekty „dobře podmíněné“, u kterých při řešení úloh problémy s šíření chyb nejsou.

Příklad: Uvažujme výpočet determinantu matice, konkrétně $\begin{vmatrix} 100 & 100 \\ 100 & x \end{vmatrix}$ pro $x = 101$. Hravě spočítáme, že výsledek je 100. Do výpočtu teď vneseme půlprocentní chybu na vstupu, tedy dosadíme $x = 100.5$. Dostáváme výsledek 50, relativní chyba je tedy 0.5 neboli padesát procent! Tento výpočet je proto spíše nestabilní a dá se čekat, že dotyčná matice bude špatně podmíněná, ať už je přesná definice tohoto pojmu jakákoliv.

Matici teď mírně upravíme, budeme počítat $\begin{vmatrix} 100 & 130 \\ 120 & x \end{vmatrix}$. Dosazení 101 dává -5500, zatímco dosazení 100.5 dává -5550. Relativní chyba výsledku je teď jen 0.009 neboli necelé procento, což je mírně větší než na vstupu, ale ne zas výrazně. Bez přesné definice není jasné, zda již jde o dobře podmíněnou matici, ale rozhodně to není tragedie.

Nakonec původní matici upravíme ještě jednou, přidáním znaménka: $\begin{vmatrix} 100 & -100 \\ 100 & x \end{vmatrix}$. Dosazení $x = 101$ dává 20100, zatímco dosazení 100.5 dává 20050. Relativní chyba je cca 0.0025, poloviční ve srovnání s relativní chybou vstupu. Jakkoliv to může vypadat zvláštně, dostáváme případ, kdy je výsledek věrohodnější než vstupní data, máme krásně stabilní výpočet. Toto tedy nejspíše bude dobře podmíněná matice.

Vidíme, že výpočet determinantu matice je úloha, u které je věrohodnost výsledku závislá na štěstí při zadání matice, jde tedy o nestabilní výpočet a je nutné být opatrný.

△

4. Značení O

Při numerickém výpočtu obvykle máme možnost volit, jak přesně budeme pracovat, často hovoříme o kroku metody h, což je typicky malé kladné číslo. Pokud se například rozhodneme zaokrouhlovat všechna čísla na jedno desetinné místo, tak pracujeme s možnou chybou $h = 0.05$. Při numerické integraci typicky h reprezentuje, jaký je rozestup bodů, ve kterých integrovanu funkci vyhodnocujeme.

Tato vstupní přesnost pak ovlivňuje hodověrnost výsledku, při troše štěstí ji dokážeme vyjádřit.

Příklad:

Uvažujme čísla $x, y > 0$. Z důvodu zaokrouhlení použijeme místo nich čísla $x + h$, $y + h$. Při sčítání pak dostaneme

$$(x + h) + (y + h) = x + y + 2h,$$

dá se tedy čekat (maximální) chyba $2h$. Při násobení

$$(x + h) \cdot (y + h) = xy + (x + y)h + h^2$$

se pak dá čekat chyba $(x + y)h + h^2$.

△

Druhý příklad nás přivádí k vlastní terminologii. Vidíme tam jakoby dva zdroje chyby, $(x+y)h$ a h^2, a nabízí se otázka, zda je dokážeme porovnat. Trocha experimentování ukáže, že když začneme h zmenšovat, tak se dříve či později člen h^2 stane zanedbatelným v porovnání s členem $(x+y)h$, a to bez ohledu na volbu x, y. Jediný vliv těchto dvou konstant spočívá v určení okamžiku, jak malé h musíme vzít, aby už h^2 začalo být ve srovnání s $(x+y)h$ zanedbatelným. Je to pěkně vidět z výpočtu

$$\lim_{h\to 0^+}\Big(\frac{h^2}{(x+y)h}\Big) = 0.$$

Můžeme tak neformálně říct, že při násobení je (pro malá h) chyba v zásadě dána výrazem $(x+y)h$. Tyto úvahy jsou velice užitečné nejen v matematice, ale i v inženýrských oborech, fyzice atd., proto se vyplatí vymyslet jazyk, který by je dokázal vyjádřit a podepřít korektní matematikou.

Definice.
Uvažujme číslo $a \in \mathbb{R}$, popřípadě $a = \pm\infty$, nechť f, g jsou dvě funkce definované na nějakém (prstencovém) okolí bodu a.
Řekneme, že $f \in o(g)$ pro $x \to a$, jestliže $\lim_{x\to a}\Big(\frac{|f(x)|}{|g(x)|}\Big) = 0$.
Řekneme, že $f \in O(g)$ pro $x \to a$, jestliže existuje nějaká konstanta C a prstencové okolí P bodu a takové, že $|f| \le C|g|$ na P.

Říkáme, že „f je malé ó g“ či „f je velké ó g“. Formálně se dá brát $o(g)$ a $O(g)$ jako množina všech funkcí f, které splňují podmínku z definice, s tím se ovšem zase trochu pere fakt, že velká část uživatelů píše spíš $f = o(g)$ a $f = O(g)$. Já to budu dělat taky.

Význam: Situace $f = o(g)$ říká, že když je x velice blízké k a, tak se dá f v porovnání s g zanedbat. Často se to používá v případě $a = \infty$, v nekonečnu je třeba $x = o(x^3)$, takže pokud dosazujeme opravdu velká čísla, tak $2x^3 - 15000x$ dává v zásadě stejné výsledky jako $2x^3$.

To bychom pomocí druhého pojmu vyjádřili jako $2x^3 - 15000x = 2x^3 + O(x)$ pro $x \to \infty$, pojem $O(g)$ nám tedy umožňuje „schovat“ nepodstatné faktory výrazu a soustředit se jen na ty části, které o něčem rozhodují. Můžeme také psát $2x^3 - 15000x = 2x^3 + O(x^2)$ pro $x \to \infty$, v obou případech říkáme, že pokud pracujeme s velkými čísly, tak je náš výraz složen z $2x^3$ a něčeho dalšího, ale to něco nás nezajímá, protože je to zastíněno tím konkrétním výrazem. Přesně tento způsob vyjadřování se často hodí při aproximaci.

Pomocí O také můžeme vyjádřit, že umíme shora odhadnout rychlost růstu, tedy $2x^3 - 15000x = O(2x^3)$ či ještě lépe $2x^3 - 15000x = O(x^3)$ pro $\to \infty$, protože značené O umožňuje nepsat násobící konstantu.

Nás ovšem bude zajímat zejména případ $a = 0$ (často zajímá také fyziky), typicky dokone pracujeme jen s kladnými čísly.

Fakt.
Jestliže $a > b \ge 0$, tak $h^a = o(h^b)$ pro $h \to 0$, popř. $h \to 0^+$.

Mezi pojmy je zjevná hierarchie a také užitečná vazba.

Fakt.
Předpokládejme, že $f = o(g)$ pro $x \to a$. Pak platí:
(i) $f = O(g)$ pro $x \to a$.
(ii) $f + g = O(g)$ pro $x \to a$.

Část (ii) je matematickým ospravedlněním pro obvyklý přístup ke schovávání nepodstatných částí do velkého O. Při praktickém používání se nám často stane, že se různá „óčka“ potkají v rámci aritmetických výrazů. Dají se vypracovat jednoduchá pravidla:

Fakt.
(i) Pro $a \geq b \geq 0$ platí $\alpha O(h^a) \pm \beta O(h^b) = O(h^b)$ pro $h \to 0$.
(ii) Pro $a, b \geq 0$ platí $O(h^a) \cdot O(h^b) = O(h^{a+b})$ pro $h \to 0$
(iii) Pro $a \geq b \geq 0$ platí $O(h^+)/O(h^b) = O(h^{a-b})$ pro $h \to 0$.

Šla by vymyslet i další pravidla, třeba pro míchání o a O, ale tato nám bohatě postačí. Popravdě řečeno je asi i zbytečné si něco takového pamatovat, pokud člověk dobře rozumí hierarchii mocnin a významu symboliky O, tak jsou tyto vztahy zřejmé.

5. Formát čísel

V této kapitole jsme se dívali na numerickou matematiku pohledem inženýra, proto pro úplnost připomeňme základní pravidla pro technické vyjadřování výsledků. Východisko: Máme nějakou hodnotu výpočtu či měření x a k tomu odhad chyby e_x. Pravidla jsou následující:

• Čísla x a e_x se udávají ve vědeckém tvaru se stejným exponentem. Tedy ne $2.76 \times 10^5 \pm 0.3 \times 10^4$, ale $27.6 \times 10^4 \pm 0.3 \times 10^4$ neboli $(27.6 \pm 0.3) \times 10^4$.

• Bývá zvykem chybu e_x zaokrouhlit na jedno platné místo, s jednou výjimkou. Pokud je toto platné místo 1, pak se zaokrouhluje na dvě platná místa. Tradičně se volí takový řád vyjádření, aby první platná číslice chyby byla za desetinou tečkou. Při vyjádření výsledků se pak číslo x zokrouhlí tak, aby při vyjádření se stejným exponentem končilo u stejného řádu jako odhad chyby.

Příklad: $x = 4.274581 \times 10^6$, $E = 2.34 \times 10^3$, správné vyjádření výsledku je $427.5 \times 10^4 \pm 0.2 \times 10^4$.
Příklad: $x = 4.274581 \times 10^6$, $E = 1.34 \times 10^3$, správné vyjádření výsledku je $427.46 \times 10^4 \pm 0.13 \times 10^4$.

Není to pravidlo, ale při zaokrouhlování bývá zvykem, že pokud je odříznutá (dále nezobrazovaná) část rovna přesně 5, pak se zaokrouhlí nahoru či dolů tak, aby bylo zobrazené číslo sudé. Takže například při zaokrouhlování na dvě platná místa $125 \mapsto 120$ a $135 \mapsto 140$.
△