Úvod do numerické matematiky

1 Předmět numerické matematiky

Numerická matematika je věda, která se zabývá řešením matematicky formulovaných úloh pomocí logických operací a aritmetických operací s čísly o konečné délce.

2 typy matematicky formulovaných úloh

- numericky formulované úlohy - jednoznačný funkční vztah mezi konečným počtem vstupních a výstupních dat, jedná se obvykle o algebraické úlohy, někdy je možno nalézt teoretické řešení úlohy pomocí konečné posloupnosti aritmetických a logických operací, jindy ne (lze nalézt pouze přibližné řešení)
- úlohy, které nejsou numericky formulované - obvykle úlohy matematické analýzy, ve kterých je obsažen nekonečně krátký krok (např. výpočet derivace, integrálu, řešení diferenciální rovnice); tyto úlohy je třeba nějakým způsobem převést na numerické úlohy

Numerickou metodou rozumíme postup výpočtu numerické úlohy nebo její převod na úlohu jednodušší či postup, který nahrazuje matematickou úlohu úlohou numerickou.

Algoritmem rozumíme realizaci numerické metody, tj. konkrétní konečnou posloupnost operací, která s požadovanou přesností převede vstupní data na výsledné hodnoty. Algoritmus lze programovat na počítači.

2 Chyby - nepřesnosti při řešení úloh

Zdroje chyb

1. Chyby vstupních dat (např. chyby měření, chyby modelu reality)
2. Chyby metody (Truncation errors) - v důsledku převedení matematické úlohy na numerickou
3. Zaokrouhlovací chyby (Roundoff errors) - v důsledku zaokrouhlování při výpočtech s čísly o konečné délce

Definice chyb

$\boxed{x}$ - přesná hodnota $\boxed{\tilde{x}}$ - přibližná hodnota

$$A(x) = |\tilde{x} - x| \leq a(x)$$

$A(x)$ - absolutní chyba $a(x)$ - odhad absolutní chyby

$$R(x) = \frac{A(x)}{|x|} \leq r(x) \qquad \longrightarrow \sim \quad r(x) \simeq \frac{a(x)}{|\tilde{x}|} \ldots . \text{pokud} r(x) \ll 1$$

$R(x)$ - relativní chyba $r(x)$ - odhad relativní chyby

Intervalový odhad x

$$\tilde{x} - a(x) \leq \ x \leq \tilde{x} + a(x) \quad \longrightarrow \ x = \tilde{x} \pm a(x)$$

$$x = \tilde{x} \cdot (1 \pm r(x))$$

Relativní chyba $\leftarrow\Rightarrow$ Počet platných číslic

Nechť $x \neq 0$, x zapíšeme pomocí číslic $x_1, x_2, \ldots, x_n$, nechť m $\leq$ n

$$x = \pm 0.x_1 x_2 \ldots x_{m-1} x_m x_{m+1} \ldots x_n \ \cdot \ 10^p$$

nechť $x_1, \ldots, x_m$ jsou platné cifry, $x_1 \neq 0$. Pak

$$r(x) < 5.10^{-m} \quad \ldots\ldots (\text{přesněji} \quad r(x) < 5.10^{-m}/x_1)$$

Přesnost výpočtu je obvykle dána relativní chybou.

3 Chyby metody (aproximace)

- Při výpočtech derivace, integrálu apod. nahrazujeme nekonečně krátký krok $\boxed{\mathrm{d}x}$ konečným krokem $\boxed{h}$
- Tento typ chyby nijak nesouvisí se zaokrouhlováním
- 1 veličinu lze aproximovat mnoha různými způsoby
- Řád metody - Je-li chyba $\boxed{\delta y}$ veličiny $\boxed{y}$ úměrná $\delta y \sim h^\alpha \sim O(h^\alpha)$ pak $\boxed{\alpha}$ nazýváme řádem metody

Ukázky různých způsobů aproximace - odvození chyby metody

derivace

$$y = \left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{x_1} \longrightarrow y \simeq \frac{f(x_1+h)-f(x_1)}{h} = f'(x_1) + \frac{h}{2}f''(x_1) + \cdots$$

$$\delta y = \frac{f''(x_1)}{2}\, h + \cdots \qquad \ldots\ldots \quad \text{metoda 1. řádu}$$

$$y = \left.\frac{\mathrm{d}f}{\mathrm{d}x}\right|_{x_1} \longrightarrow y \simeq \frac{f(x_1+h/2)-f(x_1-h/2)}{h} = f'(x_1) + \frac{h^2}{24}f'''(x_1) + \cdots$$

$$\delta y = \frac{f'''(x_1)}{24}\, h^2 + \cdots \qquad \ldots\ldots \quad \text{metoda 2. řádu}$$

integrál - obdélníková metoda šířka intervalu $h = \frac{b-a}{N}$

$$y = \int_a^b f(x)\mathrm{d}x \longrightarrow y \simeq h \sum_{i=1}^{N} f(x_i) \quad \ldots\ldots\ldots \quad x_i = a + \frac{2i-1}{2}\, h$$

chyba v 1 podintervalu

$$\begin{aligned} \int_{x_i-h/2}^{x_i+h/2} f(x)\ \mathrm{d}x &= \int_{-h/2}^{h/2} \left[f(x_i) + \delta f'(x_i) + \frac{\delta^2}{2} f''(x_i) + \cdots \right] \mathrm{d}\delta = \\ &= hf(x_i) + 0 + \frac{h^3}{12} f''(x_i) \end{aligned}$$

celková chyba

$$|\delta y| \simeq \left| \frac{h^3}{12} \sum_{i=1}^{N} f''(x_i) \right| \leq \frac{h^3}{12}\ N \max_{\{x_i\ i \in \langle 1,N \rangle\}} |f''(x_i)| \leq \frac{|b-a|}{12} \max_{x \in (a,b)} |f''(x)| h^2 \sim O(h^2)$$

Chyba metody vyššího řádu klesá rychleji při zmenšování kroku h. Pokud jsou metody jinak rovnocenné, vybereme metodu vyššího řádu.

Znalost řádu metody umožňuje

- Odhad chyby
- Zpřesnění výsledku

Nejjednodušší způsob - spočtu odhady výsledku y_h pro krok h a $y_{h/2}$ pro krok $h/2$

Příklad pro metodu 1.řádu

$$\begin{aligned} y_h &= y + ah + bh^2 \\ y_{h/2} &= y + a\frac{h}{2} + b\left(\frac{h}{2}\right)^2 \end{aligned}$$

Koeficienty a,b často nemohu spočítat, ale přesto platí

$$\begin{aligned} \delta y_{h/2} &\simeq y_h - y_{h/2} \qquad \ldots\ldots \quad (\simeq ah/2\ ) \\ \tilde{y}_h &= 2y_{h/2} - y_h = y - \frac{b}{2}\ h^2 = y + O(h^2) \end{aligned}$$

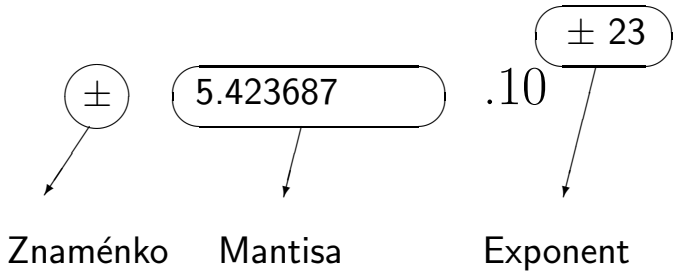
a tedy kombinací y_h a $y_{h/2}$ je chyba odhadnuta a řád metody o 1 zvýšen.

4 Zaokrouhlovací chyby

4.1 Reprezentace čísla v počítači

1. **Celá čísla** - přesné výpočty, velmi omezený rozsah
   - INTEGER - 2 byty (INTEGER*2) - 16 bitů 2^{16} čísel $\langle -32768, 32767 \rangle$
   - LONGINT - 4 byty (INTEGER*4) - 32 byty 2^{32} čísel $\langle -2^{31}, 2^{31} - 1 \rangle$

2. **Reálná čísla** - čísla v pohyblivé desetinné tečce - FLOATING POINT $\simeq$ vědecký tvar čísla



V počítači mantisa a exponent v dvojkové soustavě.

Délka MANTISY - tj. počet bitů na mantisu $\Longrightarrow$ přesnost čísla

přesnost $\Longleftrightarrow$ počet čísel mezi 1 a 2
interval mezi čísly mezi 1 a 2 je rovnoměrný - do paměti se mohou ukládat jenom čísla 1, $1 + \varepsilon$, $1 + 2\ \varepsilon$, $\cdots$, $2 - \varepsilon$. Čím více bitů na mantisu, tím menší $\varepsilon \Longrightarrow$ menší chyby při zaokrouhlování (u mezivýsledků je v registrech procesoru přesnost vyšší).
Byly uvedeny všechny mantisy, při změnách změnách exponentů se krok mezi čísly zvýší úměrně $2^{exponent}$ (relativní chyba čísla se ale nemění).

Délka EXPONENTU - tj. počet bitů na exponent - určuje rozsah

pozn. 1 exponent se zvlášť musí vyhradit pro 0, která nemá logaritmus, mantisy u tohoto exponentu lze využít pro vyznačení chyb (overflow, undefined), dále se může využít pro řídkou síť čísel pod minimem k ošetření podtečení (underflow)

8 - 11 bitů na exponent
8 bitů na exponent např. $\langle -128, 126 \rangle - 2^{2^7} \simeq 3.4 \ . \ 10^{38}$
9 bitů na exponent - $2^{2^8} \simeq 1.15 \ . \ 10^{77}$
10 bitů na exponent - $2^{2^9} \simeq 1.32 \ . \ 10^{154}$
11 bitů na exponent - $2^{2^{10}} \simeq 1.74 \ . \ 10^{308}$

Jednoduchá přesnost = 4 byty

TurboPascal	-	Single
Fortran	-	Real = Real*4
norma IEEE	často	1 bit znaménko + 8 bitů exponent
ne dělení M-E		+ 23 bitů mantisa $\Rightarrow \ \varepsilon \simeq 1.2 \ . \ 10^{-7}$

Přesnost 1,5 = 6 bytů

TurboPascal	- Real	1 bit znaménko + 8 bitů exponent
		+ 39 bitů mantisa $\Rightarrow \ \varepsilon \simeq 1.8 \ . \ 10^{-12}$

Dvojitá přesnost = 8 bytů

TurboPascal	-	Double
Fortran	-	Double = Real*8
norma IEEE	často	1 bit znaménko + 11 bitů exponent
		+ 52 bit mantisa $\Rightarrow \ \varepsilon \simeq 4.4 \ . \ 10^{-16}$

Další typy

TurboPascal - Extended (Real*10), Comp (Integer*8)

4.2 Šíření chyb ve výpočtech

Nebezpečné jsou operace, které mohou podstatně zvětšit relativní chybu !!

- sčítání, odečítání

$$a(x \pm y) = a(x) + a(y) \quad \longrightarrow \quad r(x \pm y) = \frac{a(x) + a(y)}{|x \pm y|}$$

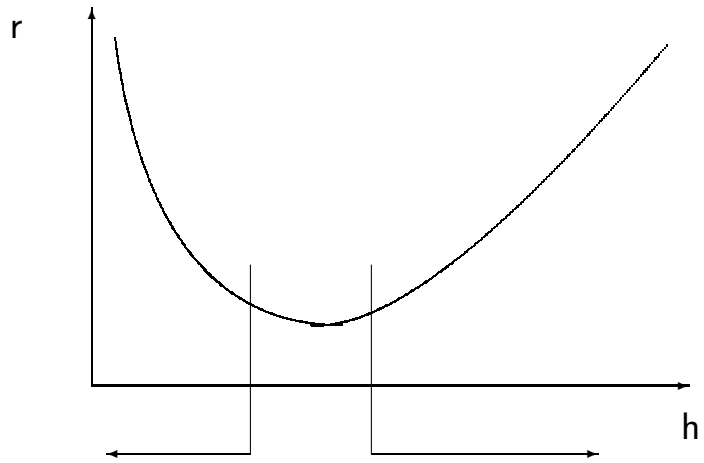
Pokud výsledek malý $\Longrightarrow$ zvětší se silně $\boxed{r}$!! Často nemohu rozhodnout, zda výsledek je 0 nebo není !!
Odečteme-li např. čísla 1.32483726, 1.32483357 známá na 9 platných číslic $r(x) = r(y) \simeq 5 \cdot 10^{-9}$ (přesněji $3.8 \cdot 10^{-9}$] dostaneme výsledek $x - y = 0.00000369$ s přesností na 3 platné číslice $r(x-y) \simeq 5 \cdot 10^{-3}$ (přesněji $r(x-y) = 1 \cdot 10^{-3}$).
Motivace vývoje řady numerických postupů - snaha vyhnout se odečítání dvou přibližně stejně velkých čísel.

- násobení, dělení

$$a(x \cdot y) = |x|a(y) + |y|a(x) \quad \longrightarrow \quad r(x \cdot y) = r(x) + r(y)$$
$$a\left(\frac{x}{y}\right) = \frac{|x|a(y) + |y|a(x)}{y^2} \quad \longrightarrow \quad r\left(\frac{x}{y}\right) = r(x) + r(y)$$

Násobení a dělení nemohou podstatně zvětšit zaokrouhlovací chybu, nejsou tedy nebezpečné.
Pozn. Dělení číslem 0 je hrubá chyba - nejde o zaokrouhlovací chybu.

4.3 Závislost charakteru chyby na velikosti kroku h



Zaokrouhlovací chyby při malém h vznikají z různých příčin - u derivace v důsledků odečítaní přibližně stejných čísel, u integrálu v důsledku počtu operací
Pozn. V obou případech při stanovení příliš krátkého kroku h může dojít i k hrubým chybám vzhledem k rozdílu rovném 0 v důsledku zaokrouhlení nebo vzhledem k opakovanému provádění součtů, které se při dané numerické přesnosti neprojeví $u + \delta u = u$!

5 Korektnost a podmíněnost úlohy

Korektnost úlohy

Definice: Nechť úlohou je najít řešení $\vec{y}\ \epsilon\ \mathbf{N}$ ($\mathbf{N}$ je množina možných řešení) pro zadaný vektor $\vec{x}\epsilon\ \mathbf{M}$ ($\mathbf{M}$ je množina vstupních dat). Pak úloha je korektní právě tehdy, jsou-li splněny následující dvě podmínky

1. $\exists$ právě jedno řešení $\vec{y}$ pro $\forall\ \vec{x}\epsilon\ \mathbf{M}$.
2. Řešení spojitě závisí na vstupních datech, tj. jestliže pro $\forall n$ z množiny přirozených čísel je $\vec{y}_n$ řešení pro vstupní data $\vec{x}_n$, a jestliže $\vec{y}$ je řešení pro vstupní data $\vec{x}$, nechť dále ρ je norma v množině vstupních dat a σ je norma v množině možných řešení, pak platí

$$x_n \xrightarrow{\rho} x \ \Rightarrow\ y_n \xrightarrow{\sigma} y$$

V praxi se řeší i nekorektní úlohy, ale 1. krok řešení spočívá v nalezení vhodného způsobu, jak převést úlohu na úlohu korektní (např. podmínkou na výsledek; interpretací vstupních dat; vhodnou volbou normy v prostoru řešení apod.)

Podmíněnost úlohy

Definice: Podmíněnost úlohy C_p je daná poměrem relativní změny výsledku ku relativní změně vstupních dat, tj.

$$C_p = \frac{\frac{\|\delta y\|}{\|y\|}}{\frac{\|\delta x\|}{\|x\|}} \approx \frac{r(y)}{r(x)}$$

Pokud $C_p \sim 1$, říkáme, že úloha je dobře podmíněná, pokud $C_p > 100$, úloha je špatně podmíněná.
Pokud je přesnost použitého typu čísel ε $(r(x) = \varepsilon)$, pak úloha s $C_p > \varepsilon^{-1}$ není v rámci dané přesnosti řešitelná.
Často se pro špatně podmíněné úlohy používají speciální metody, které omezují růst zaokrouhlovacích chyb.

Příklad: Soustava lineárních rovnic s maticí blízkou k singulární (špatně podmíněná matice). Nechť je dána úloha

$$x + \alpha\ y = 1$$
$$\alpha\ x + y = 0$$

Nechť vstupem je hodnota α a výstupem hodnota x. Pak

$$x = \frac{1}{1-\alpha^2} \qquad \text{a} \qquad C_p = \frac{\frac{|\delta x\|}{|x|}}{\frac{|\delta\alpha|}{|\alpha|}} \simeq \left|\frac{\alpha \frac{\mathrm{d}x}{\mathrm{d}\alpha}}{x}\right| = \frac{2\ \alpha^2}{|1-\alpha^2|}$$

Při $\alpha^2 \to 1$ je úloha špatně podmíněná.

6 Numerická stabilita

U nestabilní metody (algoritmu) se relativně malé chyby v jednotlivých krocích výpočtu postupně akumulují tak, že dojde ke katastrofální ztrátě přesnosti numerického řešení úlohy.

U stabilních metod roste chyba výsledku s počtem kroků N nejvýše lineárně (v ideální, ale vzácné situaci, kdy je znaménko chyby náhodné, chyba roste $\sim \sqrt{N}$). U nestabilních metod roste zaokrouhlovací chyba rychleji, např. geometrickou řadou $\sim q^N$, kde $|q| > 1$.

Nestabilita algoritmu vzniká v důsledku akumulace chyb. Typicky se objevuje v rekurzivních algoritmech. Nestabilita metody může vznikat jak v důsledku akumulace zaokrouhlovacích chyb, tak iv důsledku akumulace chyby metody, stabilita metody může záviset na velikosti použitého kroku h. Nestabilita metody se často objevuje při numerickém řešení počátečního problému pro obyčejné a parciální diferenciální rovnice.

6.1 Příklady nestabilních algoritmů

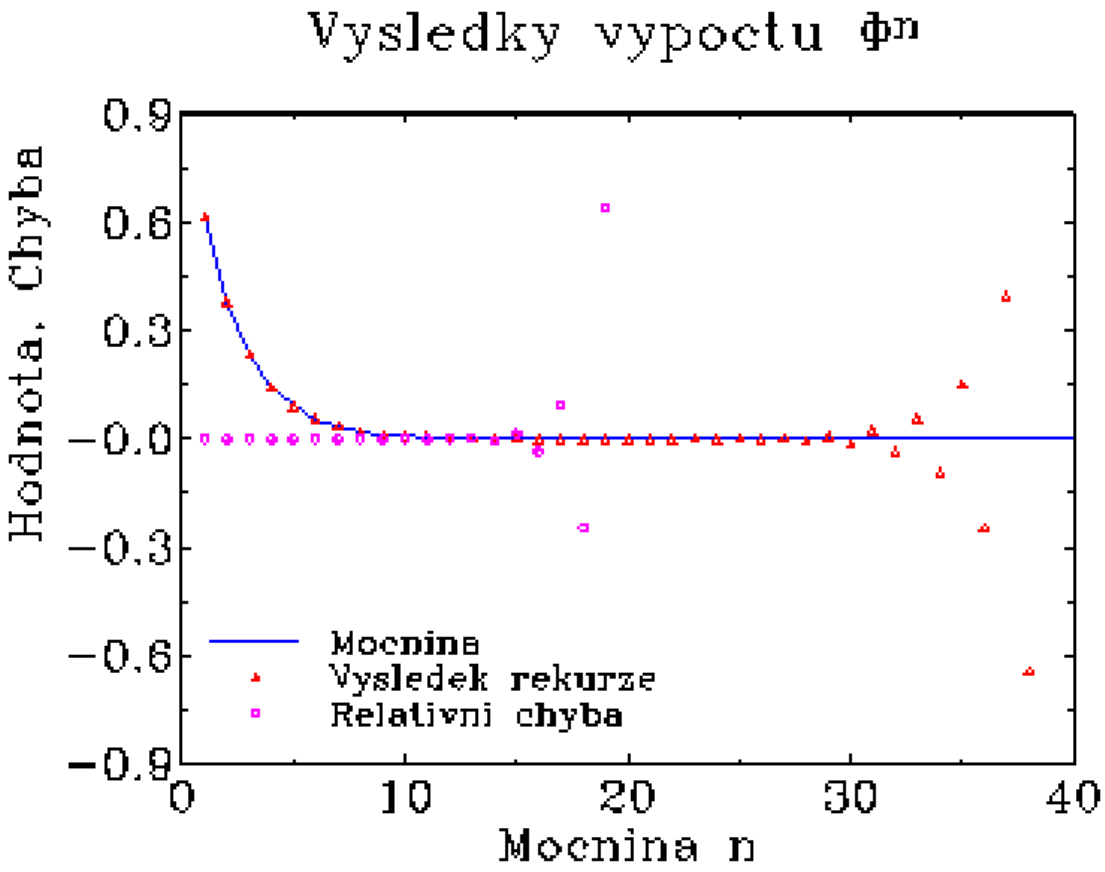
1. Nestabilní rekurze - ukážeme si na poněkud umělém případu počítání mocnin čísla Φ zvaného "Zlatý řez"

   $$\Phi \equiv \frac{\sqrt{5}-1}{2} \simeq 0.61803398$$

   Lehce ukážete, že mocniny Φ^n splňují jednoduchý rekursní vztah

   $$\Phi^{n+1} = \Phi^{n-1} - \Phi^n$$

   Protože známe $\Phi^0 = 1$ a $\Phi^1 = 0.61803398$ mohli bychom zkusit počítat mocniny odečítáním, což je obvykle rychlejší než násobení.



Obrázek ukazuje, že uvedený postup zcela nepoužitelný, při jednoduché přesnosti dostaneme viditelné chyby výsledky už od $n = 16$, kdy $\Phi^n \simeq 5.10^{-4}$. Pro $n = 20$ dostanu poprvé záporný výsledek rekurze, a tedy rekurze už nijak neaproximuje hodnotu mocniny. Nejdříve vzroste relativní chyba (chyba mění znaménko), pak se objeví záporné hodnoty Φ^n a nakonec začne dokonce růst absolutní hodnoty Φ^n. Nestabilita se projeví i ve dvojité přesnosti, zaokrouhlovací chyba narůstá ale z menší hodnoty a tak by se 1. záporný výsledek rekurze objevil pro n=40.

Příčina nestability je v tom, že uvedená rekursní formule má ještě druhé řešení $\Phi_2 = -(\sqrt{5}+1)/2 < -1 < -\Phi$. Protože rekurzivní relace je lineární, absolutní velikost zaokrouhlovací chyby bude narůstat geometrickou řadou s kvocientem $q = |\Phi_2| > 1$. Protože navíc řešení klesá, relativní velikost zaokrouhlovací chyby roste geometrickou řadou s kvocientem $q' = |\Phi_2|/\Phi > 1$.

Uvedený příklad byl umělý, nicméně u mnoha speciálních funkcí (např. Besselovy funkce) se k výpočtu hodnoty funkcí různých řadů používají podobné rekurzivní relace, vždy ovšem tak, aby metoda byla stabilní.

2. Nestabilní metoda pro výpočet obyčejných diferenciálních rovnic

   Nechť řešíme obyčejnou diferenciální rovnici 1. řádu

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

   Na příkladu rovnice $y' = -y$ s počáteční podmínkou $y(0) = 1$ řešené ve směru růstu proměnné x ukážeme, že dvoukroková metoda 2. řádu

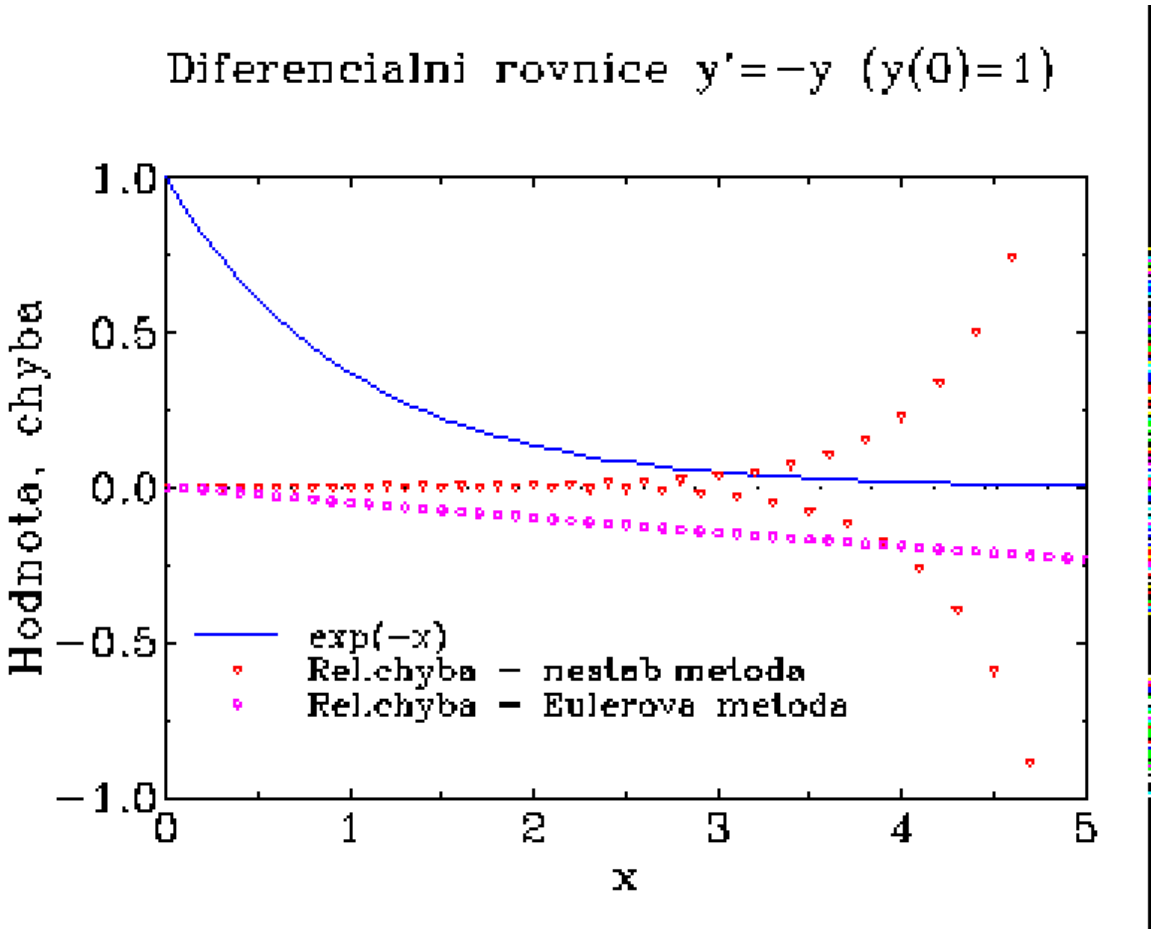
   $$y' \simeq \frac{y(x+2h) - y(x)}{2h} = -y(x+h)$$

   je nestabilní. Jde vlastně o podobnou rekurzi jako výše a pro poměr q = y(x+h)/y(x) existují 2 řešení, $q_1 = -h + \sqrt{1+h^2}$ je v absolutní hodnotě menší než 1 a odpovídá prvním třem členům Taylorova rozvoje řešení $y = y(0) \cdot \exp(-x)$, druhý kořen $q_2 = -h - \sqrt{1+h^2}$ je v absolutní hodnotě větší než 1 způsobuje nestabilitu algoritmu.

   Na následujícím grafu je porovnána celková relativní chyba uvedené nestabilní metody s chybou Eulerovy metody

   $$y(x+h) = y(x) + h \cdot y'(x) = y(x) - h \cdot y(x)$$

   Eulerova metoda se obvykle nepoužívá, neboť jde o metodu 1. řádu s velkou chybou metody, nicméně je pro uvedený případ stabilní vůči zaokrouhlovacím chybám.

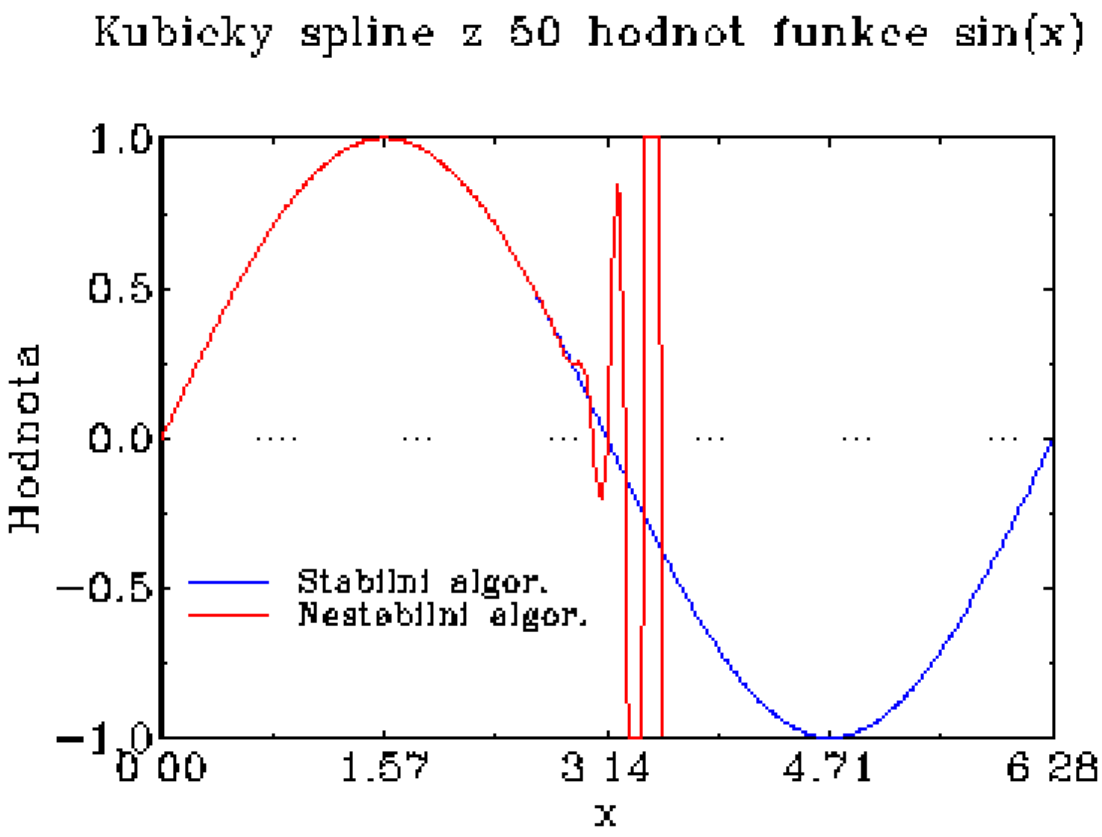


Obrázek ukazuje, že na začátku řešení je nestabilní metoda vzhledem k relativně malé chybě metody přesnější, ale postupný růst chyby přivede nakonec ke katastrofálním chybám. Katastrofálním chybám nelze zabránit zkracováním kroku, užití dvojné přesnosti katastrofu pouze oddálí. U stabilní metody roste chyba s délkou intervalu nejvýše lineárně a chybu lze zmenšit zkracováním kroku.

3. Nestabilní spline

Při interpolaci dat pomocí kubického splinu (lokální interpolace kubickým polynomem se spojitou derivací) je třeba zadat 2 podmínky (např. hodnotu derivace funkce) v obou krajních bodech. Nesprávnou a nestabilní metodu dostaneme, pokud obě podmínky zadáme v 1 z okrajových bodů. Pokud jako 2. podmínku v prvním okrajovém bodu zadáme např. jako 2. derivaci rovnou hodnotě druhé derivace, která vyšla při stabilním postupu, tj. zadaných 1. derivacích v obou okrajových bodech, obě úlohy jsou z matematického hlediska zcela ekvivalentní a v případě počítání s přesnými čísly bych dostal totožný výsledek. Pokud však numericky počítám s ko-

nečnou délkou čísel, zaokrouhlovací chyba však při postupném počítání od 1 okraje narůstá a řešení začne mezi zadanými body silně oscilovat.



Na grafu je ukázáno je několik prvních oscilací chybně počítaného kubického splinu, další hodnoty dále oscilují, ale jejich hodnota je velmi velká (až 10^{13}). Při počítání v dvojité přesnosti se viditelné oscilace objeví pro $x > 4$.

7 Volba metody (algoritmu)

- **Základním požadavkem je možnost vyřešení úlohy s dostatečnou přesností**. Často je sledována konvergence[1], což znamená schopnost vyřešit úlohu s libovolně vysokou přesností (omezené jen zaokrouhlovací chybou) při kroku $h \to 0$ nebo při počtu operací $N \to \infty$.
- Při výběru metody hraje roli i složitost algoritmu (počet operací nutných k získání výsledku se zadanou přesností) a paměťové nároky.
- Je k dipozici spolehlivá implementace příslušné metody?

Numerické knihovny

- Pro drtivou většinu úloh jsou k dispozici procedury ve standardních knihovnách. Pokud úloha není triviální, neprogramuji ji sám!
- Většina knihoven je ve FORTRANU
- Profesionální knihovny jsou drahé (bývají k dispozici na velkých počítačích) - nejznámější NAG, IMSL
- Pro ukázky budeme používat knihovny NUMERICAL RECIPES (je přílohou knihy) - FORTRAN, C, Pascal
- - volně (byť často s omezeními) dostupný software - vyhledávání na http://gams.nist.gov, mnoho softwaru na serverech NETLIB, např. http://www.netlib.org.

[1] konvergenci budeme přesněji definovat pro konkrétní úlohy