Michal Křížek; Milan Práger; Emil Vitásek
Spolehlivost numerických výpočtů

Pokroky matematiky, fyziky a astronomie, Vol. 42 (1997), No. 1, 8--23

Persistent URL: http://dml.cz/dmlcz/139198

Neustálá pohromu věštící varování vědců nakonec už nejsou brána vážně. Podnebí se vlastně tak rychle nemění, určitě ne v průběhu jednoho volebního období německého parlamentu. Ale není ani seriózní, když politici hodnotí loňské horké léto jako úkaz blížící se změny klimatu. A je stejně nebezpečné, když se pokoušejí zástupci hospodářské sféry bagatelizovat vědecká pozorování a věří, že nemusejí vyvozovat žádné důsledky. U politiků rozhoduje bohužel často stranická příslušnost o tom, jakým způsobem přijímají vědecké výpovědi.

Extrémní požadavky po nicnedělání na jedné straně a obrovský aktivismus na straně druhé zpravidla neslibují žádné řešení. Vědcům je ale téměř znemožněno vést o těchto důležitých otázkách neideologickou diskusi.

Po tomto praktickém a aktuálním problému základního výzkumu a společnosti bych se rád vrátil ještě jednou na začátek. Jakým způsobem se má pěstovat a podporovat základní výzkum, aby prospíval společnosti? Dovolte mi skončit slovy slavného přítele Alexandera von Humboldta, kterého obdivoval. Konkrétně slovy muže, který byl zároveň přítelem a obdivovatelem vaší země. Mám na mysli Johanna Wolfganga Goethe, který řekl: *„Ale ve vědě je ta nejabsolutnější svoboda nutná: jen pak účinkuje nejen dnes a zítra, ale po nekonečně pokračující časová údobí.“*

Spolehlivost numerických výpočtů

Michal Křížek, Milan Práger a Emil Vitásek, Praha

„How much do you believe your numerical results?“ (Do jaké míry věříte svým numerickým výsledkům?) Takto se často ptá prof. Ivo Babuška na mezinárodních konferencích věnovaných numerické matematice. Použijme jeho okřídlené otázky jako příležitosti k malému zamyšlení nad numerickým počítáním.

Otázka se týká kvality numerických výsledků. To jsou čísla vyjádřená ve tvaru g-adického zlomku (při ručním počítání dekadického, při počítání na počítači většinou dvojkového).

Numerickým hodnotám čísel se nelze vyhnout, chceme-li popsat objekty reálného světa a vztahy mezi nimi. Tak např. měděný drát má specifický odpor $1{,}745 \cdot 10^{-8}\ \Omega\mathrm{m}$ a na tom se nedá nic změnit. Chceme-li tento drát bezpečně použít k přenosu elektrické energie, musíme tuto hodnotu vzít v úvahu při výpočtu intenzity proudu, tloušťky vodiče apod.

RNDr. Michal Křížek, DrSc. (1952), RNDr. Milan Práger, CSc. (1930), a RNDr. Emil Vitásek, CSc. (1931), jsou vědečtí pracovníci Matematického ústavu AV ČR, Žitná 25, 115 67 Praha 1, e-mail: krizek@beba.cesnet.cz.

Tato práce vznikla v rámci grantu č. A 1019601 GA AV ČR.

Pro získání numerických hodnot se nabízejí v podstatě dvě cesty. První cestou je vypočítat řešení vzorcem (analyticky, teoreticky), tj. získat poměrně jednoduchou a přehlednou kombinaci písmen a matematických symbolů. Do tohoto vzorce pak dosadíme numerické hodnoty námi použitých objektů (většinou geometrické a fyzikální konstanty) a získáme numerickou hodnotu, která nám umožní učinit potřebné rozhodnutí.

Druhou cestou je dosadit numerické hodnoty už do formulace problému, a pak manipulací s těmito hodnotami (tj. prováděním aritmetických operací, výpočtů hodnot jednoduchých funkcí) dojdeme k požadovanému výsledku. Tento postup nikterak není kopií cesty, kterou se ubíráme při analytickém řešení. Konstrukce metod pro tyto výpočty je předmětem numerické matematiky. Této druhé cestě se pak říká numerické počítání.

Na první pohled je patrný rozdíl mezi oběma postupy. Při prvním postupu, kromě zisku požadované numerické hodnoty, dostaneme vzorec, z něhož můžeme vyčíst závislost výsledku na parametrech (nezávisle proměnných) a mnoho jiných užitečných věcí.

Tuto možnost nám druhá cesta nedává, i když opakováním výpočtů můžeme některé dílčí závislosti zjistit. Výhodou ovšem je, že takto lze postupovat i v případech, kdy je první cesta neschůdná.

Uveďme malý příklad. Mějme diferenciální rovnici

$$y' = 1 + y^2$$

s počáteční podmínkou $y(0) = 0$ a nechť je naším cílem najít hodnotu $y(1)$.

Přesné řešení rovnice, jak snadno zjistíme, je $y(x) = \operatorname{tg} x$, a tedy $y(1) = \operatorname{tg} 1 = 1{,}557\ldots$ Zároveň vidíme, že funkce v okolí bodu $x = 1$ roste, můžeme zjistit, jak asi rychle, a máme o jejím chování velmi dobrý přehled.

Zaměníme-li naši rovnici na $y' = x + y^2$ a ponecháme-li původní počáteční podmínku, řešení úlohy pomocí „konečného“ vzorečku neexistuje. Přesto dávno vyzkoušeným numerickým postupem snadno najdeme, že $y(1) = 0{,}557\ldots$

Výhodou numerického řešení je, že je použitelné pro řešení mnohem většího počtu úloh než řešení analytické, přičemž lze zajistit, že získané hodnoty budou dostatečně přesné.

Je samozřejmé, že je možno použít kombinace obou metod, zvláště když jsou v poslední době k dispozici počítačové programy pro symbolické manipulace [2, 6].

Tím se dostáváme k tomu, abychom se podívali podrobněji na čísla, s nimiž počítač pracuje. Místo množiny reálných čísel je totiž na počítači pouze konečná množina tzv. čísel zobrazených v pohyblivé řádové čárce, což je číslo 0 a čísla tvaru

$$x = \pm 0{,}i_1 i_2 \ldots i_d \cdot z^E, \tag{1}$$

kde $d, z, E, i_1, \ldots, i_d$ jsou celá čísla, $d \geqq 1$ je daná délka mantisy, $z \geqq 2$ je daný základ (většinou 2, 8, 10, 16), E je exponent ležící v daném intervalu $E_- \leqq E \leqq E_+$, $i_1 \geqq 1$ a $0 \leqq i_j \leqq z - 1$ pro $j = 1, \ldots, d$. Při provádění aritmetických operací na počítači tak dochází k zaokrouhlování (pokud nepracujeme pouze s celočíselnými proměnnými).

Věříme, že hustota zvolené množiny racionálních čísel tvaru (1) je dost velká na to, abychom zachytili podstatné jevy vyskytující se v našem problému. Nemusí to být ovšem vždy pravda, a proto také počítače mají číselné sítě různé hustoty.

Vždycky, alespoň teoreticky, se může mezi dvěma sousedními racionálními čísly sítě objevit nějaký jev (velká změna hodnot funkce, peak funkce), který nebude zachycen. Potom je nutno užít speciální opatření.

Z této vlastnosti čísel užívaných při numerickém počítání a z toho, že jich můžeme zpracovat jen konečný, i když velký počet, vznikají některé problémy. Předně se musíme starat o to, jak funkce definované na nespočetné množině bodů (jako je třeba interval) nahradit jednoduššími funkcemi, které lze definovat konečným počtem hodnot, např. polynomy, funkcemi po částech lineárními apod. Tuto náhradu je zapotřebí učinit s dostatečnou přesností. Tato starost je hlavní náplní práce v numerické matematice.

Dále je třeba vědět, že se náš problém při převodu do numerického roucha poněkud „rozmaže", a toto „rozmazávání" pokračuje v průběhu celého výpočtu a je zásadní otázka, zda toto „rozmazávání" bude mít za následek také malou změnu výsledku nebo zda změna ve výsledku bude velká. Hovoříme pak o numericky stabilním nebo nestabilním chování.

Ústřední otázkou vědeckotechnických výpočtů je stanovit míru spolehlivosti získaných výsledků. Tato otázka nabývá na významu zejména v případě, kdy jde o bezpečnost lidí (např. při výpočtu přehrad, trupů letadel, jaderných elektráren). Hlavním zdrojem potíží u takových úloh je ale to, že jsou často nelineární, velkého rozměru, špatně podmíněné, nestabilní, mají násobná řešení, singularity apod. Kvalita získaného přibližného řešení pak závisí nejen na aritmetické přesnosti počítače, ale hlavně na volbě použitých numerických metod. Pokud po ukončení výpočtu neprovedeme odpovídající aposteriorní odhad chyby, nevíme vlastně, co jsme spočítali.

Cílem tohoto článku je poukázat na některá úskalí a nebezpečí při mechanickém používání numerických metod na počítačích bez znalosti teorie. Omezili jsme se přitom na 10 základních typů úloh. Příklady v nich jsme úmyslně volili jednoduché, aby byl výklad přístupný co možná nejširšímu okruhu čtenářů Pokroků, aby se hlavní myšlenky neztratily v technických detailech a abychom znali (pokud možno) přesné řešení, které je zapotřebí k výpočtu chyby (chyba je rozdíl přesného a přibližného řešení).

Pokud nemáme potíže s velikostí paměti počítače, je vždy lepší počítat ve dvojnásobné aritmetice než v jednoduché, protože výpočet netrvá o mnoho déle. Poznamenejme ještě, že v jednoduché, dvojnásobné, resp. rozšířené aritmetice (single, double, resp. extended precision) se číslo uchovává ve 4 (někdy pěti či šesti), 8, resp. 10 bytech. Ale ani dvojnásobná aritmetika není všelék. Budete-li například počítat hodnotu součtu $10^{20} + 1 - 10^{20}$, pak většina počítačů vám dá výsledek 0. Nulu naopak nemusíte dostat při výpočtu rozdílu $50000 \cdot 50000 - 50000{,}0 \cdot 50000{,}0$. První člen je totiž typu longinteger a příslušný součin se nevejde do 4 bytů, což některé překladače neohlásí. V tomto případě pak po odečtení druhého členu typu real dostanete nesmysly.

Některé softwarové produkty umožňují používat aritmetiku s „libovolnou" přesností. Všech 10 následujících úloh bylo řešeno na běžných osobních počítačích PC ve *dvojnásobné aritmetice.*

1. Řady

Ke značným potížím numerického charakteru může dojít při sčítání alternujících řad. Uvažujme nekonečnou řadu

$$s(x) = \sum_{k=0}^{\infty} \frac{(-1)^k x^{2k+1}}{(2k+1)k!} \quad \left(= \int_0^x \mathrm{e}^{-t^2}\,\mathrm{d}t \right), \tag{2}$$

která konverguje pro každé x reálné. Například pro $x = 10$ máme

$$s(10) = 10 - \frac{10^3}{3} + \frac{10^5}{10} - \frac{10^7}{42} + \cdots.$$

Vzájemné „katastrofální“ rušení velkých alternujících členů způsobí, že pro $x = 10$ dostaneme ve dvojnásobné aritmetice naprosto nesmyslný výsledek $-7{,}515 \cdot 10^{24}$. Laplaceův–Gaussův integrál v závorce (2) je totiž pro $x = 10$ kladný. Podobně pro $x = 7, 8, 9$ postupně dostaneme $-1{,}839 \cdot 10^2$, $-3{,}449 \cdot 10^8$ a $-2{,}269 \cdot 10^{15}$. Ve všech testech bylo sečteno jen 300 členů řady (2), protože další členy byly tak malé, že už částečný součet v konečné aritmetice počítače nezměnily. Pro $x = 6$ jsme dostali kladnou hodnotu 0,891, ale ta je paradoxně větší než hodnota integrálu v (2) pro $x = \infty$, tj. $\sqrt{\pi}/2 = 0{,}886$. K získání přijatelnějších výsledků by byla nutná aritmetika s mnohem větším počtem cifer (viz [12]).

Poznamenejme ještě, že v důsledku zaokrouhlování není strojová operace sčítání (a ani násobení) asociativní. Proto se při sčítání pomalu konvergující řady $s = \sum_{k=1}^{\infty} a_k$ vyplatí sčítat ji „odzadu“. Tak totiž můžeme sečíst více členů, než kdybychom ji sčítali odpředu. Pokud sčítáme řadu odpředu, k přibližné hodnotě s se sice dostaneme poměrně brzy, ale od jistého n je a_n vůči $s_n = \sum_{k=1}^{n} a_k$ tak malé, že se hodnota s_n přičtením a_{n+1} v počítačové aritmetice již nezmění. Naproti tomu při sčítání řady na počítači odzadu je zprvu částečný součet velmi malý, což (díky pohyblivé řádové čárce) umožňuje přičítat členy a_k, pro které $a_k \neq 0$ v počítačové aritmetice.

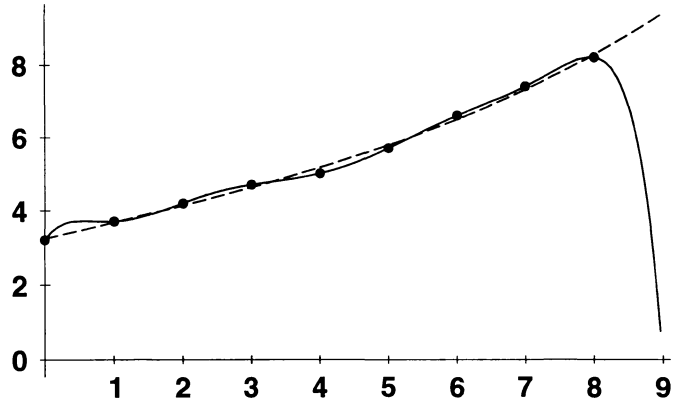
2. Interpolace

Hledejme polynom $p = p(x)$ stupně n, který pro $n+1$ různých argumentů x_i nabývá zadaných hodnot f_i, $i = 0, \ldots, n$. (Ty mohou představovat např. růst obyvatel.) Je tedy

$$p(x) = \sum_{i=0}^{n} f_i \ell_i(x),$$

kde $\ell_i = \ell_i(x)$ jsou elementární Lagrangeovy interpolační polynomy takové, že

$$\ell_i(x_j) = \begin{cases} 1 & \text{pro } i = j, \\ 0 & \text{pro } i \neq j. \end{cases}$$



Obr. 1. Plnou čarou je znázorněn graf interpolačního polynomu p stupně 8, který prochází zadanými hodnotami •; čárkovaně polynom stupně 3 získaný metodou nejmenších čtverců.

Předpokládejme, že $n = 8$, $f_0 = 3{,}2$, $f_1 = 3{,}7$, $f_2 = 4{,}2$, $f_3 = 4{,}7$, $f_4 = 5$, $f_5 = 5{,}7$, $f_6 = 6{,}6$, $f_7 = 7{,}4$, $f_8 = 8{,}2$, a pro jednoduchost uvažujme ekvidistantní dělení $x_i = i$ pro $i = 0, \ldots, 8$. Snadno zjistíme, že hodnota $p(9)$ je záporná, a tedy nikterak neodpovídá očekávanému monotónnímu růstu hodnot $p(i) = f_i$, $i = 0, \ldots, 8$ (viz obr. 1).

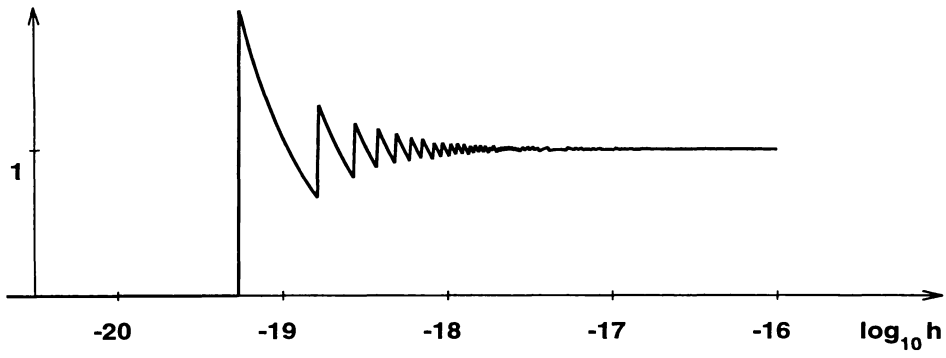
Nezkušený student se často domnívá, že přidáním dalších (naměřených) hodnot, např. v bodech $i + \frac{1}{2}$, by bylo možno nepříznivý průběh Lagrangeovy interpolace zlepšit. To však většinou situaci nezlepší, ale naopak způsobí nežádoucí oscilace interpolačního polynomu mezi zadanými hodnotami. Proto abychom nějakým způsobem vystihli monotonii hodnot f_i, použijeme k aproximaci polynom nižšího stupně, jehož koeficienty určíme metodou nejmenších čtverců (viz [11, 13]). Jeho graf sice obecně neprochází všemi zadanými hodnotami, ale jeho chování mimo interval $\langle 0, 8\rangle$ je „příznivější" — viz obr. 1. Jiná možnost je použít metodu extrapolace [11].

3. Numerické derivování

Pro reálné x uvažujme funkci $f(x) = 1 + x$. Hodnoty podílů

$$\frac{(1+h)-1}{h} \quad \text{pro } h \to 0 \tag{3}$$

se rovnají hodnotě její derivace $f'(0)$. Na obrázku 2 vidíme ale podivné chování výrazu (3) v počítačové aritmetice (viz [5]). Hlavní příčinou tohoto jevu je fakt, že při odečítání dvou skoro stejně velkých čísel vzniká velká zaokrouhlovací chyba. Konečnost aritmetiky navíc dělá veškeré výpočty v okolí nuly a „nekonečna" velmi nespolehlivými. Připomeňme, že *strojová přesnost* Ψ je definována vztahem $\Psi = 1^+ - 1$, kde 1^+ je nejmenší číslo větší než 1 v počítačové aritmetice. V našem případě je $\Psi \approx 10^{-19}$ a právě v okolí tohoto bodu už dochází na obrázku 2 k velmi nepřesnému výpočtu derivace, i když nejmenší kladné zobrazitelné číslo x na užitém PC bylo zhruba 10^{-38}.



Obr. 2. Hodnoty podílu $((1+h)-1)/h$ pro $h \to 0$ ve dvojnásobné aritmetice.

Obecně lze říci, že čím vyšší derivaci numericky počítáme, tím dříve (tj. pro větší h) se projeví konečnost počítačové aritmetiky. Pro reálné $x \geqq 0{,}99$ uvažujme racionální funkci (viz [8])

$$g(x) = \frac{4970x - 4923}{4970x^2 - 9799x + 4830} \tag{4}$$

a počítejme numericky její druhou derivaci v bodě 1 pomocí druhé centrální diference, tzn. pomocí podílu

$$\delta_h^2 g(1) = \frac{g(1+h) - 2g(1) + g(1-h)}{h^2}, \tag{5}$$

který se, jak známo, blíží hodnotě $g''(1)$ pro $h \to 0$. Kdybychom nevěděli, že $g''(1) = 94$, těžko bychom z prostředního řádku tabulky 1 zjistili, která ze zde uvedených hodnot vlastně druhou derivaci nejlépe aproximuje. Pro „velká“ h se totiž $g(1-h)$ prudce mění. Pro „malá“ h zase nastává úplná ztráta přesnosti, protože se v čitateli (5) odečítají skoro stejně velká čísla.

Tab. 1. Numerická hodnota $g''(1)$ počítaná pomocí vztahů (5) a (6).

$\delta_h^2 g(1)$	$h = 10^{-2}$	$h = 10^{-3}$	$h = 10^{-4}$	$h = 10^{-5}$	$h = 10^{-6}$	$h = 10^{-7}$
(5)	−91769,95	−2250,2	70,94	93,71	116,42	−151340
(6)	−91769,95	−2250,2	70,79	93,77	94,00	94,00

Vztah (5) lze pomocí (4) upravit na tvar

$$\delta_h^2 g(1) = \frac{94(1 - 70^2 71^2 h^2)}{(1 - 71^2 h^2)(1 - 70^2 h^2)}, \tag{6}$$

který již dává přijatelné numerické výsledky (viz poslední řádek tab. 1). Touto úpravou jsme se zbavili nebezpečného vlivu odečítání skoro stejných čísel při výpočtu $\delta_h^2 g(1)$, tj. pro jednu funkci tvaru (4) v jednom bodě. Často je možné podobně postupovat i v jiných případech, ale není to obecný návod pro všechny situace. Jiná možnost je použít aritmetiku s větším počtem cifer či metodu extrapolace (viz [11]).

4. Numerická integrace

K numerickému výpočtu integrálu

$$I = \int_0^{\pi/2} \frac{\cos x}{\sqrt{\sin x}}\,\mathrm{d}x = \left[2\sqrt{\sin x}\right]_0^{\pi/2} = 2 \tag{7}$$

použijme známý složený Gaussův kvadraturní vzorec

$$I(n) = \frac{h}{2}\sum_{i=1}^{n}\Big(f\big(x_{i-1} + \tfrac{1}{6}(3-\sqrt{3})h\big) + f\big(x_i - \tfrac{1}{6}(3-\sqrt{3})h\big)\Big) \tag{8}$$

pro $f(x) = \cos x/\sqrt{\sin x}$, $x \in (0, \pi/2\rangle$, kde $0 = x_0 < x_1 < \ldots < x_n = \pi/2$ je dělení intervalu $\langle 0, \pi/2\rangle$ na n stejně dlouhých dílků délky $h = \pi/(2n) = x_i - x_{i-1}$, $i = 1, \ldots, n$. Pomalá konvergence chyby $I - I(n)$ (viz druhý sloupec tabulky 2) je způsobena zejména tím, že se funkce f v blízkosti nuly chová jako $x^{-1/2}$. Konvergenci je možno podstatně urychlit tím, že nepříjemnou singularitu $x^{-1/2}$ „odseparujeme". Pišme tedy I ve tvaru

$$I = \int_0^{\pi/2}\left(\frac{\cos x}{\sqrt{\sin x}} - \frac{1}{\sqrt{x}}\right)\mathrm{d}x + \int_0^{\pi/2}\frac{\mathrm{d}x}{\sqrt{x}} \equiv I_1 + I_2 \tag{9}$$

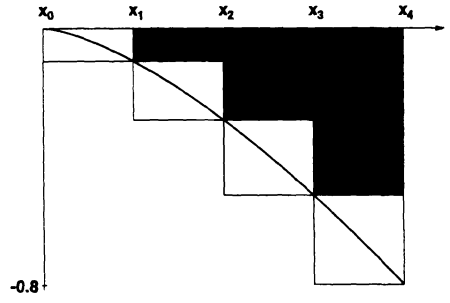
a počítejme jen první integrál I_1 numericky a druhý analyticky. Zřejmě $I_2 = \sqrt{2\pi}$ a nechť $I_1(n)$ jsou hodnoty dané vzorcem (8) pro funkci $f_1(x) = f(x) - x^{-1/2}$, $x \in (0, \pi/2\rangle$, kterou spojitě dodefinujeme vztahem $f_1(0) = 0$. Ze třetího sloupce tabulky 2 vidíme, že chyba $I - \tilde{I}(n) = I - \big(I_1(n) + I_2\big)$ konverguje k nule velmi rychle, např. již pro $n = 1$ je v absolutní hodnotě zhruba rovna chybě $I - I(n)$ pro $n = 65536$, což představuje rozdíl ve spotřebě strojového času o několik řádů.

Tab. 2. Chyby Gaussova kvadraturního vzorce (8) při výpočtu integrálu I dvěma způsoby a aposteriorní odhady pro I.

n	$I - I(n)$	$I - \tilde{I}(n)$	dolní odhad I	horní odhad I
1	4,36 E−1	−1,74 E−3	1,25331	2,50663
2	3,10 E−1	−3,06 E−4	1,65418	2,28084
4	2,19 E−1	−5,31 E−5	1,83478	2,14811
⋮	⋮	⋮	⋮	⋮
65536	1,71 E−3	−2,18 E−15	1,99999	2,00001

Protože je funkce f_1 klesající, můžeme snadno odhadnout integrál I_1 shora i zdola pomocí součtu ploch obdélníků z obrázku 3. Pro přesnou hodnotu integrálu (7) tak celkem dostaneme následující dolní a horní aposteriorní odhad (viz tabulku 2)

$$h\sum_{i=1}^{n} f_1(x_i) + I_2 \leqq I \leqq h\sum_{i=0}^{n-1} f_1(x_i) + I_2.$$



Obr. 3. Horní a dolní aposteriorní odhad integrálu z funkce f_1.

Odhlédneme-li od zaokrouhlovacích chyb, získáme tedy aproximaci hodnoty I a zároveň velikost maximální chyby, jíž se můžeme dopustit.

Některé paradoxní příklady na numerický výpočet integrálů s nehladkými či rychle oscilujícími funkcemi, numerické integrování funkcí na nekonečných intervalech, rekurentní výpočty integrálů apod. lze nalézt v [1, 3, 9, 10, 11]. Potíže vznikají i s vícerozměrnými integrály (např. když má integrační oblast křivočarou hranici). Bohužel neexistuje obecný recept, jak nějaký daný integrál numericky vypočítat, a tak je třeba každou třídu úloh vyšetřovat zvlášť. Často pomůže vhodná substituce, nerovnoměrné dělení, vzorce vyšších řádů, váhové integrační vzorce, Taylorův rozvoj, extrapolace apod. Podobným způsobem jako v (9) lze též odseparovat známý typ singularity nejenom v samotné integrované funkci, ale i v jejích derivacích. Jindy zase pomůže zabývat se inverzní funkcí. Např. při výpočtu integrálu $J = \int_0^1 \sqrt[3]{x}\,\mathrm{d}x$ pomocí (8) pro interval $\langle 0, 1\rangle$ dostaneme poměrně velkou chybu, protože derivace funkce $y(x) = \sqrt[3]{x}$ má v nule singularitu. Uvažujeme-li ale její inverzní funkci, pak vidíme, že $J = 1 - \int_0^1 y^3\,\mathrm{d}y$, a pomocí (8) dostaneme dokonce nulovou diskretizační chybu, protože tento vzorec je přesný pro všechny kubické polynomy.

5. Soustavy lineárních algebraických rovnic

Zabývejme se soustavou

$$Ax = b, \tag{10}$$

kde A je regulární matice typu $n \times n$, x je vektor neznámých a b je zadaný vektor. Potíže s řešením této soustavy vznikají zejména tehdy, je-li A tzv. špatně podmíněná (tj. např. když se koeficienty první rovnice jen málo liší od koeficientů rovnice druhé). Řada příkladů tohoto typu vedoucích k patologickým výsledkům je uvedena v [3, 8, 9, 11, 15]. My se ale pokusíme poukázat na jiné nebezpečí, které souvisí s velikostí paměti počítače a počtem prováděných operací.

Nechť f je funkce definovaná na polyedrické oblasti Ω v eukleidovském prostoru dimenze $d \in \{1, 2, 3\}$. Řešení Poissonovy rovnice

$$-\Delta u = f \quad \text{v} \quad \Omega \tag{11}$$

s jistými okrajovými podmínkami metodou konečných prvků (viz [7, 13, 14]) vede na soustavu (10), jejíž matice je pásová, symetrická a pozitivně definitní. Řešme tuto soustavu klasickou Gaussovou eliminační metodou a metodou sdružených gradientů (viz [11, 13]).

Tab. 3. Nároky na paměť dvou různých metod pro $n \to \infty$.

NÁROKY NA PAMĚŤ POČÍTAČE			
Metoda	$d = 1$	$d = 2$	$d = 3$
Gaussova	$\mathcal{O}(n)$	$\mathcal{O}(n^{3/2})$	$\mathcal{O}(n^{5/3})$
sdružených gradientů	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$

Tab. 4. Počet aritmetických operací dvou různých metod v závislosti na dimenzi.

POČET ARITMETICKÝCH OPERACÍ			
Metoda	$d = 1$	$d = 2$	$d = 3$
Gaussova	$\mathcal{O}(n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^{7/3})$
sdružených gradientů	$\mathcal{O}(n^2)$	$\mathcal{O}(n^{3/2})$	$\mathcal{O}(n^{4/3})$

Z tabulek 3 a 4 vidíme, že Gaussova metoda je vhodná, je-li Ω jednodimenzionální ($d = 1$), a není vhodná pro trojrozměrné oblasti ($d = 3$), kdy potřebujeme řádově $n^{5/3}$ paměťových buněk, zatímco pro metodu sdružených gradientů stačí řádově jen n buněk. Je to důsledek toho, že šíře pásu matice A je postupně $\mathcal{O}(1)$, $\mathcal{O}(n^{1/2})$, $\mathcal{O}(n^{2/3})$ pro $d = 1, 2, 3$ a že se tento pás při eliminaci zaplní obecně nenulovými prvky, kdežto při použití metody sdružených gradientů se A nezmění. Horní odhady uvedené v tabulkách 3 a 4 jsou optimální v tom smyslu, že je nelze zlepšit.

Poznamenejme ještě, že iterační proces metody sdružených gradientů se ukončuje, jakmile je diskretizační chyba metody konečných prvků řádově rovna iterační chybě. Počet aritmetických operací obou metod podstatně závisí na dimenzi d. Pro Gaussovu metodu exponent u n v tabulce 4 roste s rostoucím d, zatímco pro metodu sdružených gradientů klesá. Porovnání obou metod tak opět vychází v neprospěch Gaussovy metody pro $d \geqq 2$. Přesto se Gaussova eliminace těší značné popularitě, pokud nejsme omezeni strojovým časem a pamětí počítače.

6. Vlastní čísla matic

Vlastní čísla $\lambda_1, \ldots, \lambda_n$ čtvercové matice A stupně n jsou kořeny jejího charakteristického polynomu, tj. jsou řešením rovnice

$$\det(\lambda I - A) = 0, \tag{12}$$

kde I je jednotková matice. Je známo, že pro polynom stupně $n > 4$ nelze obecně analyticky vyjádřit jeho kořeny. (Řešit rovnici (12) pro $n = 3, 4$ pomocí Cardanových vzorců je těžkopádné.) Proto se vlastní čísla matic počítají většinou numericky.

Pro reálná čísla ε a α uvažujme matici

$$A = \begin{pmatrix} 0 & \alpha & & & \\ & 0 & \alpha & & \\ & & \ddots & \ddots & \\ & & & 0 & \alpha \\ \varepsilon & & & & 0 \end{pmatrix}, \tag{13}$$

kde všechny ostatní nevyznačené prvky jsou nulové. V tomto případě má (12) tvar

$$\lambda^n - \varepsilon\alpha^{n-1} = 0. \tag{14}$$

Vidíme, že pro $\varepsilon = 0$ jsou všechna vlastní čísla λ_i nulová. Je-li ale $\varepsilon = 10^{-15}$, což je téměř strojová nula, pak např. pro $\alpha = 10$ a $n = 16$ jsou všechna vlastní čísla rovnoměrně rozdělena na jednotkové kružnici se středem v počátku, tj. $|\lambda_i| = 1$ pro $i = 1, \ldots, 16$. Změnou jediného prvku matice A o pouhých 10^{-15} se nám tedy podstatně změní všechna vlastní čísla. Nestability tohoto typu pak činí velké potíže při numerickém výpočtu vlastních čísel zejména nesymetrických matic. V důsledku zaokrouhlování se nám totiž mohou nepatrně pozměnit prvky matice.

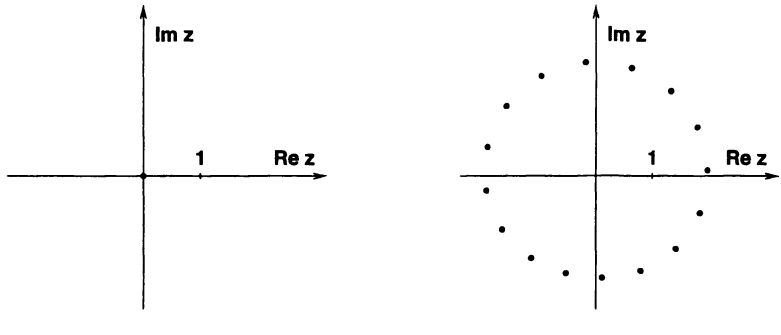
Zabývejme se např. maticí

$$M = QAQ^{-1}, \tag{15}$$

kde A je matice z (13) pro $\varepsilon = 0$, $\alpha = 10$ a $n = 16$ a kde

$$Q = \left(\sqrt{\frac{2}{n+1}} \sin \frac{ij\pi}{n+1}\right)_{i,j=1}^{n}$$

je ortogonální matice (tj. $Q^{-1} = Q^{\top}$), která je navíc symetrická (tj. $Q = Q^{\top}$). Lze ukázat, že vlastní čísla matice M jsou nulová a že všechny její prvky leží v intervalu $(-10, 10)$. Pomocí standardního LR-algoritmu na matici v Hessenbergově tvaru (viz [11, 16]) dostaneme, že takto vypočtená vlastní čísla leží zhruba na kružnici o poloměru 2,031. Při použití známého QR-algoritmu vyjde poloměr dokonce 2,137 — viz obr. 4. Pokud tedy předem nevíme, že matice M vznikla transformací (15), nejsme schopni



a) b)

Obr. 4. a) Vlastní čísla matice M.
b) Vlastní čísla M získaná QR-algoritmem ve dvojnásobné aritmetice.

pomocí výše uvedených algoritmů vypočítat (ve dvojnásobné aritmetice), že M má mnohonásobné vlastní číslo nulu. Poměrně nedávno se zjistilo (viz [5]), že přesnost výpočtu vlastních čísel libovolné matice A je tím horší, čím větší je norma matice $A^\top A - AA^\top$.

7. Kořeny polynomů

Uvažujme Wilkinsonův polynom [11, 15]

$$p(x) = (x+1)(x+2)\dots(x+20) = x^{20} + 210x^{19} + \cdots,$$

jehož reálné kořeny $-1, -2, \dots, -20$ jsou oproti předchozímu příkladu (srov. (14)) dobře separovány. Předpokládejme, že pouze jeden koeficient u x^{19} zmenšíme o 2^{-23}. Tato malá změna způsobí obrovské změny v poloze kořenů; některé z nich pak vyjdou dokonce komplexní, „daleko" od reálné osy — např. $-16{,}731 \pm 2{,}813\mathrm{i}$, $-13{,}992 \pm 2{,}519\mathrm{i}$. Polynom, u něhož malá změna v koeficientech může způsobit velkou změnu v kořenech, se nazývá *špatně podmíněný*. Při zadávání polynomu do počítače se jeho koeficienty obecně zaokrouhlují. Proto je velice obtížné počítat numericky kořeny špatně podmíněných polynomů.

8. Počáteční úlohy pro obyčejné diferenciální rovnice typu stiff

Přestože teorie numerických metod pro řešení úloh s počátečními podmínkami je v současné době vypracována už do značných podrobností (viz např. [4, 14]), mohou v konkrétní situaci nastat značné praktické obtíže. Ilustrujme je na následujícím jednoduchém příkladě, jehož přesné řešení je $y(x) = \mathrm{e}^{-x}$. Máme určit hodnotu řešení y diferenciální rovnice

$$y'' + (1 + 10^6)y' + 10^6 y = 0$$

s počáteční podmínkou $y(0) = 1$, $y'(0) = -1$, neboli, což je totéž, řešit soustavu diferenciálních rovnic

$$\begin{aligned} y' &= z, \\ z' &= -10^6 y - (1 + 10^6)z \end{aligned} \tag{16}$$

s počátečními podmínkami $y(0) = 1$, $z(0) = -1$, v bodě $x = 1$ s chybou nepřesahující 10^{-3}.

Řešme soustavu (16) explicitní Eulerovou metodou. To v našem případě znamená počítat přibližné řešení y_n a z_n (aproximující hodnoty funkcí y a z v bodě $x = nh$, kde h je krok metody, který si volíme) z rekurencí

$$\begin{aligned} y_{n+1} &= y_n + hz_n, \\ z_{n+1} &= z_n + h[-10^6 y_n - (1 + 10^6)z_n], \quad n = 0, 1, \dots \end{aligned} \tag{17}$$

s počátečními podmínkami $y_0 = 1$ a $z_0 = -1$. Snadno se zjistí, že platí

$$y_n = (1-h)^n, \quad z_n = -(1-h)^n.$$

Odtud dostáváme při $h = \frac{1}{200}$ jako aproximaci hodnoty $y(1) = \exp(-1) = 0{,}36788\ldots$ číslo $y_{200} = 0{,}36695\ldots$, což splňuje požadovanou přesnost. Provedeme-li však při $h = \frac{1}{200}$ výpočet rekurentně podle vzorců (17) (a tak musíme v obecném případě postupovat, neboť jen výjimečně se podaří vyřešit soustavu typu (17) analyticky), dostaneme už po pouhých 20 krocích jako aproximaci čísla $\exp(-0{,}1)$ číslo $y_{20} = -4{,}220 \cdot 10^{44}$, tedy zcela nesmyslný výsledek.

Užijeme-li místo metody (17) tzv. implicitní Eulerovu metodu, která je v případě soustavy diferenciálních rovnic (16) daná rekurencemi

$$\begin{aligned} y_{n+1} &= y_n + hz_{n+1}, \\ y_{n+1} &= z_n + h[-10^6 y_{n+1} - (1+10^6)z_{n+1}], \quad n = 0, 1, \ldots \end{aligned}$$

s počátečními podmínkami $y_0 = 1$ a $z_0 = -1$, dostaneme při $h = \frac{1}{200}$ naprosto uspokojivou aproximaci $y_{200} = 0{,}36879\ldots$

Řešíme-li na druhé straně obyčejnou (explicitní) a implicitní Eulerovou metodou soustavu

$$\begin{aligned} y' &= z, \\ z' &= 2y + z \end{aligned}$$

s počátečními podmínkami $y(0) = 1$ a $z(0) = -1$, která má stejné řešení jako soustava (16) s příslušnými počátečními podmínkami, dostaneme při $h = \frac{1}{200}$ jako aproximaci řešení v bodě $x = 1$ čísla $0{,}36695\ldots$ a $0{,}36879\ldots$, tj. v obou případech zhruba stejně kvalitní aproximaci.

Obě užité metody jsou konvergentní v tom smyslu, že volbou dostatečně malého integračního kroku lze docílit libovolné přesnosti aproximace. V případě soustavy (16) a explicitní Eulerovy metody integrační krok $h = \frac{1}{200}$ nebyl zřejmě ještě „dostatečně" malý, abychom dostali výsledky, které mají vůbec nějaký smysl. Snadno se zjistí, že tento požadavek splňuje teprve integrační krok h, pro nějž platí $h < 2 \cdot 10^{-6}$. Užitím tohoto integračního kroku bychom sice získali přesnější aproximaci než výše uvedenou — to jsme však ani nechtěli — zato výpočetní práce vzroste zhruba 2500krát.

Soustava (16) je typickým představitelem tzv. „stiff" diferenciální soustavy, což je taková soustava diferenciálních rovnic, jejíž matice (resp. Jacobián v nelineárním případě) má některá vlastní čísla se zápornou reálnou částí v absolutní hodnotě obrovská ve srovnání s jinými. Některé složky řešení se tedy neobyčejně rychle tlumí. Užití nevhodné metody pak vyžaduje nutnost dobré aproximace i těchto složek, které nás vlastně ani nezajímají, a tedy nutnost brát neúnosně malý integrační krok. Kvalitní soubor programů pro řešení úloh s počátečními podmínkami musí být schopen toto chování odhalit a reagovat na ně.

9. Parciální diferenciální rovnice

Demonstrujme na následujícím příkladě, jak v principu správná, ale poněkud povrchní heuristická úvaha může vést ke katastrofálním důsledkům. Řešme diferenciální rovnici

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \qquad 0 < x < 0, \quad 0 < t < T \tag{18}$$

(tzv. rovnice pro vedení tepla) s počáteční podmínkou

$$u(x, 0) = \sin \pi x$$

a okrajovými podmínkami

$$u(0, t) = u(1, t) = 0$$

metodou sítí. Přesné řešení má tvar $u(x, t) = \mathrm{e}^{-\pi^2 t} \sin \pi x$.

Algoritmus jedné z variant metody sítí je dán vztahy

$$\begin{aligned} u_j^{(\ell+1)} &= \beta u_{j-1}^{(\ell)} + (1 - 2\beta) u_j^{(\ell)} + \beta u_{j+1}^{(\ell)}, \quad j = 1, \ldots, n-1, \quad \ell = 0, \ldots, r-1, \\ u_j^{(0)} &= \sin(j\pi/n), \quad j = 1, \ldots, n-1, \\ u_0^{(\ell)} &= u_n^{(\ell)} = 0, \quad \ell = 0, \ldots, r, \end{aligned} \tag{19}$$

kde n, r jsou přirozená čísla, $k = T/r$, $h = 1/n$,

$$\beta = k/h^2 \tag{20}$$

a číslo $u_j^{(\ell)}$ značí aproximaci přesného řešení v bodě $x = jh$ a $t = \ell k$. Rovnice (19) vznikly tak, že jsme v diferenciální rovnici (18) nahradili derivace diferenčními podíly a příslušné chyby jsme zanedbali. Tyto chyby jsou úměrné výrazu $k + h^2$, proto jsme zvolili k řádově tak velké jako h^2. Přibližné řešení a jeho chyba v bodě $x = \frac{1}{2}$ jsou uvedeny pro dvě volby parametrů k a h v tabulce 5. Z tabulky vidíme, že jemnější síť dává naprosto nesmyslné výsledky, zatímco druhá alternativa volby k a h je vcelku přijatelná. Ve skutečnosti je k uspokojivému chování chyby nejen třeba, aby k a h byly malé, ale zároveň ještě musí platit pro poměr (20) nerovnost $0 < \beta \leqq \frac{1}{2}$.

10. Limita funkce

V posledním příkladu si zkuste sami zodpovědět otázku: Proč při výpočtu limity

$$\lim_{x \to 0} \frac{1 - \cos x}{x^2} = \tfrac{1}{2}$$

na počítači dostanete nulu?

Tab. 5. Chování chyby při řešení rovnice pro vedení tepla může být pro menší diskretizační krok h podstatně horší, aniž by to bylo způsobeno zaokrouhlovacími chybami.

	$k = 10^{-4},\ h = 10^{-2}$		$k = 10^{-4},\ h = 2 \cdot 10^{-2}$	
t	přibl. řeš.	chyba	přibl. řeš.	chyba
0,0008	0,9921322	−0,0000032	0,9921341	−0,0000013
0,0009	0,9911531	−0,0000036	0,9911552	0,0000014
0,0010	0,9901749	−0,0000040	0,9901773	−0,0000016
⋮				
0,0098	$-4{,}32 \cdot 10^{27}$	$-4{,}32 \cdot 10^{27}$	0,9077938	−0,0000144
0,0099	$6{,}43 \cdot 10^{27}$	$6{,}43 \cdot 10^{27}$	0,9068981	−0,0000146
0,0100	$-2{,}03 \cdot 10^{26}$	$-2{,}03 \cdot 10^{26}$	0,9060033	−0,0000147
⋮				
0,0498	$-8{,}5 \cdot 10^{219}$	$8{,}5 \cdot 10^{219}$	0,6116548	−0,0000494
0,0499	$-2{,}5 \cdot 10^{220}$	$-2{,}5 \cdot 10^{220}$	0,6110513	−0,0000495
0,0500	$7{,}6 \cdot 10^{220}$	$7{,}6 \cdot 10^{220}$	0,6104485	−0,0000496
⋮				
1,0000	∞	∞	0,0000516	−0,0000001

Závěrečné poznámky

V předešlých příkladech jsme viděli různé efekty, které vedly při počítání k neúspěchu. Můžeme je rozdělit na několik skupin:

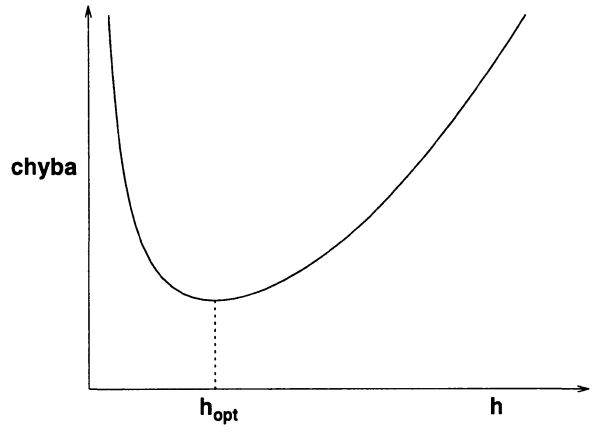
a) nevhodné použití metody — příklad 2, 4, 8, 9,

b) vliv zaokrouhlovacích chyb a nestabilita úlohy — příklad 1, 3, 6, 7, 8, 10,

c) potíže, které mohou vzniknout při velkém rozsahu úlohy — příklad 5.

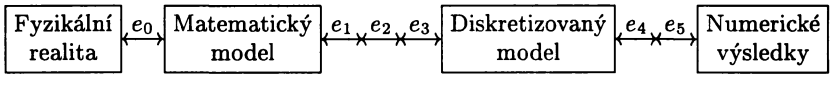
K bodu a) je třeba říci především to, že je nutno se s použitou metodou důkladně seznámit. Vhodný pramen poučení přitom může být rozhodující. Neočekávanému vlivu zaokrouhlování (viz b)) se asi nikdy nelze zcela vyhnout, protože nebezpečí jejich vzniku může být skryto už v samotné úloze.

Obecný graf chyby v závislosti na diskretizačním parametru h, který řídí přesnost výpočtu, je na obrázku 5. To, že graf v okolí 0 klesá, je způsobeno zaokrouhlovacími chybami. Proto není vhodné volit velikost parametru h tak, aby odpovídal klesající části grafu. Čím je h menší, tím více operací musíme provádět. Chyba má obvykle tvar $C_1 h^p + C_2 h^{-1}$ pro $C_1, C_2, p > 0$ — viz [14, str. 30]. Při počítání je stálý konflikt mezi snahou o přesnost výpočtu a o jeho rychlost, tj. o počet prováděných operací. Je jasné, že požadavek na přesnost má přednost, ale pak výpočet může ztroskotat na množství operací, z čehož dále hrozí nebezpečí vlivu zaokrouhlování, nebo na enormních požadavcích na paměť počítače. Proto je důležité zabývat se i otázkou správné interpretace získaných numerických výsledků.



Obr. 5. Chování velikosti chyby při numerických výpočtech v závislosti na diskretizačním parametru h.

Při řešení složitějších problémů než modelových úloh uvedených v odstavcích 1–10 dochází ke vzniku celé řady nejrůznějších chyb, které se bohužel vzájemně neruší, ale jejich chování se řídí statistickými zákony.



Obr. 6. e_0 – chyba modelu, e_1 – chyba diskretizace, e_2 – chyba aproximace zakřivené hranice, e_3 – chyba numerické integrace, e_4 – iterační chyba, e_5 – zaokrouhlovací chyby.

Abychom stručně naznačili, jaké typy chyb mohou vzniknout, uvažujme numerické řešení Poissonovy rovnice (11) v ohraničené oblasti $\Omega \subset R^2$ s jistými okrajovými podmínkami. Tato rovnice přibližně popisuje např. rozložení gravitačního či elektrického potenciálu. Odtud vzniká chyba modelu e_0 (viz obr. 6), jejíž studium není předmětem numerické matematiky. Pokud neznáme analytické řešení uvažovaného problému, je třeba použít nějakou přibližnou metodu — například metodu konečných prvků (viz [7]). Ta umožňuje nekonečně dimenzionální problém převést na konečně dimenzionální. Tím se ovšem dopustíme tzv. chyby diskretizace e_1. Pokud je hranice oblasti Ω zakřivená, je třeba ji vhodným způsobem aproximovat, a tak vznikne chyba e_2. V metodě konečných prvků je ale zapotřebí počítat integrály, v nichž vystupuje funkce f z (11), přes jisté podoblasti (prvky), což se většinou neobejde bez numerické integrace, která způsobí další chybu e_3. Přibližné řešení rovnice (11) s příslušnými okrajovými podmínkami se pak převádí na řešení soustavy lineárních algebraických rovnic. Pokud ji budeme řešit iteračně, dopustíme se chyby e_4. Konečně celý výpočet je samozřejmě provázen zaokrouhlovacími chybami e_5. Z praktického hlediska nemá smysl dělat např. chybu e_1 co možno nejmenší, pokud adekvátně nezmenšíme i další chyby e_i.

Při řešení parciálních diferenciálních rovnic však mohou vzniknout ještě další chyby např. z aproximace počátečních či okrajových podmínek, aproximace různých koeficientů, nelinearit apod. Nejdůležitější je ale umět citlivě odhadnout, která chyba má dominantní postavení. K tomu nám slouží různé aposteriorní odhady — viz např. [7, str. 136].

Praktické zkušenosti ukazují, že při aproximaci nekonečně dimenzionálních problémů konečně dimenzionálními je žádoucí zachovat co nejvíce vlastností původního problému (např. splnění principu maxima, podmínek rovnováhy, podmínky nestlačitelnosti, ...). Stabilita řešení (v závislosti na počátečních podmínkách, pravé straně apod.) vyšetřovaného nekonečně dimenzionálního problému je nutnou (ne vždy postačující!) podmínkou pro získání „dobrého" numerického řešení. To ovšem závisí zejména na volbě vhodné numerické metody. Získané výsledky závisejí samozřejmě také na způsobu naprogramování i použitém počítači. Dnes je zapotřebí, aby program sám sledoval a hodnotil výpočet už v jeho průběhu a upozornil na nebezpečí ztráty přesnosti. Kromě toho by měl řídit i rozsah výpočtu.

Numerická matematika nám podává nezbytný teoretický podklad k tomu, aby numerické počítání mohlo být úspěšné.

Literatura

[1] I. Babuška, M. Práger, E. Vitásek: *Numerical Processes in Differential Equations.* John Wiley & Sons, Ltd., London 1966.
[2] A. Doktor: Rozhledy mat.-fyz. *70* (1992), 223–229.
[3] G. E. Forsythe: Amer. Math. Monthly *77* (1970), 931–956.
[4] P. Henrici: *Discrete Variable Methods in Ordinary Differential Equations.* John Wiley & Sons, New York 1962.
[5] F. Chatelin, V. Frayssé: *Qualitative Computing.* Thomson – CSF, 1993.
[6] J. Chleboun: Vesmír *72* (1993), 507–508.
[7] M. Křížek: PMFA *37* (1992), 129–140.
[8] U. Kulisch, W. L. Miranker: SIAM Review *28* (1986), 1–40.
[9] J. Mikloško: *„Patologické" javy v numerickej matematike.* Sborník konference software a algoritmy numerické matematiky, JČMF, VŠSE Plzeň 1980, 201–232.
[10] W. H. Press et al: *Numerical Recipes.* Cambridge University Press, 1986.
[11] A. Ralston: *Základy numerické matematiky.* Academia, Praha 1978.
[12] I. A. Stegun, M. Abramowitz: SIAM *4* (1956), 207–219.
[13] E. Vitásek: *Numerické metody.* SNTL, Praha 1987.
[14] E. Vitásek: *Základy teorie numerických metod pro řešení diferenciálních rovnic.* Academia, Praha 1994.
[15] J. H. Wilkinson: *Rounding Errors in Algebraic Processes.* Prentice-Hall, New York 1963.
[16] J. H. Wilkinson, C. Reinsch: *Handbook for Automatic Computation: Linear Algebra.* Springer, Berlin 1971.