1. Základní pojmy

1.1. Matematický model

- formulace reálného problému pomocí matematiky

1.2. Matematická úloha

→ numerická úloha

- vztah mezi konečným počtem vstupních a výstupních dat
- konečnost je důležitá z hlediska počítačové řešitelnosti

1.3. Numerická metoda, algoritmus

- způsob, jak vypočítat formulovanou úlohu pomocí počítače

→ zabývá se numerická matematika

1.4. Korektní úloha

Úloha je korektní na dvojici prostorů (B_1, B_2), když

a) ke každému $x \in B_1 \ \exists! \ y \in B_2$

b) toto řešení spojitě závisí na vst. datech, tj. když

$x_n \to x \ \wedge \ U(x_n) = y_n$

potom

$U(x_n) \to U(x) = y_n$

1.5. Podmíněnost úlohy

- úloha je dobře podmíněná, pokud malá změna vst. dat způsobí relativně malou změnu dat výstupních

1.6. Číslo podmíněnosti

$$C_p = \frac{\frac{\|\Delta y\|}{\|y\|}}{\frac{\|\Delta x\|}{\|x\|}} \approx \frac{\delta(y)}{\delta(x)} \quad \left(= \frac{\text{rel. chyba na výst.}}{\text{rel. chyba na vst.}} \right)$$

Zpravidla: $C_p \approx 1$ – dobře podmíněná úloha

$C_p \gtrsim 100$ – špatně podmíněná úloha

1.7. Podmíněnost algoritmu

= citlivost na poruchu v datech (viz [1.5], [1.6])

1.8. Stabilita algoritmu

- stabilita je daná:
  - a) podmíněností (viz [1.7.])
  - b) numerickou stabilitou
    - ↳ = citlivost na vliv zaokrouhlovacích chyb

1.9. Příklady

1.9.1. Špatná podmíněnost

$x^2 - 4x + 4 = 0 \quad ; \quad x_1 = x_2 = 2$

$x^2 - 4x + 3{,}9999 = 0 \; ; \; x_1 = 2{,}01 \;\; x_2 = 1{,}99$

$$\left.\begin{array}{l} \text{porucha vstupu} = 1 \cdot 10^{-4} \\ \text{porucha výstupu} = 1 \cdot 10^{-2} \end{array}\right] \Rightarrow$$ změna vstupu vyvolala 100x větší změnu výstupu

1.9.2. Numerická nestabilita

- při jednotlivých iteracích může špatnou volbou algoritmu dojít k tomu, že chyba výsledku přeroste jeho hodnotu

$$J_n = \int_0^1 \frac{x^n}{x+5}\,dx \qquad n = 0, 1, 2, \ldots$$

$$\int_0^1 x^{n-1}\,dx = \int_0^1 \frac{x^n}{x+5} + 5 \cdot \int_0^1 \frac{x^{n-1}}{x+5} \longrightarrow J_n = -5 \cdot J_{n-1} + \frac{1}{n}$$

→ počítáme-li ve 3D:

$J_0 \approx 0{,}182 \quad (\ln \frac{6}{5})$

$J_1 \approx 0{,}090$

$J_2 \approx 0{,}050$

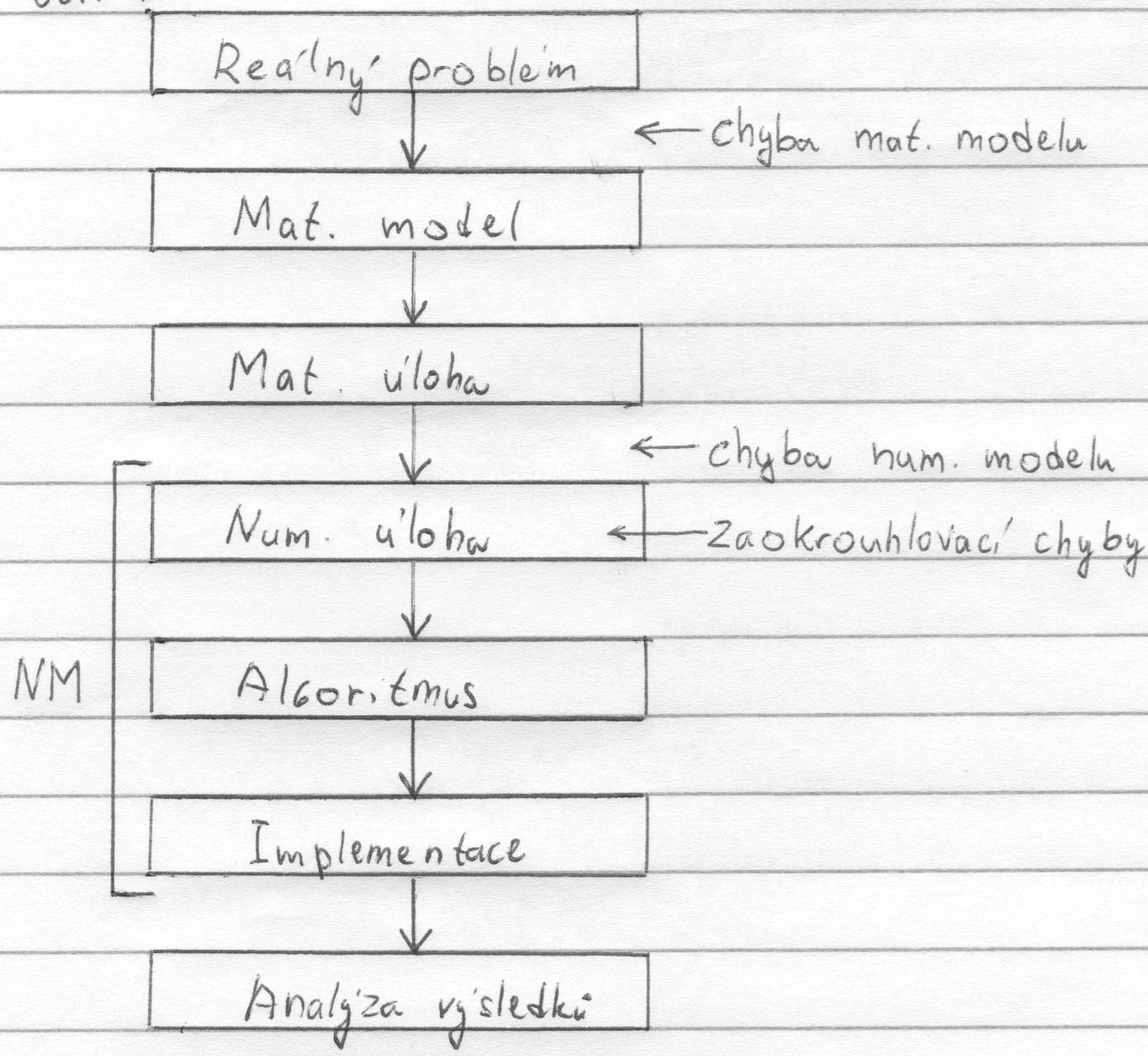
$J_3 \approx 0{,}083$

$J_4 \approx -0{,}165$ – absurdní

- příčina tkví v násobení, jelikož společně s výsledkem předchozí operace násobíme i jeho chybu
- řešením je stanovit hodnotu J_n pro dostatečně velké n a provést výpočet opačně, tj.

$$J_{n-1} = \frac{-J_n}{5} + \frac{1}{5n}$$

Obr. 1.


Reálný problém
chyba mat. modelu
Mat. model
Mat. úloha
chyba num. modelu
Num. úloha
zaokrouhlovací chyby
NM
Algoritmus
Implementace
Analýza výsledků

2. Řešení nelineárních rovnic a jejich soustav

2.1. Metoda prosté iterace

- převedeme rovnici do iteračního tvaru:

$$x = \varphi(x)$$

- x se snažíme (v zájmu rychlosti konvergence, respektive obecně konvergence) vyjádřit z nejvyšší mocniny
- hledáme geometrický průsečík fcí $y(x)$ a $\varphi(x)$

2.1.1. Podmínka konvergence

Předpokládejme, že $\varphi(x)$ je na $I := \langle a; b \rangle$ spojitá a platí

a) $\forall x \in I : \varphi(x) \in I$
   - φ zobrazuje sama do sebe

b) $\exists q \in (0;1) : |\varphi(x) - \varphi(x)| \leq q \cdot |x - y| \quad \forall x, y \in I$
   - kontrakce φ (viz [2.1.2])

2.1.2. Kontrakce

$$\left| \frac{\varphi(x_{k+1}) - \varphi(x_k)}{x_{k+1} - x_k} \right| < 1 \quad \Rightarrow \quad |\varphi'(x_k)| < 1$$

2.1.3. Odhad chyby

$$\exists q \in \langle 0; 1) : |\varphi'(x)| \leq q \quad \forall x \in I$$

$$|x_k - x^*| = |\varphi_{(k-1)} - \varphi(x^*)| \leq q\, |x_{k-1} - x^*| \leq q \cdot |x_k - x_{k-1}| + q\, |x_k - x^*|$$

$$\boxed{|x_k - x^*| \leq \frac{q}{1-q} \cdot |x_k - x_{k-1}|}$$

2.1.3.1. Příklad

$$\underbrace{x^2 - 3x - 5 = 0}_{} \;\to\; x = \sqrt{3x+5} \;\to\; \varphi'(x) = \frac{3}{2} \cdot \frac{1}{\sqrt{3x+5}} \qquad \underbrace{I = \langle 0; 10 \rangle \quad \varepsilon = 0{,}01}_{}$$

$$q = \max_{x \in \langle 0; 10 \rangle} \varphi'(x) = \varphi'(0) \doteq 0{,}67$$

$$|x_5 - x^*| \leq \frac{0{,}67}{1 - 0{,}67} \cdot \overbrace{|x_5 - x_4|}^{\leq 0{,}01 = \varepsilon} \leq 0{,}02$$

$$x_5 = 4{,}1971 \approx x^*$$

2.1.4. Rychlost konvergence

- Je-li φ dostatečně hladká:

$$\varphi(x) = \varphi(x^*) + \varphi'(x^*)\cdot(x-x^*) + \frac{\varphi''(x^*)}{2!}\cdot(x-x^*)^2 + \frac{\varphi'''(\xi)}{3!}\cdot(x-x^*)^3$$

$\varphi(x_{k-1}) = x_k$:

$$x_k - x^* = \varphi'(x^*)\cdot(x_{k-1}-x^*) + \frac{\varphi''(x^*)}{2}(x_{k-1}-x^*)^2$$

a) $\varphi'(x^*) \neq 0$: $r=1$

b) $\varphi'(x^*) = 0$; $\varphi''(x^*) \neq 0$: $r=2$

2.2. Newtonova metoda

$$f(x) = f(x_0) + f'(x_0)\cdot(x-x_0) + \underbrace{\frac{f''(\xi)}{2}(x-x_0)^2}_{\text{chyba}}$$

$$f(x_0) + f'(x_0)\cdot(x-x_0) = 0$$

$$x = x_0 - \frac{f(x_0)}{f'(x_0)} \rightarrow \boxed{x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}} \quad ; k = 0,1,2,\dots$$

$|x_{k+1} - x_k| < \varepsilon$

zast. podmínka

- zpřesňující metoda, očekává a) na $\langle a,b\rangle$ leží jediný kořen x^*
  b) x_0 je relativně blízko x^*

2.2.1. Rychlost konvergence

$\varphi'(x^*) = 0$: $r=2$

$\varphi'(x^*) \neq 0$:

$$\varphi'(x) = 1 - \frac{(f'(x))^2 - f(x)\cdot f''(x)}{(f'(x))^2}$$

$$\varphi'(x^*) = 1 - \frac{f'(x^*)}{(f'(x^*))^2} \Rightarrow r=0$$

3. Přímé metody pro řešení SLAR

3.1. Gaussova eliminační metoda

- princip je převést matici na stupňovitý tvar
- nejjednodušší (resp. nejrychlejší) metoda je určit multiplikátory každého řádku dle pivotního prvku, následně řádky ~~vynásobit jejich~~ sečíst se součinem jejich multiplikátoru a prvního řádku a postup opakovat až do dosažení stupňovitého tvaru
- problém: metoda není obecně realizovatelná pro každou regulární matici, jelikož pivotní prvek může být 0, s čímž metoda nepočítá!

G. m. je realizovatelná, pokud:

a) matice A je ostře diagonálně dominantní

nebo b) matice A je symetrická a poz. definitní

3.1.1. Pivotace

= výběr hlavního prvku

- podle typu pivotace vždy vybereme největší (v absolutní hodnotě) prvek a pomocí prohazování řádků jej umístíme na vhodné místo
- typy pivotace:
  - řádková
  - sloupcová
  - úplná (kombinovaná)
- použitím libovolného typu pivotace dosáhneme řešitelnosti G. metodou pro libovolnou reg. matici

3.1.2. Zpětný chod

- po získání horní trojúhelníkové matice dosazujeme hodnoty neznámých od konce a tím postupně získáme hodnotu všech
- počet operací = ~~$O(n^2)$~~ $\frac{N \cdot (N-1)}{2}$ dělení + $\frac{N^3-N}{3}$ násobení

$\frac{N^3-N}{3}$ sčítání

3.2. LU rozklad

a) $A = L \cdot U$ - provedeme rozklad na dolní Δ matici L a horní ∇ matici U

$$L = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} \quad U = \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}$$

b) vyřešíme soustavu $L \cdot y = b$; získáme y

c) vyřešíme soustavu $U \cdot x = y$; získáme řešení původní soustavy

- výhody především, pokud máme více soustav se stejnou maticí soustavy
- počet operací: $\frac{N^3}{3} + O(n^2)$ násobení, sčítání

$\frac{N \cdot (N-1)}{2}$ dělení

$\left(N^2 \quad \text{- opakované řešení} \right)$

3.3. Choleského rozklad

- metoda rozložení symetrické, pozitivně definitní čtv. matice na dvě stejné (po transpozici) matice, jejichž součin dává původní matici A:

$$A = L \cdot L^T \quad \text{popř.} \quad A = U^T \cdot U, \text{ kde } U = L^T$$

3.3.1. Výpočet inverzní matice

$$A^{-1} = (L^{-1})^T \cdot L^{-1}$$

3.3.2. Řešení SLAR

$L \cdot y = b$

$L^T \cdot x = y$

3.3.3. Algoritmus rozkladu

- počítáme po sloupcích (lze i po řádcích)

$$a_{11} = l_{11} \cdot l_{11} \quad \rightarrow \quad l_{11} = \sqrt{a_{11}}$$

$$a_{21} = l_{21} \cdot l_{11} \quad \rightarrow \quad l_{21} = a_{21} / l_{11}$$

- obecně:

$$a_{kk} = \sum_{i=1}^{k} l_{ki}^2 \quad \rightarrow \quad l_{kk} = \sqrt{a_{kk} - \sum_{i=1}^{k} l_{ki}^2}$$ pro prvky na diagonále

$$a_{rc} = \sum_{i=1}^{c} l_{ri} \cdot l_{ci} \quad \rightarrow \quad l_{rc} = \frac{a_{rc} - \sum_{i=1}^{c-1} l_{ri} \cdot l_{ci}}{l_{cc}}$$

3.4. Existence a jednoznačnost trojúhelníkového rozkladu

- LU rozklad (a tedy i Choleského rozklad) existuje, pokud je matice regulární a zároveň ostře diagonálně dominantní nebo symetrická a pozitivně definitní

- V závislosti na volbě diagonálních prvků matice L lze získat různé LU rozklady matice - rozklad je volbou prvků jednoznačně určen

- GEM vlastně získáme matici U => řešitelnost LU rozkladem je z hlediska existence ekvivalentní řešitelnosti pomocí GEM

3.5. Stabilita trojúhelníkového rozkladu

- při ručním výpočtu zapisujeme do tabulky už výsledky, čímž se vliv zaokrouhlení částečně omezí

3.6. Soustavy se speciální maticí

Speciální matice má alespoň jednu z násl. vlastností:

a) symetričnost a pozitivní definitnost

b) diagonální dominantnost

c) pásovost

3.6.1. Symetrické a pozitivně definitní matice

Je-li A sym. a poz. def. matice, potom ∃! horní ▽ matice U s kladnými diagonálními prvky taková, že

$$A = U^T \cdot U$$

viz [3.3]

3.6.2. Pásové matice

Lze-li LU rozklad provést, pak

$i < j < i-p \; : \; l_{ij} = 0 \quad \wedge \quad j < i < j-p \; : \; u_{ij} = 0$

což značí, že L a U budou pásové.

- dochází k velké úspoře operací, zejména je-li matice tří diagonální

3.6.3. Symetrické pozitivně definitní pásové matice

- velké soustavy řešeny iteračními metodami
- výhodné je použít LU rozklad či jinou eliminační metodu

4. Iterační metody pro SLAR

4.1. Jacobiho metoda

- ze všech rovnic vyjádříme i-tou neznámou

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} \cdot x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} \cdot x_j^{(k)} \right) \quad i = 1, 2, \dots, n$$

$$k = 0, 1, \dots, n$$

- iterujeme za použití výše uvedených vzorců, jako zastavovací podmínku používáme $\|x^{(k)} - x^{(k-1)}\| \leq \varepsilon$, přičemž je vhodné použít např. maximovou normu (není nezbytné)

4.1.1. Iterační matice

$$A = M + D + N$$

$$(M + D + N) \cdot x = b$$

$$D x^{(k+1)} = -(M + N) x^{(k)} + b$$

$$H_j = -D^{-1} \cdot (M + N)$$

4.2. Gauss-Seidelova metoda

- podobné jako Jacobiova, ale dosazujeme vždy nejnovější hodnoty:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij} \cdot x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} \cdot x_j^{(k)} \right) \quad i = 1, 2, \dots, n$$

$$k = 0, 1, \dots, n$$

4.2.1. Iterační matice

$$(M + D + N) \cdot x = b$$

$$(M + D) \cdot x^{(k+1)} = -N \cdot x^{(k)} + b$$

$$H_S = -(M + D)^{-1} \cdot N$$

4.3. SOR

=relaxační metoda

-jako GS, ale přidává relaxační parametr ω:

$$x_i^{(k+1)} = \frac{\omega}{a_{ii}} \cdot \left(b_i - \sum_{j=1}^{i-1} a_{ij} \cdot x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} \cdot x_j^{(k)}\right) + (1-\omega) \cdot x_i^{(k)}$$

- pro $\omega = 1$ je tento vzorec stejný jako pro GS
- ω vybíráme z intervalu $(0;2)$ ($|1-\omega| > 1$ nesmí nastat!) tak, aby byla konvergence co nejrychlejší

4.3.1. Iterační matice

$$H_\omega = (D + \omega M)^{-1} \cdot [(1-\omega)D - \omega N]$$

4.4. Podmínky konvergence

4.4.1. Nutná a postačující podmínka

Iterace $x^{(k+1)} = H \cdot x^{(k)} + g$ konvergují pro libovolné $x^{(0)}$ právě když

$$\varrho(H) = \max_i |\lambda_i(H)| < 1$$

kde $\varrho(H)$ je spektrální poloměr matice H a $\lambda_i(H)$ jsou vlastní čísla této matice.

4.4.2. Postačující podmínka

Pokud $\|H\| \leq q < 1$, potom $\{x^{(k)}\}_{k=0}^{\infty}$ konverguje při volbě libovolného $x^{(0)}$

6. Metody pro řešení úloh na vlastní čísla

~~6.1. Částečný vs. úplný problém vl. čísel~~

6.1.~~X~~ Částečný problém vl. čísel

- hledáme vlastní číslo matice A s největší absolutní hodnotou = tzv. dominantní číslo
- předpokládáme, že:
  - a) matice A má n lineárně nezávislých vektorů
  - b) dominantní vl. číslo je vedinné:

$$\lambda_1 > \lambda_2 \geq \lambda_3 \geq \dots$$

6.1.1. Mocninná metoda

- předpoklady viz [6.1]
- sestrojíme posloupnost $\{y^{(k)}\}$ pomocí

$$y^{(k)} = A \cdot y^{(k-1)} \quad ; \; k = 1, 2, \dots$$

- pro dostatečně velká k bude

$$\lambda_1 = \lim_{k \to \infty} \frac{y^{(k+1)}}{y^{(k)}}$$

aproximovat vlastní číslo λ_1

- k zastavení procesu používáme podmínku $|\lambda_1^{(k)} - \lambda_1^{(k-1)}| < \delta$ kde δ je předem stanovená hranice (zpravidla)
- z důvodu nezabírání příliš velkého prostoru v paměti počítače používáme normalizaci – pokud některá složka vektoru $y^{(k)}$ přeroste určitou hodnotu (zpravidla 100), vydělíme touto složkou (popř. nejvyšší ze složek v abs. hodnotě) složky ostatní a složku nahradíme jedničkou
- počet operací: řádově n^2 na jednu iteraci

6.1.2. Metoda Rayleighova podílu

- oproti předpokladům mocninné metody (viz [6.1]) je zde navíc předpoklad, že

  A je symetrická

- iterace probíhají analogicky k mocninné metodě (viz [6.1.1]), ale k výpočtu používáme vzorec

$$\lambda_1 = \lim_{k \to \infty} \frac{y^{(k-1)T} \cdot y^{(k)}}{y^{(k-1)T} \cdot y^{(k-1)}}$$

který pro dostatečně velká k aproximuje vl. číslo λ_1 a to až dvakrát rychleji než vzorec mocninné metody (viz [6.1.1])

6.2. Úplný problém vlastních čísel

a) metody využívající charakteristický polynom

b) metody využívající vztah podobnosti matic:

6.2.1. Metoda LU rozkladu

1) rozložíme A_k na $L_k \cdot U_k$

2) spočítáme $A_{k+1} = U_k \cdot L_k$

3) iterujeme, než dostaneme horní $\triangle$ matici

- získaná matice má vlastní čísla původní matice na diagonále

6.3. Singulární rozklad matice

K $\mathbb{R}$ matici A typu $m \times n$, $m \geq n$, existují takové ortogonální matice U $(n \times m)$ a V $(n \times n)$ že prvky σ matice $\Sigma = U^T \cdot A \cdot V$ $(m \times n)$ mají vlastnosti:

a) $\sigma_{ij} = 0$; $i \neq j$

b) $\sigma_{ij} = \sigma_i \geq 0$

– číslům σ_i se říká singulární čísla matice

Tady by toho přece jen mohlo být více...

7. Aproximace funkcí

7.1. Základní úlohy

a) výběr typu aproximující funkce $g(x)$

b) výběr kriteria pro vyjádření blízkosti funkcí $f(x)$ a $g(x)$

7.2. Aproximace Taylorovým polynomem

– aproximujeme T. polynomem n-tého stupně – funkce musí mít derivaci n-tého stupně!

$$T_n(x) = f(x_0) + \frac{f'(x_0)}{1} \cdot (x - x_0) + \frac{f''(x_0)}{2} \cdot (x - x_0)^2 + \dots + \frac{f^{(n)}(x_0)}{n!} (x - x_0)^n = p_n(x)$$

7.2.1. Chyba aproximace

$E(x) = f(x) - p_n(x)$ – v bodě x_0 je rovno 0

Obecně:

$$E(x) = f(x) - p_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \cdot (x - x_0)^{n+1}$$ kde ξ je neurčité; $\xi \in (x; x_0)$

7.2.2. Odhad chyby aproximace

– jelikož ξ je neznámé číslo, dává vzorec pro výpočet chyby (viz [7.2.1]) pouze kvalitativní informaci

– umíme-li stanovit M takové, že

$$|f^{(n+1)}(x)| \leq M \quad \text{pro } \forall x \in (a,b)$$

potom

$$|E(x)| \leq \frac{M}{(n+1)!} |x - x_0|^{n+1}$$ – úloha se převádí na odhad n+1 ní derivace původní fce

7.3. Aproximace interpolačním polynomem

- pokud máme zadanou funkci pouze jako soubor jejích hodnot (typicky tabulkou), můžeme hledat hodnotu v nějakém bodě $\hat{x}$ ležícím mezi dvěma tabulkovými body

1) stanovíme třídu aproximujících fcí a vybereme jednu konkrétní
2) pomocí zvolené funkce interpolujeme zadané hodnoty

- různé používané druhy interpolačních polynomů:
  - Lagrangeův
  - Newtonův
  - Nevilleův - získáme přímo hodnotu v bodě, nikoli předpis

7.4. L_2 aproximace

- jsou-li hodnoty zatíženy chybami, není interpolace nejvhodnější způsob - volíme raději třídu aproximujících funkcí V_n ($n<m$; $\varphi(x_i) = f(x_i)$, $i=0,1,..,m$) a snažíme se minimalizovat souhrnnou odchylku aproximující funkce $\varphi(x_i)$ od zadaných dat
  → řešíme tedy vlastně metodu nejmenších čtverců:

$$R(f,\varphi) = \sum_{i=0}^{m} \left[f(x_i) - \varphi(x_i) \right]^2 \quad = 0$$

- vypočítáme parciální derivace $R(f,\varphi)$ a položíme je rovno nule → dostaneme soustavu $n+1$ rovnic o $n+1$ neznámých pro c_j, $j=0,1,..,n$ → $\varphi(x) = \sum_{j=0}^{m} c_j \cdot \varphi_j(x)$

tzv. normální rovnice : $\dfrac{\partial R(f,\varphi)}{\partial c_j}$ $j = 0,1,..,n$

Je-li systém základních fcí $\{\varphi_j, j=0,1,..,n\}$ lineárně nezávislý, pak $\exists!$ aproximace $\varphi^*(x) = \sum_{j=0}^{m} c_j^* \cdot \varphi_j(x)$

7.4.1. Spojitá L_2 aproximace

- aproximujeme analogicky k diskrétní L_2 aproximaci (viz [7.4]):

$$R(f,\varphi) = \int_a^b \left[f(x) - \varphi(x)\right]^2 dx \quad = 0$$

- v praxi nemá moc velký význam

7.5. Diskrétní Fourierova transformace

- aproximace obecné funkce trigonometrickými funkcemi

$$f(x) = \sum_{k=0}^{\infty} (a_k \cdot \cos k\omega x + b_k \cdot \sin k\omega x)$$

kde

$$a_k = \frac{1}{\pi} \cdot \int_{-\pi}^{\pi} f(x) \cdot \cos kx \, dx \qquad b_k = \frac{1}{\pi} \cdot \int_{-\pi}^{\pi} f(x) \cdot \sin kx \, dx$$

Fourierova transformace více viz M2/MA2/...

8. Numerické derivování a integrování

- použití v případech, kdy analyticky úloha vyřešit nejde nebo je to moc pracné
- Je-li integrovatelná funkce φ dobrou aproximací funkce f na celém intervalu $\langle a;b \rangle$, je integrál z φ dobrou aproximací integrálu f:

$$\left| \int_a^b f(x)\,dx - \int_a^b \varphi(x)\,dx \right| \leq \int_a^b |f(x) - \varphi(x)|\,dx \leq (b-a) \cdot \sup_{x \in \langle a;b \rangle} |f(x) - \varphi(x)|$$

8.1. Diference

- poměr rozdílu funkčních hodnot a vstupních hodnot

Levá poměrná diference: $$f'(x) = \frac{f(x_0) - f(x_0-h)}{h}$$

Pravá poměrná diference: $$f'(x) = \frac{f(x_0+h) - f(x_0)}{h}$$

Centrální pom. diference: $$f'(x) = \frac{f(x_0+h) - f(x_0-h)}{2h}$$

8.2. Richardsonova extrapolace

- vypočítáme výsledek s různými ~~x~~ kroky a poté výsledky interpolujeme

$$R(h) = \frac{2^n \cdot A(\frac{h}{2}) - A(h)}{2^n - 1}$$ – obecně nemusí být faktor = 2

- Opakováním extrapolace dochází k dalšímu zpřesňování původních výsledků

8.3. Newton-Cotesovy vzorce

- na zadaném intervalu <a;b> nahradíme funkci f jejím interpolačním polynomem s ekvidistantními uzly, vzdálenými $h = (b-a)/N$ kde N je jejich počet

8.3.1. Obdélníkové pravidlo

- funkci nahradíme na celém intervalu funkcí konstantní, s funkční hodnotou, jaké nabývá původní funkce v bodě $x_k + \frac{1}{2}h$, tedy

$$\int_{x_k}^{x_{k+1}} f(x)\,dx \approx h \cdot f\left(x_k + \frac{h}{2}\right) \quad = R_2(f;h)$$

Složený vzorec:

$$\int_a^b f(x)\,dx = \sum_{k=0}^{N-1} \int_{x_k}^{x_{k+1}} f(x)\,dx \approx h \cdot \sum_{k=0}^{N-1} f\left(x_k + \frac{h}{2}\right) \quad = R_2(f;h)$$

8.3.2. Lichoběžníkové pravidlo

- původní funkci nahradíme funkcí lineární – přímkou, spojující bod $[x_k; f(x_k)]$ a $[x_{k+1}; f(x_{k+1})]$

$$\int_{x_k}^{x_{k+1}} f(x)\,dx \approx \frac{h}{2} \cdot \left[f(x_k) + f(x_{k+1})\right] \quad = T_2(f;h)$$

Složený vzorec:

$$\int_a^b f(x)\,dx \approx \frac{h}{2}\left[f(x_0) + 2 \cdot \sum_{k=1}^{N-1} f(x_k) + f(x_N)\right] = h \cdot \left[\frac{1}{2} f(x_0) + \sum_{k=0}^{N-1} f(x_k) + \frac{1}{2} f(x_N)\right] = T_2(f;h)$$

8.3.3. Simpsonovo pravidlo

- nahrazujeme parabolou, která interpoluje v x_k, x_{k+1}, x_{k+2}

$$\int_{x_k}^{x_{k+2}} f(x)\,dx \approx \frac{h}{3}\left[f(x_k) + 4 \cdot f(x_{k+1}) + f(x_{k+2})\right] \quad = S_2(f;h)$$

Složený vzorec: $\int_a^b f(x)\,dx \approx \frac{h}{3}\left[f(x_0) + 4 \cdot f(x_1) + 2 \cdot f(x_2) + 4 \cdot f(x_3) + \ldots + f(x_N)\right] = S_2(f;h)$

N musí být sudé číslo!

8.4. Gaussovy kvadraturní vzorce

- aproximace neekvidistantními uzly
- algebraický řád G. kvadr. vzorce s $n+1$ uzly je $2n+1$ - G. kv. vzorce mají nejvyšší možný algebraický řád
- pro $n \to \infty$ tvoří konvergentní kvadraturu - přesně aproximují hodnotu integrálu integrované funkce, přičemž rychlost konvergence závisí na hladkosti funkce
- vzorce konvergují i pro po částech spojité (s konečným počtem skoků) funkce, byť pomalu
- je vhodné mít uchovány hodnoty koeficientů w_i a uzlů x_i pro jednotlivé stupně vzorce
- uzly x_i jsou rozloženy symetricky okolo středu intervalu