

Numerická matematika

Karel Velička

May 31, 2025

Obsah

1	Principles of Numerical Mathematics	4
1.1	Well-posedness	4
1.2	Continuous Dependence on Data	4
1.2.1	Scalar Case	4
1.2.2	Stability	5
1.3	Errors	5
1.4	Floating Point Arithmetic	5
2	Stability Analysis of Linear Systems.	6
2.1	Metody řešení	6
3	Matrix norms	6
3.1	Matrix Norm	6
3.2	Sensitivity Analysis	7
3.3	Trojúhelníkové soustavy	8
3.3.1	Rounding error analysis	8
3.4	Gaussova eliminační metoda (GEM)	8
3.4.1	Postup	8
3.4.2	LU rozklad	8
4	Spektrální vlastnosti LU rozkladu	10
4.1	Zaokrouhlovací chyby	10
4.2	Problémy stability	10
4.3	Implementace LU rozkladu	10
4.4	Pivotování	10
4.4.1	Částečné pivotování	11
4.4.2	Úplné pivotování	11
4.5	Choleského rozklad	11
4.6	QR rozklad	12
4.7	Singulární rozklad (SVD)	12
5	Iterative methods	13
5.1	Stacionární iterační metody	13
5.1.1	Jacobiho metoda	13
5.1.2	Gaussova-Seidelova metoda	13
5.1.3	SOR metoda (Successive Overrelaxation)	13
5.2	Konvergence iteračních metod	14
5.3	Adaptivní SOR metoda	14

6	Konjugované gradienty (Krylovovy podprostorové metody)	15
6.1	Metoda největšího spádu	15
6.2	Výběr směrových vektorů	15
6.2.1	Optimalita řešení	16
6.3	Konjugovaná ortogonalita	16
6.3.1	Algoritmus konjugovaných gradientů	16
6.4	Celkový algoritmus konjugovaných gradientů	16
7	Výpočet vlastních čísel a vektorů	17
7.1	Power method – Algoritmus	17
7.1.1	Odvození konvergence	17
7.1.2	Výpočet vlastního čísla	18
7.2	Krylovovy podprostorové metody	18
7.2.1	Projection methods	18
7.2.2	Implementace	19
7.2.3	Algoritmus pro SPD matice	19
7.2.4	Vlastnosti	19
7.3	GMRES	19
7.4	Aplikace na problém vlastních čísel	19
8	Řešení nelineárních rovnic	20
8.1	Podmíněnost problému	20
8.2	Bisection Metoda (půlení intervalů)	20
8.3	Rychlejší metody konvergence	21
8.3.1	Chord metoda (tětiva)	21
8.3.2	Secant metoda (sečna)	21
8.3.3	Newtonova metoda	21
8.4	Metoda pevného bodu	21
8.5	Analýza konvergence	22
8.6	Kritéria zastavení iterací	22
8.6.1	Reziduální kritérium	22
8.6.2	Kritérium přírůstku	23
9	Úvod do nelineárních systémů	24
9.1	Newtonova metoda	24
9.2	Kořeny polynomů	25
9.3	Hornerovo schéma	25
9.3.1	Deflační postup	26
9.4	Newton-Hornerova metoda	26
9.4.1	Zpřesňování výsledků	26
10	Aproximace	27
10.1	Polynomiální aproximace	27
10.2	Lagrangeova interpolace	27
10.3	Interpolační chyba	28
10.3.1	Rungeův jev	28
10.3.2	Řešení: Čebyševovy uzly	29
10.4	Po částech polynomiální interpolace	29

11 Splines	30
11.1 Podmínky pro interpolační spliny	30
11.2 Kubické spliny ($k = 3$)	30
11.3 Splines in Parametric Form	31
11.4 De Casteljaouovo schéma	31
12 Orthogonal Polynomials	33
12.1 Three-term rekurence	33
12.2 Zobecněná Fourierova řada	33
12.3 Chebyshev Polynomials	34
13 Numerická Integrace	35
13.1 Quadrature rules.	35
13.1.1 Interpolatory Quadrature	35
13.1.2 Rectangular Rule	35
13.1.3 Midpoint Rule	35
13.1.4 Composite Midpoint Rule	35
13.1.5 Trapezoidal Rule	36
13.1.6 Composite Trapezoidal Rule	36
13.1.7 Simpson's Rule	36
13.1.8 Composite Simpson's Rule	36
13.2 Newton-Cotes Formulas	36
13.3 Gauss Quadrature	36

1 Principles of Numerical Mathematics

1.1 Well-posedness

Matematický problém je obecně formulován jako: najděme proměnnou x , t.ž.

$$F(x, d) = 0,$$

kde d je množina dat a F je funkční vztah mezi x a d .

Well-posed problem (stable) je typ problému, kde existuje *jednoznačné* řešení x , které *spojitě* závisí na datech d . Jinak je problém **Ill-posed (nestabilní)**.

1.2 Continuous Dependence on Data

Pro chyby δx a δd platí vztah

$$F(x + \delta x, d + \delta d) = 0.$$

Problém je stabilní, pokud existují konstanty $y_0 > 0$ a K_0 takové, že:

$$\forall \delta d : \|\delta d\| \leq y_0 \implies \|\delta x\| \leq K_0 \|\delta d\|$$

Definition 1.1 (Relative Condition Number). *Relative condition number problému $F(x, d) = 0$ je:*

$$K(d) := \sup_{\delta d \in D} \frac{\|\delta x\|/\|x\|}{\|\delta d\|/\|d\|}$$

Definition 1.2 (Absolute Condition Number). *Absolute condition number je:*

$$K(d) := \sup_{\delta d \in D} \frac{\|\delta x\|}{\|\delta d\|}$$

Věta 1.1. *Relative condition number pro $x = G(d)$ může být aproximováno formulí $K(d) \approx \frac{\|G'(d)\|}{\|G(d)\|} \|d\|$.*

Proof. Pokud je problém $F(x, d) = 0$ stabilní, pak existuje spojitá funkce G taková, že $x = G(d)$. Pro jednoduchost předpokládejme, že G je diferenciovatelná, pak platí:

$$G(d + \delta d) - G(d) = G'(d) \cdot \delta d + \theta(\|\delta d\|), \quad \delta d \rightarrow 0.$$

Potom tedy dostaneme absolute condition number:

$$K_{\text{abs}}(d) = \sup_{\delta d \in D} \frac{\|\delta x\|}{\|\delta d\|} = \sup_{\delta d \in D} \frac{\|G(d + \delta d) - G(d)\|}{\|\delta d\|} \approx \sup_{\delta d \in D} \frac{\|G'(d)\delta d\|}{\|\delta d\|} \approx \sup_{\delta d \in D} \frac{\|G'(d)\| \|\delta d\|}{\|\delta d\|},$$
$$K_{\text{abs}}(d) \approx \|G'(d)\|.$$

Relative condition number je:

$$K(d) \approx \frac{\|G'(d)\|}{\|G(d)\|} \|d\|.$$

□

1.2.1 Scalar Case

Pro skalární funkci $x = f(d) \in \mathbb{R}$, kde d je vstup, x výstup a f diferencovatelná funkce, je

$$K(d) \approx \frac{|f'(d)|}{|f(d)|} |d|.$$

Pokud je $K(d)$ velké, pak je problém ill-conditioned, naopak pro malé $K(d)$ (≈ 1) je well-conditioned.

1.2.2 Stability

Mějme matematický stabilní problém:

$$F(x, d) = 0, \quad x \in \mathcal{X}, \quad d \in \mathcal{D},$$

a mějme diskretizaci tohoto problému, kterou dostaneme aproximací:

$$F_n(x_n, d_n) = 0, \quad x_n \in \mathcal{X}_n, \quad d_n \in \mathcal{D}_n.$$

Očekáváme, že řešení x_n diskrétního problému $F_n(x_n, d_n) = 0$ konverguje $x_n \rightarrow x$, pro $n \rightarrow \infty$. Pokud pro každé (x, d) platí $F_n(x, d) \rightarrow 0$ pro $n \rightarrow \infty$, pak je aproximace přípustná.

Jelikož je $F_n(x_n, d_n) = 0$ nový problém, můžeme definovat jeho *condition number* jako:

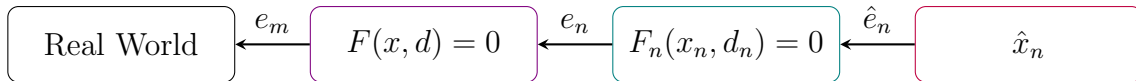
$$K_n(d_n) = \sup_{\delta \in D_n} \frac{\|\delta x_n\| / \|x_n\|}{\|\delta d_n\| / \|d_n\|}$$

Pokud $K_n(d_n)$ je malé, pak je metoda *well-conditioned*, pokud je $K_n(d_n)$ velké, pak je metoda *ill-conditioned*.

Pokud $F_n(x_n, d_n) = 0$ je stabilní, pak existuje spojitá funkce G_n taková, že $x_n = G_n(d_n)$. Pro jednoduchost předpokládejme, že je G_n diferencovatelná, potom:

$$K_n(d_n) \approx \frac{\|G'_n(d_n)\|}{\|G_n(d_n)\|}$$

1.3 Errors



- *Modeling error*: Chyba e_m v matematickém modelu.
- *Discretization error*: Chyba $e_n = x - x_n$ způsobená diskretizací.
- *Computational error*: Chyba $\hat{e}_n = x_n - \hat{x}_n$ způsobená konečnou přesností výpočtů.

Celkovou chybu počítáme jako $e = e_m + e_n + \hat{e}_n$.

1.4 Floating Point Arithmetic

Reálné číslo x je reprezentováno s konečným počtem bytů jako $fl(x)$.

Vlastnost 1.1. Pokud $x \in \mathbb{R}$ a $x_{\min} < |x| < x_{\max}$ (bez podtečení / přetečení), pak

$$fl(x) = x(1 + \delta), \quad |\delta| \leq u,$$

kde u je tzv. *machine precision*.

Důsledek 1.1. Relativní chyba je:

$$E_{rel}(x) = \frac{|x - fl(x)|}{|x|} \leq u$$

2 Stability Analysis of Linear Systems.

Uvažujme soustavu lineárních rovnic $Ax = b$, kde $A \in \mathbb{R}^{m \times m}$ je matice koeficientů, $b \in \mathbb{R}^m$ je vektor pravé strany, a $x \in \mathbb{R}^m$ je hledaný vektor řešení. Pro stabilní problém platí ekvivalentní podmínky:

- A je regulární (existuje A^{-1}),
- $\det(A) \neq 0$,
- $\text{rank}(A) = m$,
- Všechna vlastní čísla matice A jsou $\lambda_j \neq 0$.

2.1 Metody řešení

Pro spočítání x můžeme použít buď přímé, nebo iterační metody:

Přímé metody. Poskytují přesné řešení v konečném počtu kroků (např. Gaussova eliminace). Používá se třeba *Cramerovo pravidlo* (je ale výpočetně náročné): $x_j = \frac{A_j}{\det(A)}$.

Iterační metody. Generují posloupnost aproximací $x_n \rightarrow x$. Jsou vhodné pro rozsáhlé soustavy, kde přímé metody nejsou efektivní.

3 Matrix norms

Definice 3.1 (Condition Number Matrix). *Condition number matice A je:*

$$K(A) = \|A\| \cdot \|A^{-1}\|$$

kde $\|\cdot\|$ je maticová norma.

3.1 Matrix Norm

Indukovanou normu značíme: $\|A\| = \max_v \frac{\|Av\|}{\|v\|}$. Pokud platí $\|Av\| \leq \|A\| \cdot \|v\|$, nazýváme ji **compatible**, pokud platí $\|AB\| \leq \|A\| \cdot \|B\|$, nazýváme ji **submultiplicative**.

Eucleidean norm je $K_2(A) = \|A\|_2 \cdot \|A^{-1}\|_2 = \lambda_n/\lambda$, pokud je A SPD a λ_i vlastní čísla.

Definice 3.2 (Matrix Norm). $\|\cdot\| : \mathbb{R}^{n \times n} \rightarrow \mathbb{R} \geq 0$, pokud:

1. $\|A\| \geq 0 \wedge \|A\| = 0 \iff A = 0$
2. $\|\alpha A\| = |\alpha| \|A\|$
3. $\|A + B\| \leq \|A\| + \|B\|$

Definice 3.3 (Consistency Compatibility). $\|Ax\| \leq \|A\| \cdot \|x\|$, kde $\|A\|$ je matrix norm a $\|x\|$ je vector norm.

- Speciální případy:
 - $\|A\|_1 = \max_j \sum_i |a_{ij}|$, maximální součet sloupce,
 - $\|A\|_\infty = \max_i \sum_j |a_{ij}|$, maximální součet řádku,
 - $\|A\|_2 = \sqrt{\rho(A^T A)}$, spektrální norma – max ‘scale’ by which matrix can ‘stretch’ vector,
 - Frobeniova norma: $\|A\|_F = \sqrt{\sum_{i,j} |a_{ij}|^2}$.

3.2 Sensitivity Analysis

Zkoumáme, zda je problém *well*, nebo *ill-conditioned*.

Věta 3.1. Předpokládejme, že $A + \delta A$ je invertovatelná, $\|A^{-1}\| \|\delta A\| < 1$ a $b \neq 0$, pak pro relativní chybu řešení platí:

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{K(A)}{1 - K(A) \frac{\|\delta A\|}{\|A\|}} \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right)$$

kde $K(A) = \|A\| \|A^{-1}\|$ je condition number matice A .

Proof.

$$(A + \delta A)(x + \delta x) = b + \delta b \iff x + \delta x = (A + \delta A)^{-1}(b + \delta b)$$

Nejprve vyjádříme δx :

$$\begin{aligned} \delta x &= (A + \delta A)^{-1}(b + \delta b) - x = (A + \delta A)^{-1}(b + \delta b - (A + \delta A)x) \\ &= (A + \delta A)^{-1}(b + \delta b - Ax - (\delta A)x) \\ &= [A(I + A^{-1}\delta A)]^{-1}(\delta b - (\delta A)x) = (I + A^{-1}\delta A)^{-1}A^{-1}(\delta b - (\delta A)x) \end{aligned}$$

Pro normu δx tedy:

$$\begin{aligned} \|\delta x\| &\leq \|(I + A^{-1}\delta A)^{-1}A^{-1}(\delta b - (\delta A)x)\| \\ &\leq \|(I + A^{-1}\delta A)^{-1}\| \|A^{-1}\| \|\delta b - (\delta A)x\| \\ &\leq \|(I + A^{-1}\delta A)^{-1}\| \|A^{-1}\| (\|\delta b\| + \|\delta A\| \|x\|) \end{aligned}$$

Předpokládejme, že $\|A^{-1}\delta A\| \leq \|A^{-1}\| \|\delta A\| < 1$, pak víme, že $(I + A^{-1}\delta A)^{-1}$ existuje a

$$\begin{aligned} I &= I - A + A \iff (I - A)^{-1} = I + A(I - A)^{-1} \\ \|(I - A)^{-1}\| &\leq 1 + \|A(I - A)^{-1}\| \\ \|(I - A)^{-1}\| &\leq 1 + \|A\| \|(I - A)^{-1}\| \\ (1 - \|A\|) \|(I - A)^{-1}\| &\leq 1 \iff \|(I - A)^{-1}\| \leq \frac{1}{1 - \|A\|} \end{aligned}$$

Proto, jelikož $\|A^{-1}\delta A\| \leq \|A^{-1}\| \|\delta A\| < 1$:

$$\|(I + A^{-1}\delta A)^{-1}\| \leq \frac{1}{1 - \|A^{-1}\delta A\|} \leq \frac{1}{1 - \|A^{-1}\| \|\delta A\|}$$

Platí tedy:

$$\begin{aligned} \|\delta x\| &\leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\| \|\delta A\|} (\|\delta b\| + \|\delta A\| \|x\|) \\ \frac{\|\delta x\|}{\|x\|} &\leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\| \|\delta A\|} \left(\frac{\|\delta b\|}{\|x\|} + \|\delta A\| \right) \end{aligned}$$

A protože $Ax = b$ a $\|b\| = \|Ax\| \leq \|A\| \|x\| \implies \|x\| \geq \frac{\|b\|}{\|A\|}$, tak dostáváme:

$$\frac{\|\delta x\|}{\|x\|} \leq \frac{\|A^{-1}\|}{1 - \|A^{-1}\| \|\delta A\|} \left(\frac{\|\delta b\|}{\|x\|} + \|\delta A\| \right) \leq \frac{\|A^{-1}\| \|A\|}{1 - \|A^{-1}\| \|A\| \left(\frac{\|\delta A\|}{\|A\|} \right)} \left(\frac{\|\delta b\|}{\|b\|} + \frac{\|\delta A\|}{\|A\|} \right),$$

Kde $K(A) = \|A^{-1}\| \|A\|$.

□

3.3 Trojúhelníkové soustavy

Forward substitution. Pro dolní trojúhelníkovou soustavu $Lx = b$:

$$x_1 = \frac{b_1}{l_{11}},$$
$$x_i = \frac{1}{l_{ii}} \left(b_i - \sum_{j=1}^{i-1} l_{ij} x_j \right), \quad i = 2, \dots, n$$

Backward substitution. Pro horní trojúhelníkovou soustavu $Ux = b$:

$$x_n = \frac{b_n}{u_{nn}},$$
$$x_i = \frac{1}{u_{ii}} \left(b_i - \sum_{j=i+1}^n u_{ij} x_j \right), \quad i = n-1, \dots, 1$$

3.3.1 Rounding error analysis

Uvažujme dolní trojúhelníkovou soustavu $(L + \delta L)\hat{X} = b$, kde:

- L je dolní trojúhelníková matice rozměru $n \times m$
- δL představuje chyby vzniklé zaokrouhlováním
- $\hat{X} = x + \delta x$ je vypočtené řešení s chybou

Za předpokladu $n \cdot u < 1$ (kde u je jednotkové zaokrouhlení), platí následující odhad relativní chyby:

$$\frac{\|x - \hat{x}\|}{\|x\|} \leq \frac{nu \cdot K(L)}{1 - nu \cdot K(L)} = nu \cdot K(L) + \mathcal{O}(u^2)$$

kde $K(L) = \|L\| \|L^{-1}\|$ je condition number matice L .

Pokud u je opravdu malé, pak $n \cdot K(L) \cdot u$ není velké ani pro velké nebo ill-conditioned matice.

3.4 Gaussova eliminační metoda (GEM)

3.4.1 Postup

Transformace $Ax = b$ na horní trojúhelníkový tvar $Ux = c$ pomocí řádkových operací:

- V kroku k eliminujeme sloupec k pod diagonálou pomocí multiplikátorů:

$$m_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}$$

- Aktualizace prvků:

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}$$

3.4.2 LU rozklad

GEM je ekvivalentní rozkladu A na $A = LU$, kde:

- L je dolní trojúhelníková matice s 1 na diagonále,
- U je horní trojúhelníková matice.

Věta 3.2. *LU rozklad existuje a je jednoznačný právě tehdy, když všechny hlavní vedlejší determinanty jsou nenulové.*

Definice 3.4 (Diagonal Dominant Matrix (DDM)). *Matice A je diagonálně dominantní, pokud:*

$$|a_{ii}| \geq \sum_{j=1, j \neq i}^n |a_{ij}| \quad (\text{řádková dominance})$$

nebo

$$|a_{ii}| \geq \sum_{j=1, i \neq j}^n |a_{ji}| \quad (\text{sloupcová dominance})$$

Vlastnost 3.1. *Pokud A je DDM, pak existuje LU rozklad a multiplikátory splňují $|l_{ij}| \leq 1, \forall i, j$.*

4 Spektrální vlastnosti LU rozkladu

Determinant. Pro LU rozklad platí:

$$\det(A) = \det(L) \cdot \det(U) = \det(U)$$

protože $\det(L) = 1$ (jedničky na diagonále).

4.1 Zaokrouhlovací chyby

Při výpočtu na počítači dostáváme:

$$A \neq \hat{A} = \hat{L}\hat{U}$$

kde $\hat{L}$ a $\hat{U}$ jsou ovlivněny zaokrouhlovacími chybami. Chyba rozkladu je:

$$\hat{A} - A = \delta A$$

Odhad chyby. Pro relativní chybu platí:

$$|\delta A| \leq \frac{nu}{1 - nu} |A|$$

kde u je strojové epsilon.

4.2 Problémy stability

Pokud jsou pivotní prvky malé, mohou multiplikátory být extrémně velké, což vede k nestabilitě.

Například pro matici $A = \begin{pmatrix} 1 & 2 & 1 \\ 1 & 2 + \epsilon & -1 \\ 0 & 1 & 1 \end{pmatrix}$ je multiplikátor $m_{32} = 1/\epsilon$, který je velký pro malé ϵ .

4.3 Implementace LU rozkladu

Následující MATLAB kód implementuje LU rozklad (kji verze):

```
function [A] = lu_kji(A)
[n,n] = size(A);
for k = 1:n-1
    A(k+1:n,k) = A(k+1:n,k)/A(k,k);
    for j = k+1:n
        for i = k+1:n
            A(i,j) = A(i,j) - A(i,k)* A(k,j);
        end
    end
end
end
```

4.4 Pivotování

GEM selže, pokud je pivot nulový nebo blízký nule. Řešením je výměna řádků.

Například pro matici:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 4 & 5 \\ 7 & 8 & 9 \end{pmatrix}$$

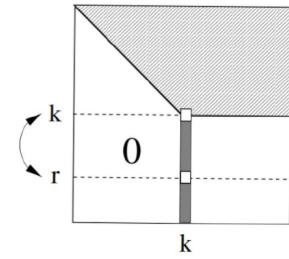
výměna (permutace) druhého a třetího řádku umožní úspěšnou eliminaci. Permutace řádků se reprezentuje permutační maticí P :

$$PA = LU$$

4.4.1 Částečné pivotování

V podstatě $A \rightarrow A^{(2)} \rightarrow \dots \rightarrow A^{(k)}$. Viděli jsme, že malé multiplikátory způsobují nestabilitu. Idea je následující:

- $A^{(k)}(k:\text{end}, k) \leftarrow$ hledáme prvek s největší magnitudou
- Prohodíme daný řádek s k -tým řádkem



Výslednou horní trojúhelníkovou matici U vyjádřit obecným vzorcem:

$$U = A^{(n)} = M_{n-1} \cdot P_{n-1} \cdots M_1 \cdot P_1 \cdot A^{(1)},$$

kde M_k jsou eliminační matice, P_k jsou matice permutací řádků a $A^{(1)} = A$ je původní matice. Definujeme celkovou permutaci a transformaci:

$$P = P_{m-1} \cdots P_1, \quad M = M_{m-1} \cdot P_{m-1} \cdots M_1 \cdot P_1$$

Pak můžeme psát:

$$U = MA \iff U = (MP^{-1})PA$$

kde MP^{-1} je dolní trojúhelníková matice. Odtud dostáváme LU rozklad pro permutovanou matici:

$$PA = LU$$

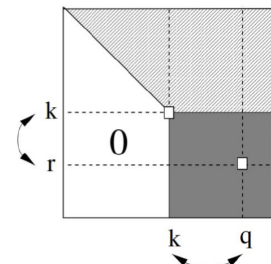
kde $L = (MP^{-1})^{-1}$ je dolní trojúhelníková matice s jedničkami na diagonále.

4.4.2 Úplné pivotování

Zahrnuje výměnu řádků i sloupců:

$$PAQ = LU,$$

kde Q je permutační matice pro sloupce.



Částečné pivotování zlepšuje stabilitu oproti GEM bez pivotování, zatímco úplné pivotování je stabilnější, ale výpočetně náročnější.

4.5 Choleského rozklad

Pokud A je SPD matice, pak existuje rozklad $A = C^T C$, kde C je horní trojúhelníková matice s kladnými diagonálními prvky.

Platí následující vlastnosti:

- $A = LU, \quad D = \text{diag}(U) \implies C = (D^{-1/2})U = (D^{1/2})L^T$ a tedy

$$LU = LD^{1/2}D^{-1/2}U = C^T C.$$

- Rozklad C existuje a je jednoznačný
- Dekompozice je stabilní.

4.6 QR rozklad

Pro nesingulární matici $A \in \mathbb{R}^{n \times n}$ existuje rozklad:

$$A = QR$$

kde Q je ortogonální ($Q^T Q = I$) a R je nesingulární horní trojúhelníková matice.
(QR rozklad může být použit i u obdélníkových matic.)

Aplikace. Při řešení soustavy $Ax = b$:

$$QRx = b \implies Rx = Q^T b$$

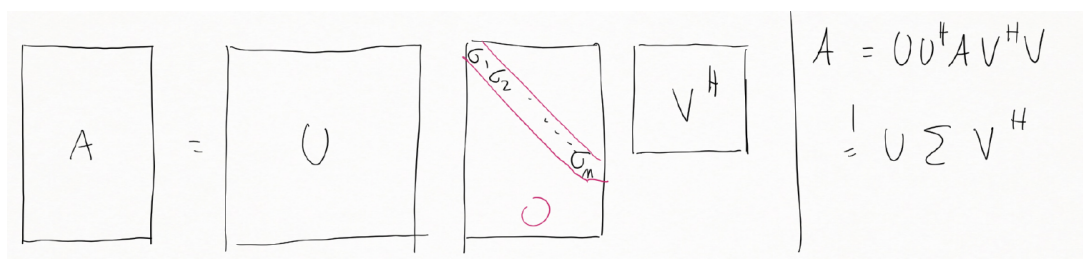
což lze řešit zpětnou substitucí.

4.7 Singulární rozklad (SVD)

Pro libovolnou matici $A \in \mathbb{C}^{m \times n}$ existuje rozklad:

$$A = U \Sigma V^H$$

kde $U \in \mathbb{C}^{m \times m}$, $V \in \mathbb{C}^{n \times n}$ jsou unitární a Σ obsahuje singulární hodnoty $\sigma_1 \geq \dots \geq \sigma_p \geq 0$.



Platí následující vztahy:

$$A^H A = V^H \Sigma^H U U^H \Sigma V = V^H \Sigma^H \Sigma V, \quad \text{a}$$

$$A A^H = U \Sigma V^H V \Sigma^H U^H = U \Sigma \Sigma^H U^H$$

Nenulové prvky $\Sigma^H \Sigma$, $\Sigma \Sigma^H$ jsou $|\sigma_1|^2, |\sigma_2|^2, \dots, |\sigma_p|^2 \implies \sigma_i(A) = \sqrt{\lambda_i(A^H A)} = \sqrt{\lambda_i(A A^H)}$, pro $i = 1, \dots, p$.

Platí $\text{rank}(A) = \text{počet nenulových singulárních hodnot}$

5 Iterative methods

Iterační metody jsou vhodné pro velké matice (přímé metody jsou příliš náročné) a pro řídké matice (násobení matice a vektoru je rychlé).

Základní princip je následující:

- Začneme s počáteční aproximací x_0
- V každé iteraci vypočítáme novou aproximaci x_{k+1}
- Proces opakujeme, dokud nedosáhneme požadované přesnosti

5.1 Stacionární iterační metody

Rozklad matice A na $A = K - L$, kde K je snadno invertovatelná matice.

$$(K - L)x = b \iff Kx - Lx = b \iff Kx = Lx + b \iff x = K^{-1}Lx + K^{-1}b.$$

Iterační proces/rekurze:

$$x_{n+1} = K^{-1}Lx_n + K^{-1}b,$$

pro počáteční odhad x_0 .

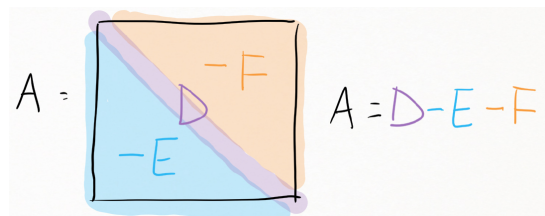
Pokud $\{x_n\}_{n=0}^{\infty}$ konverguje, pak pro $n \rightarrow \infty$ je $x = K^{-1}Lx + K^{-1}b \implies x$ řešení.

5.1.1 Jacobiho metoda

Cílem je zničit i-tou složku zbytku.

- Rozklad: $K_J = D$ (diagonální část A), $L_J = E + F = K_J - A$ (ostře trojúhelníkové části).
- Iterační vzorec: $x^{(k+1)} = K_J^{-1}(L_J x^{(k)} + b)$.
- Komponentně:

$$\xi_i^{(m+1)} = \frac{1}{a_{ii}} \left(- \sum_{j \neq i} a_{ij} \xi_j^{(m)} + b_i \right).$$



5.1.2 Gaussova-Seidelova metoda

Cíl je podobný jako v Jacobiho metodě, akorát okamžitě aktualizujeme aproximované řešení.

- Rozklad: $K_G = D - E = L + D$, $L_G = F = -U$.
- Okamžité využití vypočtených složek.
- Iterační vzorec: $x^{(k+1)} = K_G^{-1}(L_G x^{(k)} + b)$.

5.1.3 SOR metoda (Successive Overrelaxation)

- Zavádí relaxační parametr $\omega \neq 0 \rightsquigarrow \omega Ax = \omega b$.
- Rozklad: $\omega A = \underbrace{(D - \omega E)}_K - \underbrace{(\omega F + (1 - \omega)D)}_L$.
- Iterační vzorec: $x_{m+1} = (D - \omega E)^{-1}[\omega F + (1 - \omega)D]x_m + \omega(D - \omega E)^{-1}b$.
- Pro $\omega = 1$ přechází na Gaussovu-Seidelovu metodu.

5.2 Konvergence iteračních metod

Všechny předchozí metody jsou ve formě $x_{n+1} = Gx_n f$, kde G je iterační matice a $n \rightarrow \infty$. Pokud $x_n \rightarrow x$, pak pro $n \rightarrow \infty$ platí $x = Gx + f$.

Metoda x_{n+1} konverguje $\iff \rho(G) < 1$, kde $\rho(G)$ je spektrální poloměr iterační matice $G = K^{-1}L$. Postačující podmínka je, aby platilo $\|G\| < 1$ pro libovolnou konzistentní maticovou normu.

Asymptotická rychlost konvergence.

$$\rho = \lim_{n \rightarrow \infty} \left(\frac{\|x_n - x\|}{\|x_0 - x\|} \right)^{1/n} = \rho(G)$$

Definice 5.1 ((Weakly) Diagonally Dominant (DD)). *Matice je diagonálně dominantní, pokud*

$$|a_{ii}| \geq \sum_{j=1, j \neq i}^n |a_{ij}|, \quad i = 1, \dots, n$$

podle řádků.

Definice 5.2 (Strictly Diagonally Dominant (SDD)). *Matice je striktně diagonálně dominantní, pokud*

$$|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}|, \quad i = 1, \dots, n$$

podle řádků.

Věta 5.1. *Pro ostře diagonálně dominantní matice konvergují Jacobiho i Gaussova-Seidelova metoda.*

Proof. (Pro Jacobiho).

$$G_J = I - D^{-1}A = D^{-1}(E + F) = D^{-1}(D - A) = I - D^{-1}A.$$

$$\|G_J\|_{\infty} = \max_{i=1, \dots, n} \left(\sum_{j=1}^N |(G_J)_{ij}| \right) = \max_{i=1, \dots, n} \sum_{j=1, j \neq i}^N \frac{|a_{ij}|}{|a_{ii}|} + \left| 1 - \frac{a_{ii}}{a_{ii}} \right| < 1$$

□

Věta 5.2 (Ostrowski). *Pro SPD matice konverguje SOR metoda právě když $0 < \omega < 2$*

5.3 Adaptivní SOR metoda

Metodu SOR můžeme zlepšit tak, že budeme měnit ω v každé iteraci pro urychlení konvergence a využijeme tak informace z předchozích iterací. Obecný iterační vzorec tak je:

$$x_{n+1} = G_n x_n + f_n$$

6 Konjugované gradienty (Krylovovy podprostorové metody)

Pro SPD matici A uvažujeme kvadratickou formu:

$$\Phi(y) = \frac{1}{2}y^T A y - y^T b$$

Minimum nastává v řešení $Ax = b$, protože:

$$\nabla \Phi(y) = Ay - b$$

Vlastnosti řešení

Naopak, pokud x řeší $Ax = b$, pak:

$$\begin{aligned}\Phi(y) &= \frac{1}{2}(x + (y - x))^T A(x + (y - x)) - (x + (y - x))^T b \\ &= \Phi(x) + \frac{1}{2}(y - x)^T A(y - x)\end{aligned}$$

Poznámka:

$$\Phi(y) - \Phi(x) = \frac{1}{2}\|y - x\|_A^2 = \frac{1}{2}(y - x)^T A(y - x)$$

6.1 Metoda největšího spádu

Idea:

- Směry: $X_{n+1} = X_n + \alpha_n P_n$
- Reziduum: $\nabla F(x_n) = Ax_n - b = -r_n$

Optimalizace kroku

Pro $x_{n+1} = x_n + \alpha_n r_n$, jaké je nejlepší α_n ?

$$\begin{aligned}\Phi(x_{n+1}) &= \frac{1}{2}(x_n + \alpha p_n)^T A(x_n + \alpha p_n) - (x_n + \alpha p_n)^T b \\ \frac{d}{d\alpha} \Phi(x_{n+1}) &= p_n^T A x_n + \alpha p_n^T A p_n - p_n^T b \\ &= p_n^T (A x_n - b) + \alpha p_n^T A p_n \\ \alpha_n &= \frac{p_n^T r_n}{p_n^T A p_n}\end{aligned}$$

6.2 Výběr směrových vektorů

Jak volit p_n ?

- Pokud $p_n = r_n = -\nabla \Phi(x_n) \rightarrow$ gradientní metoda
- Chceme vylepšit. Chceme minimalizovat rozdíl:

$$\Phi(x_n) = \frac{1}{2}\|x_n - x\|_A^2$$

tj. minimalizovat A -normu chyby

Poznámka:

$$\begin{aligned}x_n &= x_0 + \alpha_0 p_0 + \alpha_1 p_1 + \dots + \alpha_{n-1} p_{n-1} \\ \Rightarrow x_n &\in x_0 + \text{span}\{p_0, \dots, p_{n-1}\}\end{aligned}$$

6.2.1 Optimalita řešení

$$\|x - x_n\|_A = \min_{z \in x_0 + \text{span}\{p_0, \dots, p_{n-1}\}} \|x - z\|_A$$

Pythagorova věta:

$$\|x - z\|_A^2 = \|x - x_n\|_A^2 + \|z - x_n\|_A^2 \geq \|x - x_n\|_A^2$$

Podmínka:

$$p_j \perp_A (x - x_n) \Leftrightarrow p_j^T A(x - x_n) = 0 \Leftrightarrow p_j^T r_n = 0$$

pro $j = 0, \dots, m-1$

6.3 Konjugovaná ortogonalita

$$\begin{aligned} 0 &= p_j^T r_n = p_j^T (Ax_n - b) = p_j^T (A(x_{n-1} + \alpha_{n-1}p_{n-1}) - b) \\ &= p_j^T r_{n-1} - \alpha_{n-1} p_j^T A p_{n-1} \end{aligned}$$

Pro $j = m-1$:

$$\alpha_{n-1} = \frac{p_{n-1}^T r_{n-1}}{p_{n-1}^T A p_{n-1}}$$

Pro $j = 0, \dots, m-2$:

$$\begin{aligned} p_j^T r_{n-1} &= 0 \quad \text{a} \quad p_j^T A p_{n-1} = 0 \\ \Rightarrow \{p_0, \dots, p_{n-1}\} &\text{ tvoří } A\text{-ortogonální bázi} \end{aligned}$$

6.3.1 Algoritmus konjugovaných gradientů

Předpokládejme $p_0 = r_0 = b - Ax_0$

Obecný směr:

$$p_n = r_n - \beta_{n-1}p_{n-1} - \dots - \beta_0p_0$$

Podmínka A -ortogonalita:

$$\beta_j = \frac{p_j^T A r_n}{p_j^T A p_j}$$

6.4 Celkový algoritmus konjugovaných gradientů

1. Inicializace: $p_0 = r_0 = b - Ax_0$

2. Výpočet kroku:

$$\alpha_n = \frac{\|r_n\|^2}{p_n^T A p_n}$$

3. Aktualizace řešení: $x_{n+1} = x_n + \alpha_n p_n$

4. Aktualizace rezidua: $r_{n+1} = r_n - \alpha_n A p_n$

5. Výpočet koeficientu:

$$\beta_n = \frac{\|r_{n+1}\|^2}{\|r_n\|^2}$$

6. Aktualizace směru: $p_{n+1} = r_{n+1} - \beta_n p_n$

7 Výpočet vlastních čísel a vektorů

Výpočet vlastních čísel (eigenvalues) a vlastních vektorů (eigenvectors) je klíčový pro mnoho aplikací:

- Inženýrství: analýza vibrací a rezonancí
- Počítačové vědy: PageRank (Google) je v jádru problém vlastních vektorů

Matematický problém: $Av = \lambda v$, $v \neq 0$, kde

$$A = V\Lambda V^{-1} \quad \text{nebo} \quad VJV^{-1},$$

pro Λ diagonální matici a J Jordanovu matici. Platí také $AV = \Lambda V$.

7.1 Power method – Algoritmus

Předpokládejme, že A je diagonalizovatelná s $|\lambda_1| > |\lambda_2| \geq \dots \geq |\lambda_n|$. Vlastní číslo λ_1 má algebraic multiplicity 1 a nazýváme ho *dominantním*.

Algoritmus:

1. Zvolíme $q^{(0)} \in \mathbb{C}^n$ s $\|q^{(0)}\|_2 = 1$

$$2. \text{ Iterace: } \begin{cases} z^{(k)} = Aq^{(k-1)} \\ q^{(k)} = z^{(k)} / \|z^{(k)}\|_2 \\ \gamma^{(k)} = (q^{(k)})^H Aq^{(k)} \end{cases}$$

Pro $\gamma^{(k)} \rightarrow \lambda$ aproximuje dominantní vlastní číslo a odpovídající vlastní vektor.

7.1.1 Odvození konvergence

Matematickou indukci dokážeme:

$$q^{(k)} = \frac{A^k q^{(0)}}{\|A^k q^{(0)}\|}, \quad \forall k \geq 1$$

Předpokládáme, že $A = X\Lambda X^{-1}$ je diagonalizovatelná, kde X je báze $\mathbb{C}^n$:

$$q^{(0)} = \sum_{i=1}^n \alpha_i x_i, \quad \alpha_i \in \mathbb{C}$$

a platí $Ax_i = \lambda_i x_i$.

$$\begin{aligned} Aq^{(0)} &= \alpha_1 \lambda_1 x_1 + \alpha_2 \lambda_2 x_2 + \dots + \alpha_m \lambda_m x_m \\ A^2 q^{(0)} &= \alpha_1 \lambda_1^2 x_1 + \alpha_2 \lambda_2^2 x_2 + \dots + \alpha_m \lambda_m^2 x_m \end{aligned}$$

Obecně:

$$A^k q^{(0)} = \alpha_1 \lambda_1^k \left(x_1 + \underbrace{\sum_{i=2}^n \frac{\alpha_i}{\alpha_1} \left(\frac{\lambda_i}{\lambda_1} \right)^k x_i}_{y^{(k)}} \right)$$

Za předpokladu $\frac{|\lambda_i|}{|\lambda_1|} < 1$ pro $i = 2, \dots, n$ dostáváme:

$$\begin{aligned} \left(\frac{\lambda_i}{\lambda_1} \right)^k &\rightarrow 0 \text{ pro } k \rightarrow +\infty \implies y^{(k)} \rightarrow 0 \\ \implies q^{(k)} &= \frac{\alpha_1 \lambda_1^k (x_1 + y^{(k)})}{\|\alpha_1 \lambda_1^k (x_1 + y^{(k)})\|_2} \rightarrow \beta_k x_1 \quad \text{směr vlastního vektoru } x_1 \end{aligned}$$

Např. pro $\lambda_1 > 0$ a $\alpha_1 \in \mathbb{R}$: $q^{(k)} \rightarrow x_1$.

Věta 7.1. *Nechť A splňuje předchozí předpoklady a $\alpha_1 \neq 0$. Pak existuje $C > 0$ takové, že:*

$$\|\tilde{q}^{(k)} - x_1\|_2 \leq C \left| \frac{\lambda_2}{\lambda_1} \right|^k, \quad k \geq 1$$

kde:

$$\tilde{q}^{(k)} = \frac{q^{(k)} \|A^k q^{(0)}\|_2}{\alpha_1 \lambda_1^k} = x_1 + \sum_{i=2}^n \frac{\alpha_i}{\alpha_1} \left(\frac{\lambda_i}{\lambda_1} \right)^k x_i$$

Proof. Předpokládejme x_i , že $\|x_i\|_2 = 1$. Pak:

$$\begin{aligned} & \left\| x_1 + \sum_{i=2}^n \left(\frac{\alpha_i}{\alpha_1} \left(\frac{\lambda_i}{\lambda_1} \right)^k x_i \right) - x_1 \right\|_2 = \left\| \sum_{i=2}^n \left(\frac{\alpha_i}{\alpha_1} \left(\frac{\lambda_i}{\lambda_1} \right)^k x_i \right) \right\|_2 = \\ & = \left| \sum_{i=2}^n \left(\frac{\alpha_i}{\alpha_1} \left(\frac{\lambda_i}{\lambda_1} \right)^k \right)^2 \right|^{1/2} \leq \underbrace{\left| \frac{\lambda_2}{\lambda_1} \right|^k}_{\text{největší hodnota}} \underbrace{\left(\sum_{i=2}^n \left| \frac{\alpha_i}{\alpha_1} \right|^2 \right)^{1/2}}_C. \end{aligned}$$

□

Poznámka 7.1. Čím menší $\left| \frac{\lambda_2}{\lambda_1} \right|$, tím rychlejší konvergence.

7.1.2 Výpočet vlastního čísla

Pro normalizovaný vlastní vektor $\|x_1\|_2 = 1$:

$$x_1^H A x_1 = x_1^H (\lambda_1 x_1) = \lambda_1 x_1^H x_1 = \lambda_1 \implies (q^{(k)})^H A q^{(k)} = \gamma^{(k)} \rightarrow \lambda_1.$$

Pro reálné symetrické matice platí:

$$|\lambda_1 - \gamma^{(k)}| \leq K |\lambda_1 - \lambda_n| \left| \frac{\lambda_2}{\lambda_1} \right|^{2k}.$$

7.2 Krylovovy podprostorové metody

Cílem je projekce velkého problému na menší podprostor s podobnými vlastnostmi. Pomáhá nám s:

- řešením soustav $Ax = b$,
- výpočtem vlastních čísel,
- výpočtem maticových funkcí.

7.2.1 Projection methods

Pro $A \in \mathbb{R}^{N \times N}$, $x_0, b \in \mathbb{R}^N$:

- Vyhledávací podprostor: $S_m \subseteq \mathbb{R}^N$; hledáme aproximaci $x_m \in x_0 + S_m$,
- Constraints space: $\mathcal{C}_m \subseteq \mathbb{R}^N$, s $r_m = b - Ax_m \perp \mathcal{C}_m$.

Definice 7.1 (Krylovův podprostor).

$$\mathcal{K}_m(A, v) = \text{span}\{v, Av, A^2v, \dots, A^{m-1}v\}.$$

(Pokud $u \in \mathcal{K}_m(A, v)$, pak $u = \alpha_0 v + \alpha_1 Av + \dots + \alpha_{m-1} A^{m-1}v = p_{m-1}(A)v$.)

Uvažme $S_m = \mathcal{K}_m(A, r_0)$, kde $r_0 = b - Ax_0$ pro x_0 initial guess. Všimněme si, že $A(x - x_0) = Ax - Ax_0 = b - Ax_0 = r_0$. Řešení $Ay = r_0$, $x = y + x_0$ je tedy ekvivalentní k řešení $Ax = b$.

$$x_m \in x_0 + \mathcal{K}_m(A, r_0) \implies x_m - x_0 \in \mathcal{K}_m(A, r_0) \implies x_m - x_0 = p_{m-1}(A)r_0$$

kde $x_m = x_0 + V_m t_m$ s V_m bází $\mathcal{K}_m(A, r_0)$ a vektorem t_m .

Dále platí $x - x_0 = A^{-1}(r_0)$, tedy $A(x - x_0) = r_0$.

Idea: aproximace polynomu $x - x_0 = A^{-1}r_0 \approx p_{m-1}(A)r_0 = x_m - x_0$.

Constraint Space. $\mathcal{C} = \mathcal{K}_m(A, r_0)$. Nechť $r_m \perp \mathcal{C}$ a předpokládejme že V_m je ortogonální bází pro $\mathcal{K}_m(A, r_0)$.

$$\begin{aligned} V_m^T r_m = 0 &= V_m^T (b - Ax_m) = V_m^T (b - A(x_0 + V_m t_m)) = \\ &= V_m^T (b - Ax_0 - AV_m t_m) = V_m^T r_0 - V_m^T AV_m t_m \implies \\ &\implies \begin{cases} V_m^T AV_m t_m = V_m^T r_0, \\ x_m = x_0 + V_m t_m. \end{cases} \end{aligned}$$

7.2.2 Implementace

1. Sestrojíme ortonormální bází V_m prostoru $\mathcal{K}_m(A, r_0)$
2. Řešíme projekci:

$$\begin{aligned} H_m &= V_m^T AV_m \\ H_m t_m &= V_m^T r_0 \\ x_m &= x_0 + V_m t_m \end{aligned}$$

7.2.3 Algoritmus pro SPD matice

1. Inicializace: $p_0 = r_0 = b - Ax_0$
2. Pro $m = 0, 1, \dots$:

$$\begin{aligned} \alpha_m &= \frac{\|r_m\|^2}{p_m^T A p_m} \\ x_{m+1} &= x_m + \alpha_m p_m \\ r_{m+1} &= r_m - \alpha_m A p_m \\ \beta_m &= \frac{\|r_{m+1}\|^2}{\|r_m\|^2} \\ p_{m+1} &= r_{m+1} + \beta_m p_m \end{aligned}$$

7.2.4 Vlastnosti

$x_{m+1} \in x_0 + \mathcal{K}_m(A, r_0)$, dále $r_{m+1} \perp \mathcal{K}_m(A, r_0)$ a minimalizuje $\|x - x_m\|_A$.

7.3 GMRES

Pro obecné matice: $S_m = \mathcal{K}_m(A, r_0)$, $\mathcal{C}_m(A, r_0) = A\mathcal{K}_m(A, r_0)$. Je dobře definováno pro A nesingulární. Minimalizuje $\|r_n\|$.

7.4 Aplikace na problém vlastních čísel

Idea: $H_m = V_m^T AV_m \implies \sigma(H_m) \approx \sigma(A)$

8 Řešení nelineárních rovnic

Hledáme kořen $\alpha \in \mathbb{C}$ funkce $f : I = (a, b) \subseteq \mathbb{R} \rightarrow \mathbb{R}$ takový, že: $f(\alpha) = 0$.

Iterační metody. Metody generující posloupnost $\{x^{(k)}\}$ splňující:

$$\lim_{k \rightarrow +\infty} x^{(k)} = \alpha$$

Definice 8.1. Posloupnost $\{x^{(k)}\}$ konverguje k α s řádem $p \geq 1$ právě když existuje $C > 0$ a $k_0 \in \mathbb{N}$ takové, že:

$$\frac{|x^{(k+1)} - \alpha|}{|x^{(k)} - \alpha|^p} \leq C \quad \forall k \geq k_0$$

Metodě generující $\{x^{(k)}\}$ říkáme **řád** p . A pokud $C < \infty$, $x^{(k)} \rightarrow \alpha$, pak C nazýváme **faktor konvergence**.

8.1 Podmíněnost problému

Problém $F(x, d) = 0$ s řešením $x = G(d)$ má podmíněnost:

$$K(d) \approx \|G'(d)\| \cdot \frac{\|d\|}{\|G(d)\|}, \quad K_{\text{abs}}(d) \approx \|G'(d)\|$$

Hledání kořenů. Pro $f(x) = \varphi(x) - d = 0$ a $\alpha = \varphi^{-1}(d)$, problém je well-posed:

$$K(d) \approx \frac{|d|}{|\alpha| |f'(\alpha)|}, \quad K_{\text{abs}}(d) \approx \frac{1}{|f'(\alpha)|}$$

8.2 Bisection Metoda (půlení intervalů)

Princip. Počáteční interval $I_0 = [a, b]$ s $f(a)f(b) < 0$, f spojitá $\implies$ existuje $\alpha \in (a, b)$ s $f(\alpha) = 0$.

Algoritmus.

1. $x^{(k)} = \frac{a^{(k)} + b^{(k)}}{2}$
2. Pokud $f(x^{(k)})f(a^{(k)}) < 0$, pak $a^{(k+1)} = a^{(k)}$, $b^{(k+1)} = x^{(k)}$
3. Jinak $a^{(k+1)} = x^{(k)}$, $b^{(k+1)} = b^{(k)}$

Konvergence.

$$|x^{(k)} - \alpha| < \frac{|b - a|}{2^{k+1}}$$

Konverguje globálně, ale pomalu. Pro přesnost $|x^{(k)} - \alpha| < \varepsilon$ potřebujeme $|I_k| < \varepsilon$:

$$\begin{aligned} \frac{b - a}{2^{k+1}} < \varepsilon &\iff 2^{k+1} > \frac{b - a}{\varepsilon} \\ &\iff \log_2(2^{k+1}) > \log_2\left(\frac{b - a}{\varepsilon}\right) \\ &\iff k > \log_2\left(\frac{b - a}{\varepsilon}\right) - 1. \\ k &> \log_2\left(\frac{b - a}{\varepsilon}\right) - 1 \end{aligned}$$

8.3 Rychlejší metody konvergence

Idea. Taylorův rozvoj kolem kořene α :

$$f(\alpha) = 0 \approx f(x) + (\alpha - x)f'(x),$$

jelikož α známe, nahradíme ho s x^{k+1} a získáme iterační metodu danou řešením rovnice

$$f(x^{(k)}) + (x^{(k+1)} - x^{(k)})f'(x^{(k)}) = 0.$$

Aproximujeme $q_k \approx f'(x^{(k)})$ a získáme

$$x^{(k+1)} = x^{(k)} - q_k^{-1}f(x^{(k)}).$$

8.3.1 Chord metoda (tětiva)

Dáno následujícím vztahem. Řád konvergence je $p = 1$.

$$q_k = q = \frac{f(b) - f(a)}{b - a}$$

8.3.2 Secant metoda (sečna)

Dáno následujícím vztahem. Řád konvergence je $p = \frac{1+\sqrt{5}}{2} \approx 1.63$.

$$q_k = \frac{f(x^{(k)}) - f(x^{(k-1)})}{x^{(k)} - x^{(k-1)}}$$

8.3.3 Newtonova metoda

Předpokládejme, že jsme schopni spočítat $f'(x)$ a $f'(\alpha) \neq 0$. Pak $f'(x) \neq 0$ v sousedství α . Pro dané x_0 máme:

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}.$$

Řád konvergence: $p = 2$.

8.4 Metoda pevného bodu

Vždy můžeme transformovat problém z $f(x) = 0$ na ekvivalentní problém $x - \phi(x) = 0$, že $\phi(\alpha) = \alpha$, kdykoli $f(\alpha) = 0$. Pak α je **fixed point** zobrazení ϕ . Iterace:

$$x^{(k+1)} = \phi(x^{(k)}).$$

Věta 8.1. Předpokládejme, že:

1. $\phi : [a, b] \rightarrow [a, b]$
2. $\phi \in C^1([a, b])$
3. $\exists 0 < k < 1 : |\phi'(x)| < k \ \forall x \in [a, b]$

Pak ϕ má právě jeden pevný bod $\alpha \in [a, b]$ a posloupnost iterací $x^{(k)} = \phi(x^{(k-1)})$ konverguje k α .
Dokonce

$$\lim_{k \rightarrow \infty} \frac{x^{k+1} - \alpha}{x^k - \alpha} = \phi'(\alpha).$$

Tato věta nám zajistí globální konvergenci. Lokální konvergenci zajišťuje *Ostrowskiho věta*.

Věta 8.2 (Ostrowski). *Nechť ϕ je spojitě diferencovatelná v okolí J pevného bodu α . Pokud $|\phi'(\alpha)| < 1$, pak existuje $\delta > 0$ takové, že posloupnost $\{x^{(k)}\}$ konverguje k α pro všechna $x^{(0)}$ splňující $|x^{(0)} - \alpha| < \delta$.*

Vlastnost 8.1. *Nechť $\phi \in C^{p+1}(J)$, kde J je okolí pevného bodu α , $p \geq 0$. Pokud $\phi^{(i)}(\alpha) = 0$ pro $i = 1, \dots, p$, pak odpovídající metoda pevného bodu má řád konvergence $p + 1$ a platí:*

$$\lim_{k \rightarrow \infty} \frac{x^{(k+1)} - \alpha}{(x^{(k)} - \alpha)^{p+1}} = \frac{\phi^{(p+1)}(\alpha)}{(p+1)!}$$

8.5 Analýza konvergence

Metoda těživ jako metoda pevného bodu.

$$\phi(x) = x - q^{-1}f(x) = x - \frac{b-a}{f(b)-f(a)}f(x)$$

Podmínky konvergence:

- $\phi \in C^1([a, b]) \iff f(x) \in C^1 \in [a, b]$.
- $|\phi'(\alpha)| = |1 - \frac{f'(a)}{q}| \begin{cases} = 1 & \iff f'(\alpha) = 0 \quad (\text{kovergence není garantována}) \\ < 1 & \iff 0 < q^{-1}f'(a) < 2 \quad (\text{konverguje}) \end{cases}$

Dále pokud $\frac{f(b)-f(a)}{b-a}$ a $f'(\alpha)$ mají stejné znaménko a $\left| \frac{f(b)-f(a)}{f'(\alpha)} \right| < 2$.

Newtonova metoda.

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})}, \quad \phi(x) = x - \frac{f(x)}{f'(x)}$$

Předpokládejme, že $f'(\alpha) \neq 0$, pak $\phi'(\alpha) = 0$ a $\phi''(\alpha) = \frac{f''(\alpha)}{f'(\alpha)}$.

Modifikovaná Newtonova metoda Pokud α má násobnost m :

$$x^{(k+1)} = x^{(k)} - m \frac{f(x^{(k)})}{f'(x^{(k)})}$$

Tato modifikace zachovává řád konvergence 2.

8.6 Kritéria zastavení iterací

Chceme vědět, kdy zastavíme iterování $\{x^{(k)}\} \rightarrow \alpha$. Máme $e^{(k)} = \alpha - x^{(k)}$ a chceme $|e^{(k)}| < \text{daná hodnota}$.

8.6.1 Reziduální kritérium

$f(x^{(k)}) \rightarrow 0 \implies$ Kontrolujeme velikost rezidua: $|f(x^{(k)})| < \varepsilon$.

Problém: je citlivé na hodnotu $|f'(\alpha)|$:

$$|e^{(k)}| \approx \frac{|f(x^{(k)})|}{|f'(\alpha)|}$$

Chování v různých případech:

- $|f'(\alpha)| \approx 1$: test funguje dobře
- $|f'(\alpha)| \approx 0$: chyba může být mnohem větší než residuum
- $|f'(\alpha)| \gg 1$: test může být příliš restriktivní

8.6.2 Kritérium přírůstku

$$|x^{(k+1)} - x^{(k)}| < \varepsilon$$

Odhad chyby:

$$e^{(k)} \approx \frac{1}{1 - \phi'(\alpha)}(x^{(k+1)} - x^{(k)})$$

Chování v závislosti na $\phi'(\alpha)$:

- $\phi'(\alpha) \approx 1$: špatný odhad chyby
- $\phi'(\alpha) = 0$: přesný odhad (Newtonova metoda)
- $\phi'(\alpha) < 0$: přijatelný odhad, pokud $|\phi'(\alpha)|$ není příliš velké

9 Úvod do nelineárních systémů

Uvažujme systém nelineárních rovnic:

$$F : \mathbb{R}^n \rightarrow \mathbb{R}^m, \quad \text{hledáme } x^* \in \mathbb{R}^n \text{ takové, že } F(x^*) = 0$$

Problém minimalizace funkce $f : \mathbb{R}^m \rightarrow \mathbb{R}$ vede na řešení:

$$F(x) = \nabla f(x) = 0$$

kde F představuje gradient funkce f .

Definice 9.1 (Jacobianova matice). *Pro $x = (x_1, \dots, x_n)^T$, $F : D \subseteq \mathbb{R}^n \rightarrow \mathbb{R}^n$ definujeme Jacobianovu matici:*

$$(J_F(x))_{i,j} = \frac{\partial F_i}{\partial x_j}(x), \quad i, j = 1, \dots, n$$

9.1 Newtonova metoda

Základní algoritmus Iterační proces:

$$x^{(k+1)} = x^{(k)} - (J_F(x^{(k)}))^{-1} F(x^{(k)})$$

Věta 9.1. *Nechť $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ je $C^1(D)$ na konvexní otevřené množině $D \subset \mathbb{R}^n$. Nechť $x_\alpha \in D$ splňuje $F(x_\alpha) = 0$. Předpokládejme, že:*

- *Jacobiho matice $J_F^{-1}(x)$ existuje (je regulární)*
- *Existují konstanty $R, C, L > 0$ takové, že:*

$$\|J_F^{-1}(x)\| \leq C \quad \text{a} \quad \|J_F(x) - J_F(y)\| \leq L\|x - y\|, \quad \forall x, y \in B(x_\alpha, R)$$

Potom existuje $r_0 > 0$, že $\forall x^{(0)} \in B(x_\alpha, r)$, Newtonova posloupnost $\{x^{(k)}\}$ konverguje k x_α s odhadem:

$$\|x^{(k+1)} - x_\alpha\| \leq CL\|x^{(k)} - x_\alpha\|^2$$

Kvadratická (2. řádu) konvergence.

Proof. Indukcí. Pro $k = 0$ nejprve ukážeme, že $J_F^{-1}(x^{(0)})$ existuje.

Pokud $\|A\| < 1$, pak $(I - A)^{-1}$ existuje a platí:

$$\|(I - A)^{-1}\| \leq \frac{1}{1 - \|A\|}$$

Definujme $A := J_F^{-1}(x_\alpha)J_F(x^{(0)}) - I$ a odhadněme normu:

$$\|A\| \leq \|J_F^{-1}(x_\alpha)\| \|J_F(x^{(0)}) - J_F(x_\alpha)\| \leq CL\|x^{(0)} - x_\alpha\| \leq CLr$$

Zvolíme $r > 0$ tak, že $CLr < \frac{1}{2} \iff r < \frac{1}{2CL}$, a tedy $I - A$ je invertovatelné $\iff J_F^{-1}(x^{(0)})$ existuje. Odhadneme normu inverzní matice:

$$\|J_F^{-1}(x^{(0)})\| = \|J_F^{-1}(x_\alpha)(I - A)^{-1}\| \leq \|J_F^{-1}(x_\alpha)\| \|(I - A)^{-1}\| \leq \frac{C}{1 - \|A\|} \leq 2C \quad (\text{protože } \|A\| \leq \frac{1}{2})$$

Nyní je iterace dobře definována:

$$x^{(1)} = x^{(0)} - J_F^{-1}(x^{(0)})F(x^{(0)})$$

Taylorův rozvoj kolem x_α nám dá:

$$F(x_\alpha) \approx F(x^{(0)}) + J_F(x^{(0)})(x_\alpha - x^{(0)}) + \mathcal{O}(\|x_\alpha - x^{(0)}\|^2)$$

Odhad chyby:

$$\|x^{(1)} - x_\alpha\| \leq \|J_F^{-1}(x^{(0)})\| \cdot \frac{L}{2} \|x^{(0)} - x_\alpha\|^2 \leq 2C \cdot \frac{L}{2} \|x^{(0)} - x_\alpha\|^2 = CL \|x^{(0)} - x_\alpha\|^2$$

Protože $CLr \leq \frac{1}{2}$, dostáváme:

$$\|x^{(1)} - x_\alpha\| \leq \frac{1}{2} \|x^{(0)} - x_\alpha\| \implies x^{(1)} \in B(x_\alpha, r)$$

Opakováním stejných argumentů pro $k > 1$ dokončíme důkaz věty. □

9.2 Kořeny polynomů

Uvažujme polynom stupně m :

$$p_n(x) = \sum_{j=0}^n a_j x^j, \quad a_n \neq 0$$

Faktorizovaný tvar:

$$p_n(x) = a_n(x - \alpha_1)^{m_1}(x - \alpha_2)^{m_2} \cdots (x - \alpha_k)^{m_k}, \quad \sum_{j=1}^k m_j = n$$

kde m_j je násobnost kořene α_j .

Vlastnost 9.1. Pokud jsou koeficienty $a_j \in \mathbb{R}$, pak buď $\alpha_j \in \mathbb{R}$, nebo $\overline{\alpha_j}$ je také kořen.

9.3 Hornerovo schéma

Efektivní reprezentace polynomu:

$$p_n(x) = a_0 + x(a_1 + x(a_2 + \cdots + x(a_{n-1} + a_n x) \cdots))$$

Zefektivňuje výpočty s polynomy – stačí jen $O(n)$ operací.

Algoritmus. Algoritmus pro vyhodnocení polynomu p_n v bodě z :

- 1: $b_m \leftarrow a_m$
- 2: **for** $k = m - 1$ downto 0 **do**
- 3: $b_k \leftarrow a_k + b_{k+1}z$
- 4: **end for**

Výsledek: $b_0 = p_n(z)$

Definujeme přidružený polynom: $q_{n-1}(x; z) = b_1 + b_2 x + \cdots + b_n x^{n-1}$

Pro polynomy $h_n(x)$, $g_m(x)$, kde $n > m$, existují jednoznačné polynomy $\delta(x)$ a $g(x)$ takové, že:

$$h_n(x) = g_m(x)\delta(x) + g(x)$$

Speciální případ:

$$p_n(x) = (x - z)q_{n-1}(x; z) + b_0$$

9.3.1 Deflační postup

1. Najdi přibližný kořen α
2. Vypočti koeficienty $q_{n-1}(x; \alpha)$
3. Polož $p_{n-1}(x) = q_{n-1}(x; \alpha)$
4. Opakuj pro polynom nižšího stupně

Pokud z je kořen, pak:

$$p_n(x) = (x - z)q_{n-1}(x; z)$$

9.4 Newton-Hornerova metoda

Kombinace Newtonovy metody a Hornerova schématu:

1. Zvol počáteční odhad $r_j^{(0)}$ blízko kořene r_j
2. Iteruj podle Newtonovy metody:

$$r_j^{(k+1)} = r_j^{(k)} - \frac{p_n(r_j^{(k)})}{p'_n(r_j^{(k)})},$$

kde p'_n spočítáme jako $p'_n(x) = \frac{d}{dx}(b_0 + (x - z)q_{n-1}(x; z)) = q_{n-1}(x; z) + (x - z)q'_{n-1}(x; z) \implies p'_n(z) = q_{n-1}(z; z)$. A tedy

$$r_j^{(k+1)} = r_j^{(k)} - \frac{p_n(r_j^{(k)})}{q_{n-1}(r_j^{(k)}; r_j^{(k)})}$$

Pro každé k je třeba počítat $q_{n-1}(x; r_j^{(k)})$

- Jakmile je $r_j^{(k)}$ dostatečně blízko k r_j , stop Newton.
- Po dosažení přesnosti: deflace.
- Nyní má q_{n-1} menší stupeň než p_n , opakujme Newtona na $q_{n-1}(x; r_j^{(k)})$.

Výpočetní náročnost: $4(n - j)$ flopů pro j -tý kořen

9.4.1 Zpřesňování výsledků

- Začni s kořeny nejmenší velikosti.
- Pro zpřesnění použij Newtonovu metodu na původní polynom.
- Stabilita: q_{n-1} může být citlivá na chyby.

10 Aproximace

Možné aproximace funkce $f(x)$:

- Polynomy: $\{1, x, x^2, \dots\}$
- Goniometrické funkce: $\{\sin(x), \cos(x), \sin(2x), \cos(2x), \dots\}$
- Racionální funkce: $p(x)/q(x)$
- Exponenciální funkce: ae^{bx} , $a, b \in \mathbb{C}$

10.1 Polynomiální aproximace

Prostor polynomů stupně $\leq n$:

$$P_n := \{q(x) \mid q(x) \text{ je polynom s reálnými koeficienty}\}$$

Interpolace. Chceme $f(x_i) = p(x_i)$ pro uzly $x_i \in [a, b]$.

Metoda nejmenších čtverců.

$$\int_a^b (f(x) - p(x))^2 d\omega(x) = \min_{q \in P_n} \int_a^b (f(x) - q(x))^2 d\omega(x)$$

Diskrétní verze:

$$\sum_{j=0}^n \omega_j (f(x_j) - p_n(x_j))^2 = \min_{q \in P_n} \sum_{j=0}^n \omega_j (f(x_j) - q(x_j))^2$$

Čebyševova aproximace.

$$\max_{[a,b]} |f(x) - p_n(x)| = \min_{q \in P_n} \max_{[a,b]} |f(x) - q(x)|$$

10.2 Lagrangeova interpolace

Pro $n+1$ bodů (x_i, y_i) najdeme $p_n \in P_n$ splňující $p_n(x_i) = y_i$. Lagrangeův interpolační polynom stupně n definujeme jako:

$$l_i(x) = \prod_{\substack{k=0 \\ k \neq i}}^n \frac{x - x_k}{x_i - x_k},$$

Pokud $i = j$, pak $l_i(x_j) = 1$, jinak 0. Nechť $L_n(x) \in P_n$ je jednoznačný interpolační polynom:

$$L_n(x) := \sum_{i=0}^n y_i l_i(x) \implies L_n(x_j) = y_j$$

Věta 10.1 (O jednoznačnosti interpolace). Pro $n+1$ různých bodů $x_0, \dots, x_n$ a $n+1$ odpovídajících hodnot $y_0, \dots, y_n$, existuje právě jeden interpolující polynom $L_n(x) \in P_n$, že $L_n(x_i) = y_i$, $i = 0, \dots, n$.

Proof. Předpokládejme, že existuje polynom $P_n(x) \neq L_n(x)$ takový, že:

$$P_n(x_i) = y_i, \quad i = 0, \dots, n$$

a $P_n \in \mathcal{P}_n$ (tj. je to polynom stupně nejvýše n). Potom:

- Rozdíl $L_n(x) - P_n(x)$ je polynom stupně nejvýše n
- Platí $L_n(x_i) - P_n(x_i) = 0$ pro $i = 0, \dots, n$

Z toho vyplývá: $L_n(x) - P_n(x) = 0 \quad \forall x \implies L_n(x) \equiv P_n(x)$. □

10.3 Interpolační chyba

Předpokládejme, že hodnoty $y_0, \dots, y_m$ jsou dány jako $y_i = f(x_i)$, kde f je dostatečně hladká funkce.

Věta 10.2. Pro $f \in C^{n+1}(I)$ a $x_i \in I$ platí:

$$E_n(x) = f(x) - L_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega_{n+1}(x), \quad x \in I$$

kde $\omega_{n+1}(x) := \prod_{k=0}^n (x - x_k)$ a $\xi \in I$.

Proof. Pro $x = x_i$ triviální. Předpokládejme, že $x \neq x_i$ a fixujme I .

Uvažujme funkci:

$$G(t) := E_n(t) - \omega_{n+1}(t) \frac{E_n(x)}{\omega_{n+1}(x)}.$$

Jelikož $f \in C^{n+1}(I)$ a $E(t) \in C^{n+1}(I) \implies G(t) \in C^{n+1}(I)$ a má $n+2$ různých kořenů (x_i a x).

Použitím věty: “Pokud $f \in C'([a, b])$, pak $\exists c \in (a, b) : f'(c) = \frac{f(b)-f(a)}{b-a}$.” dostaneme:

$G'(t)$ má $n+1$ kořenů

$G''(t)$ má n kořenů

$\vdots$

$G^{(n+1)}(t)$ má alespoň 1 kořen $\xi \in I$

Protože:

$$E_n^{(n+1)}(t) = f^{(n+1)}(t), \quad \omega_n^{(n+1)}(t) = (n+1)!,$$

tak dostaneme:

$$G^{(n+1)}(t) = f^{(n+1)}(t) - \frac{(n+1)!}{\omega_{n+1}(x)} E_n(x)$$

Pro $t = \xi$ získáme tvrzení věty. □

Chyba interpolace.

$$E_n(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \omega_{n+1}(x)$$

Problémy:

- $\omega_{n+1}(x)$ může růst s $n \rightarrow \infty$.
- $f^{(n+1)}(s)$ může rychle růst.

10.3.1 Rungeův jev

Příklad (Runge). Předpokládejme, že máme funkci:

$$f(x) = \frac{1}{1 + 25x^2}, \quad I = [-1, 1]$$

s ekvidistantními uzly $x_j = \frac{2j}{n} - 1, \quad j = 0, \dots, n$. Dále předpokládejme:

$$\|f - L_n\|_\infty := \max_{x \in I} |f(x) - L_n(x)| \leq \max_{x \in I} \frac{|f^{(n+1)}(x)|}{(n+1)!} \max_{x \in I} |\omega_{n+1}(x)|$$

Pro ekvidistantní uzly:

$$\|\omega_{n+1}\|_\infty \leq n! \left(\frac{2}{n}\right)^{n+1} \rightarrow \infty$$

10.3.2 Řešení: Čebyševovy uzly

$$X_\alpha = \cos\left(\frac{2\alpha - 1}{2n}\pi\right), \quad \alpha = 0, \dots, n,$$

kde X_α jsou kořeny $n + 1$ čebyševových polynomů. Může být dokázáno, že:

$$\|\omega_{n+1}(x)\|_\infty = \frac{1}{2^{n-1}}.$$

10.4 Po částech polynomiální interpolace

Interval I rozdělen na k podintervalů $I_j = [x_j, x_{j+1}]$. Následně interpolujeme funkci f na každém I_j za pomoci $n + 1$ uzlů $x_j^{(i)}$, $i = 0, \dots, n$.

Problém: aproximace není diferencovatelná.

11 Splines

“Můžeme slepit dohromady na části rozdělené polynomy, abychom získali funkci $C^{(k)}(I)$? “

Definice 11.1 (Spline). Necht $a = x_0 < x_1 < \dots < x_n = b$ jsou různé uzly na $I = [a, b]$. Funkce $S_k(x)$ je **spline stupně k** vzhledem k uzlům x_i pokud:

$$S_k|_{[x_j, x_{j+1}]} \in P_k, \quad j = 0, \dots, m-1, \quad S_k \in C^{k-1}(I)$$

11.1 Podmínky pro interpolační spliny

Interpolační podmínky: $S_k(x_j) = y_j, \quad j = 0, \dots, n.$

$$\text{Podmínky spojitosti: } \begin{cases} S_{k,j-1}(x_j) = S_{k,j}(x_j) \\ S'_{k,j-1}(x_j) = S'_{k,j}(x_j) \\ S''_{k,j-1}(x_j) = S''_{k,j}(x_j) \end{cases} \quad \text{pro } j = 1, \dots, n-1.$$

Kde $S_{k,j}(x) = S_k|_{[x_j, x_{j+1}]}$ je polynom stupně k .

Periodické spliny.

$$S_k^{(m)}(a) = S_k^{(m)}(b), \quad m = 0, \dots, k-1$$

Přirozené spliny. Pro $k = 2l - 1, l \geq 2$:

$$S_k^{(l+j)}(a) = S_k^{(l+j)}(b) = 0, \quad j = 0, \dots, l-2$$

11.2 Kubické spliny ($k = 3$)

$$S_3 \in C^2([a, b])$$

Interpoluje body $(x_0, y_0), \dots, (x_n, y_n)$ a chceme určit S_3 :

$$S_3(x_i) = y_i, \quad S'_3(x_i) = m_i, \quad S''_3(x_i) = M_i, \quad i = 0, \dots, n,$$

kde y_i je dáno a m_i, M_i musíme dopočítat.

Restrikce $S_{3,i-1}(x) \in P_3 \implies S''_{3,i-1} \in P_1$. Na intervalu $[x_{i-1}, x_i]$:

$$S''_{3,i-1}(x) = M_{i-1} \frac{x_i - x}{h_i} + M_i \frac{x - x_{i-1}}{h_i},$$

kde $h_i = x_i - x_{i-1}$. Integrací dostaneme:

$$S_{3,i-1}(x) = M_{i-1} \frac{(x_i - x)^3}{6h_i} + M_i \frac{(x - x_{i-1})^3}{6h_i} + C_{i-1}(x - x_{i-1}) + D_{i-1}$$

Konstanty C_{i-1}, D_{i-1} určíme z:

$$S_{3,i-1}(x_{i-1}) = y_{i-1}, \quad S_{3,i-1}(x_i) = y_i$$

Podmínka spojitosti v uzlech:

$$S'_{3,i-1}(x_i) = S'_{3,i}(x_i)$$

Dostáváme soustavu rovnic:

$$\mu_i M_{i-1} + 2M_i + \lambda_i M_{i+1} = d_i, \quad i = 1, \dots, n-1$$

kde:

$$\mu_i = \frac{h_i}{h_i + h_{i+1}}, \quad \lambda_i = \frac{h_{i+1}}{h_i + h_{i+1}}$$

$$d_i = \left(\frac{y_{i+1} - y_i}{h_{i+1}} - \frac{y_i - y_{i-1}}{h_i} \right) \frac{6}{h_i + h_{i+1}}$$

Přidáme okrajové podmínky:

$$2M_0 + \lambda_0 M_1 = d_0$$

$$\mu_m M_{n-1} + 2M_n = d_n$$

Matice soustavy je diagonálně dominantní:

$$\begin{bmatrix} 2 & \lambda_0 & & & \\ \mu_1 & 2 & \lambda_1 & & \\ & \ddots & \ddots & \ddots & \\ & & \mu_{m-1} & 2 & \lambda_{m-1} \\ & & & \mu_m & 2 \end{bmatrix} \begin{bmatrix} M_0 \\ M_1 \\ \vdots \\ M_{m-1} \\ M_m \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_{m-1} \\ d_m \end{bmatrix}$$

Vlastnost 11.1. Pro $f \in C^2([a, b])$ a kubický spline S_3 interpolující f platí:

1. Pro přirozený spline ($S_3''(a) = S_3''(b) = 0$)
2. Nebo spline s danými derivacemi ($S_3'(a) = f'(a)$, $S_3'(b) = f'(b)$)

pak platí:

$$\int_a^b (S_3''(x))^2 dx \leq \int_a^b (f''(x))^2 dx$$

Důsledek 11.1. Kubické spliny minimalizují "energii" křivky.

11.3 Splines in Parametric Form

Mějme křivku vyjádřenou v parametrickém tvaru:

$$P(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}, \quad t \in [0, T]$$

kde $x(t)$ a $y(t)$ jsou funkce parametru t . Křivku lze vzorkovat v bodech $P_i = \begin{bmatrix} x_i \\ y_i \end{bmatrix}$, kde $x_i = x(t_i)$, $y_i = y(t_i)$, pro $i = 0, \dots, n-1$ a $0 \leq t_0 \leq \dots \leq t_n \leq T$.

11.4 De Casteljauovo schéma

De Casteljauův algoritmus konstruuje Bézierovy křivky pomocí rekurzivní lineární interpolace:

$$\begin{cases} P_i^{(0)}(t) = P_i \in \mathbb{R}^2, & i = 0, \dots, n-1, & t \in [0, 1], \\ P_i^{(j)}(t) = (1-t)P_i^{(j-1)}(t) + tP_{i+1}^{(j-1)}(t), & j = 0, \dots, n-1, & i = 0, \dots, n-j-1. \end{cases}$$

Geometrická reprezentace:

$$\begin{array}{ccccccc} & & & & & & P_0 \\ & & & & & & \searrow \\ & & & & & P_1 & \searrow P_0^{(1)} \\ & & & & & \searrow & \searrow \\ & & & & P_2 & \searrow P_1^{(1)} & \searrow P_0^{(2)} \\ & & & & \searrow & \searrow & \searrow \\ & & & P_3 & \searrow P_2^{(1)} & \searrow P_1^{(2)} & \searrow P_0^{(3)} \\ & & & \vdots & \ddots & \ddots & \ddots \end{array}$$

Vlastnosti:

- $P_i^{(j)}(t)$ je polynomiální křivka stupně j .
- Křivka je hladká.
- Interpoluje první a poslední řídicí body.
- Žádné lokální ovládání: změna jednoho řídicího bodu ovlivní celou křivku.
- Vlastnost konvexního obalu: křivka leží uvnitř konvexního obalu řídicích bodů.

Vzorec Bézierových křivek. Polynom $P^{(n)}(t)$ generovaný De Casteljaovým algoritmem se nazývá Bézierova křivka:

$$P^{(n)}(t) = B_n(t) := \sum_{k=0}^n P_k b_{n,k}(t), \quad 0 \leq t \leq 1$$

kde Bernsteinovy polynomy jsou dány vztahem:

$$b_{n,k}(t) = \binom{n}{k} t^k (1-t)^{n-k} = \frac{n!}{k!(n-k)!} t^k (1-t)^{n-k}.$$

12 Orthogonal Polynomials

Mějme váhovou funkci $\omega(x) \geq 0$ na intervalu (a, b) . Definujeme:

$$I_\omega(f) := \int_a^b f(x)\omega(x) dx, \quad (f, g)_\omega := \int_a^b f(x)g(x)\omega(x) dx, \quad \|f\|_\omega := \sqrt{(f, f)_\omega}.$$

Prostor $L_\omega^2(a, b)$ tvoří všechny funkce $f : (a, b) \rightarrow \mathbb{R}$ splňující: $\int_a^b f^2(x)\omega(x) dx < \infty$.

Definice 12.1 (Ortogonalita). *Posloupnost polynomů $p_0, p_1, p_2, \dots$ (kde $\deg(p_j) = j$) je **ortogonální** vzhledem k $\omega(x)$, pokud:*

$$(p_i, p_j)_\omega = 0 \quad \text{pro } i \neq j.$$

Pokud jsou p_j monické, tak platí rekurence:

$$p_{k+1}(x) = xp_k(x) + \sum_{j=0}^k \gamma_j p_j(x),$$

$$\text{kde } \gamma_k = -\frac{(xp_k, p_k)_\omega}{(p_k, p_k)_\omega}.$$

12.1 Three-term rekurence

$$p_{-1}(x) = 0, \quad p_0(x) = 1$$

$$P_{k+1}(x) = (x - \delta_k)p_k(x) - \beta_k p_{k-1}(x), \quad k = 0, 1, \dots,$$

$$\text{kde } \alpha_k = \frac{(xp_k, p_k)_\omega}{(p_k, p_k)_\omega}, \quad \beta_k = \frac{(xp_{k-1}, p_k)_\omega}{(p_{k-1}, p_{k-1})_\omega}.$$

12.2 Zobecněná Fourierova řada

Pro libovolnou funkci $f \in L_\omega^2$ a $\hat{f}$ fourierovy koeficienty, můžeme sestavit její rozvoj:

$$(Sf)(x) = \sum_{k=0}^{\infty} \hat{f}_k p_k(x), \quad \hat{f}_k = \frac{(f, p_k)_\omega}{\|p_k\|_\omega^2}.$$

Částečný součet $f_n(x) = \sum_{k=0}^n \hat{f}_k p_k(x)$ konverguje k f v L_ω^2 :

$$\|f - f_n\|_\omega \rightarrow 0 \quad \text{pro } n \rightarrow \infty.$$

Díky Parsevalova rovnosti, máme:

$$\|f\|_w^2 = \sum_{k=0}^{+\infty} \hat{f}_k^2 \cdot \|p_k\|_w^2$$

A dále, pro všechna n , platí:

$$\|f - f_n\|_w^2 = \left\| \sum_{k=n_1}^{+\infty} \hat{f}_k p_k(x) \right\|_w^2 = \sum_{k=n_1}^{+\infty} \hat{f}_k^2 \|p_k\|_w^2$$

Pro polynom $f_n \in P_n$ platí následující minimalizační vlastnost:

Vlastnost 12.1 (Ortogonalita $\iff$ Optimalita).

$$\|f - f_n\|_w = \min_{q \in P_n} \|f - q\|_w$$

Proof. Mějme $f - f_n = \sum_{k=n+1}^{\infty} \hat{f}_k p_k$. Potom $\forall q \in P_n$ platí $(f - f_n, q) = 0$. Proto $f - f_n$ je ortogonální k P_n . Protože $f_n - q \in P_n$, dostáváme:

$$\|f - q\|_w^2 = \|\underbrace{f - f_n}_{\perp_w P_n} + \underbrace{f_n - q}_{\in P_n}\|_w^2 = \|f - f_n\|_w^2 + \|f_n - q\|_w^2 \geq \|f - f_n\|_w^2$$

Z toho plyne:

$$\|f - f_n\|_w \leq \|f - q\|_w, \quad \forall q \in P_n$$

Navíc platí:

$$\|f - f_n\|_w = \|f - q\|_w \iff \|f_n - q\|_w = 0 \iff f_n = q.$$

□

12.3 Chebyshev Polynomials

Pro $\omega(x) = \frac{1}{\sqrt{1-x^2}}$ na $(-1, 1)$, Chebyshev polynomials $T_k(x) = \cos(k \arccos(x))$ jsou ortogonální:

$$\int_{-1}^1 T_k(x) T_j(x) \frac{dx}{\sqrt{1-x^2}} = 0 \quad \text{pro } k \neq j.$$

Normy jsou:

$$\|T_k\|_{\omega}^2 = \begin{cases} \pi & \text{pokud } k = 0, \\ \pi/2 & \text{pokud } k \neq 0. \end{cases}$$

Chebyshevovo rozšíření:

$$(Cf)(x) = \sum_{k=0}^{\infty} \hat{f}_k T_k(x), \quad \hat{f}_k = \frac{1}{c_k} \int_{-1}^1 f(x) T_k(x) (1-x^2)^{-1/2} dx.$$

Three-term rekurence:

$$T_{k+1}(x) = 2xT_k(x) - T_{k-1}(x), \quad T_0(x) = 1, \quad T_1(x) = x.$$

13 Numerická Integrace

Pro integrovatelnou funkci f na intervalu $[a, b]$:

$$I(f) = \int_a^b f(x) dx$$

13.1 Quadrature rules.

Aproximují $I(f)$ vyčíslením f ve vybraných bodech. Obecný přístup nahrazuje f aproximací f_n , kterou lze snadněji integrovat:

$$I_n(f) = \int_a^b f_n(x) dx, \quad E_n(f) = I(f) - I_n(f)$$

Chyba obecně splňuje:

$$|E_n(f)| \leq \left| \int_a^b (f(x) - f_n(x)) dx \right| \leq (b-a) \|f - f_n\|_\infty$$

13.1.1 Interpolatory Quadrature

Pro $f_n \in P_n$ má kvadrurní vzorec *stupeň přesnosti (degree of exactness) k* , pokud:

$$I_n(f) = I(f) \quad \forall f \in P_k$$

Interpolární kvadratura nahrazuje f jejím interpolačním polynomem přes $n+1$ bodů. Výsledný vzorec má stupeň přesnosti n .

13.1.2 Rectangular Rule

Interpolace f polynomem nulového stupně (konstanta) v bodě $x_0 \in [a, b]$:

$$I_n(f) = \int_a^b f(x_0) dx = f(x_0)(b-a)$$

Tento vzorec aproximuje integrál jako obsah obdélníku.

13.1.3 Midpoint Rule

Zvolíme $x_0 = \frac{a+b}{2}$. Chybu lze analyzovat pomocí Taylorova rozvoje. Předpokládejme primitivní funkci $F(x) = \int_a^x f(t) dt$. Rozšíříme $F(a+h)$ přes a a získáme tak:

$$F(a+h) = F(a) + F'(a)h + \frac{F''(a)}{2}h^2 + \frac{F'''(\xi)}{3!}h^3 = 0 + f(a)h + \frac{f'(a)}{2}h^2 + \frac{f''(\xi)}{6}h^3.$$

$$\implies E_0(f) = \int_a^b f(t) dt - hf\left(\frac{a+b}{2}\right).$$

Když rozepíšeme $f(\frac{a+b}{2})$ přes a a získáme tak: $f(\frac{a+b}{2}) = f(a) + f'(a)\frac{h}{2} + f''(\xi)\frac{h^2}{8}$. Všimněme si, že $a - \frac{a+b}{2} = \frac{b-a}{2} = \frac{h}{2} \implies E_0(f) = \frac{h^3}{2} \left(\frac{f''(\xi)}{3} + \frac{f''(\xi)}{4} \right)$. Stupeň přesnosti: 1.

13.1.4 Composite Midpoint Rule

For uniform partition $x_i = a + ih$, $h = \frac{b-a}{m}$:

$$I_{0,m}(f) = h \sum_{k=0}^{m-1} f\left(x_k + \frac{h}{2}\right), \quad E_{0,m}(f) = \frac{b-a}{24} h^2 f''(\xi)$$

13.1.5 Trapezoidal Rule

For $n = 1$ with nodes $x_0 = a$, $x_1 = b$:

$$I_1(f) = \frac{b-a}{2} (f(a) + f(b)), \quad E_1(f) = -\frac{(b-a)^3}{12} f''(\xi)$$

Degree of exactness: 1.

13.1.6 Composite Trapezoidal Rule

For uniform partition:

$$I_{1,m}(f) = h \left(\frac{f(x_0)}{2} + \sum_{k=1}^{m-1} f(x_k) + \frac{f(x_m)}{2} \right), \quad E_{1,m}(f) = -\frac{b-a}{12} h^2 f''(\xi)$$

13.1.7 Simpson's Rule

For $n = 2$ with nodes $x_0 = a$, $x_1 = \frac{a+b}{2}$, $x_2 = b$:

$$I_2(f) = \frac{b-a}{6} \left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right), \quad E_2(f) = -\frac{(b-a)^5}{2880} f^{(4)}(\xi)$$

Degree of exactness: 3.

13.1.8 Composite Simpson's Rule

For uniform partition:

$$I_{2,m}(f) = \frac{h}{6} \left(f(x_0) + 2 \sum_{k=1}^{m-1} f(x_{2k}) + 4 \sum_{k=1}^m f(x_{2k-1}) + f(x_{2m}) \right), \quad E_{2,m}(f) = -\frac{b-a}{180} \left(\frac{h}{2} \right)^4 f^{(4)}(\xi)$$

13.2 Newton-Cotes Formulas

Pro $m+1$ interpolačních bodů $x_0 < \dots < x_m$ má kvadratura tvar:

$$I_m(f) = \sum_{j=0}^m f(x_j) \omega_j, \quad \omega_j = \int_a^b l_j(x) dx$$

kde l_j jsou Lagrangeovy báze polynomy. Chyba pro uzavřené Newton-Cotesovy vzorce je:

$$E_m(f) = \frac{M_m}{(m+2)!} h^{m+3} f^{(m+2)}(\xi), \quad M_m \in \mathbb{R}$$

13.3 Gauss Quadrature

Maximalizace stupně přesnosti výběrem uzlů jako kořenů ortogonálních polynomů pro váhovou funkci $w(x)$. Výsledná Gaussova kvadratura:

$$I_n(f) = \sum_{j=0}^n f(x_j) \omega_j,$$

dosahuje stupně přesnosti $2n+1$ (maximální možný). Váhy ω_j jsou kladné a dány vztahem:

$$\omega_j = \frac{\int_a^b l_j(x) w(x) dx}{l_j(x_j)}$$

Lemma 13.1. *Váhy ω_j Gaussovy kvadratury jsou pozitivní.*

Proof. Předpokládejme $p_i(x) = \prod_{i=0, i \neq j}^n (x - x_j)^2$. To má stupeň $2(n+1) - 2 = 2n < 2n+1$.

Odtud $I(p_i) = \sum_{j=0}^n p_i(x_j) \omega_j = p_i(x_i) \omega_i$.

$$I(p_i) = \int_a^b p_i(x) d\omega(x) > 0 \implies \omega_j = \frac{I(p_0)}{p_i(x_i)} > 0$$

Pro $p_i(x) > 0, \forall x$.

□