Iterační metody v lineární algebře

Karel Vacek, Jan Valášek

October 23, 2026

Obsah

- Úvod
- Motivace
- Řešení soustav $Ax = b$
- Úvod do řídkých matic + násobení vektoru s matici
- Praktická část
- (Demoverze GMRES)

Motivace pro Iterační Řešiče

- **Tradiční Lineární Řešiče:**
- Gausova eliminace, LU rozklad, Cholezkeho rozklad
- **Iterační Řešiče:**
- Gauss-Seidel, Jacobi, Successive Over-Relaxation (SOR), Conjugate Gradient (CG), Generalized Minimal RESidual (GMRES), BiConjugate Gradient Stabilized (BiCGStab), Preconditioned Conjugate Gradient (PCG), Orthomin(m) (MINRES),

Výhody Iterativních Solverů:

- **Efektivita Paměti**
- **Použitelnost pro Velké a Řídké Systémy**
- **Jednoduchá Paralelizace**

Nevýhody Iterativních Solverů:

- **Citlivost na Konvergenční Rychlost**
- **Závislost na Počátečním Odhadu**
- **Omezená Použitelnost** (Symetrické Pozitivně Definitní Matice)

Formát Seznamu Souřadnic (COO)

Popis: Formát Seznamu Souřadnic je další způsob, jak reprezentovat řídké matice. Ukládá hodnoty nenul a jejich řádkové a sloupcové indexy.

Uložení:

- řádek: Pole obsahující řádkové indexy nenulových prvků.
- sloupec: Pole obsahující sloupcové indexy nenulových prvků.
- data: Pole obsahující odpovídající nenulové hodnoty.

$$A = \begin{bmatrix} 2 & 5 & 0 & 0 \\ 0 & 1 & 3 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -3 \end{bmatrix}$$

Formát Seznamu Souřadnic (COO):

$$\begin{aligned} \text{I pole:} &\quad [1, 1, 2, 2, 3, 4] \\ \text{J pole:} &\quad [1, 2, 2, 3, 3, 4] \\ \text{VAL pole:} &\quad [2, 5, 1, 3, 1, -3] \end{aligned}$$

The Compressed Sparse Row (CSR)

Popis: Formát The Compressed Sparse Row je populární metoda pro ukládání řídkých matic. Ukládá hodnoty nenul, sloupcové indexy a ukazatele na řádky zvlášť. **Uložení:**

- PI: Pole obsahující indexy řádků, kde začíná každý řádek v polích J a VAL.
- J: Pole obsahující sloupcové indexy odpovídajících hodnot v poli VAL.
- VAL: Pole obsahující nenulové hodnoty matice.

Formát CSR:

$$\begin{aligned} \text{PI pole:} &\quad [1,3,5,6,7] \quad \text{(poslední prvek uvádí nnz+1)} \\ \text{J pole:} &\quad [1,2,2,3,3,4] \\ \text{VAL pole:} &\quad [2,5,1,3,1,-3] \end{aligned}$$

Důležité umět vynásobit matici a vektor - rozcvička.

Násobení matice a vektoru v CSR formátu

- CSR (Compressed Sparse Row) formát efektivně ukládá řídké matice pomocí tří polí:
  - `VAL =data` – nenulové prvky matice (v řádcích),
  - `J =indices` – sloupce odpovídající prvkům v `data`,
  - `PI =indptr` – indexy začátků jednotlivých řádků v `data`.
- Násobení matice A ve formátu CSR a vektoru x:

$$y = A\,x$$

se provádí po řádcích:

$$y_i = \sum_{j=A.\texttt{indptr}[i]}^{A.\texttt{indptr}[i+1]-1} A.\texttt{data}[j] \cdot x[A.\texttt{indices}[j]]$$

Konvergence

Dnes implementace:

- Gauss-Seidelova metoda
- Successive over-relaxation (SOR) metoda
- Konjugovane gradienty (CG)

V tabulce níže jsou vlastnosti matice, které zaručují konvergenci dané metody

	GMRES	**SOR**	**Gauss-Seidel**	Jacobi	**CG**
Symetrie	Ne	Ne	Ne	Ne	Ano
Diagonální Dominance	Ne	Ano	Ano	Ano	Ne
SPD	Ne	Ano	Ano	(Někdy)	Ano

Metoda Jacobiho Iterace

Popis: Metoda Jacobiho iterace je iterativní algoritmus pro řešení lineárních soustav rovnic.
Algoritmus: Pro $i = 1, 2, \ldots, n$:

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

to lze přepsat jako

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

Gaussova-Seidelova metoda

Popis: Metoda Gauss-Seidel je iterativní algoritmus, který vylepšuje Jacobiho metodu použitím aktualizovaných hodnot ve stejné iteraci.
Algoritmus: Pro $i = 1, 2, \ldots, n$:

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

to lze přepsat jako

$$x_i^{(k+1)} = x_i^{(k)} + \frac{1}{a_{ii}} \left(b_i - \sum_{j} a_{ij} x_j^{*} \right),$$

kde

$$x_j^* = \begin{cases} x_j^{k+1} & \text{pro } j < i, \\ x_j^k & \text{pro } j \geq i. \end{cases}$$

Metoda SOR

Popis: Metoda Successive Over-Relaxation (SOR) je iterativní algoritmus, který urychluje konvergenci metody Gauss-Seidel.
Algoritmus: Pro $i = 1, 2, \ldots, n$:

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

to lze přepsat jako

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

kde

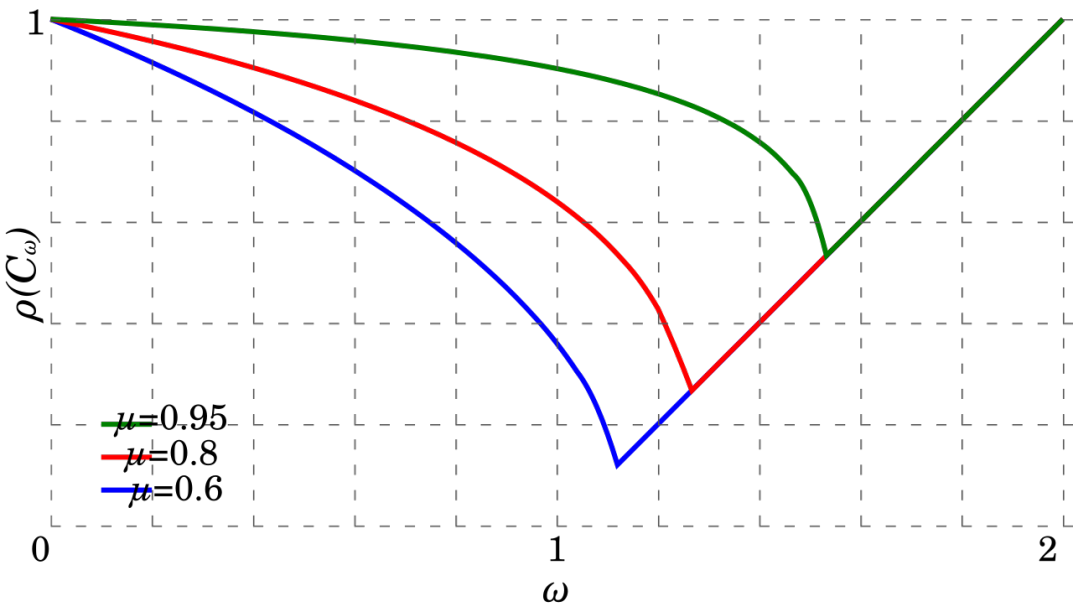
$$x_j^{*} = \begin{cases} x_j^{k+1} & \text{pro } j < i, \\ x_j^{k} & \text{pro } j \geq i. \end{cases}$$

Metoda SOR

Volba ω:

$$\omega_{opt} = 1 + \frac{\mu^2}{4} \in \langle 1, 2 \rangle,$$

kde μ je spektrální poloměr původní iterační matice $\rho(C_{Jacobi})$.



Metoda Konjugovaných Gradientů (CG)

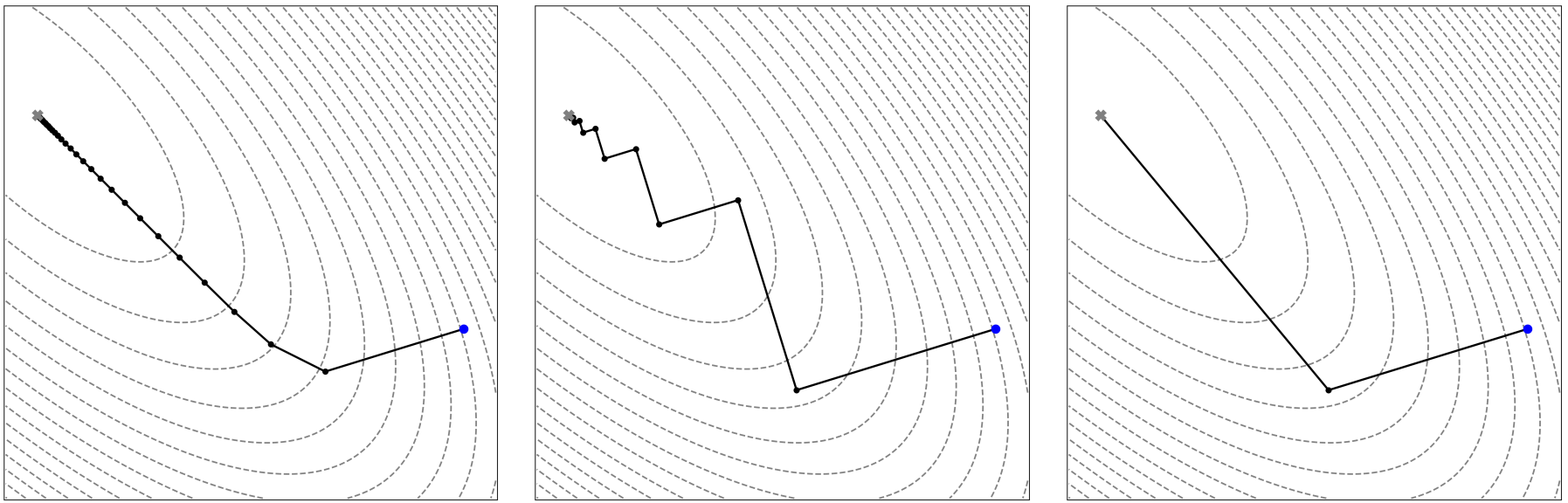
Cíl: Minimalizovat funkci $f(x) = \frac{1}{2}x^T Ax - bx + c$,
která má gradient $\nabla f = Ax - b$, (SPD A značí 1 lok.=>gl. extrém).

.

Směr $r = b - Ax_0 = -\nabla f(x_0)$ tedy ukazuje největší pokles funkce.

Algoritmus graficky:

Gradient descent vs. Steepest descent vs. **Conjugate gradient**



see: mattlanders.net/conjugate-gradient-method

kde volíme délku kroku: a) fixně, b) optimálně, c) upravíme i směr (OG v rámci SS danému maticí A).

Metoda Konjugovaných Gradientů (CG)

Popis: Metoda konjugovaných gradientů (CG) je iterativní algoritmus používaný k řešení systémů lineárních rovnic.

Algoritmus:

Pro danou symetrickou pozitivně definitní matici A a odpovídající vektor pravé strany b algoritmus CG hledá řešení x, které minimalizuje kvadratickou formu $\frac{1}{2}x^T Ax - b^T x$.

1. Vyberte počáteční odhad x_0.
2. Vypočítejte reziduum $r_0 = b - Ax_0$.
3. Nastavte počáteční směr pro vyhledávání $p_0 = r_0$.
4. Pro $k = 0, 1, 2, \ldots$:
   1. Vypočtěte velikost kroku $\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$.
   2. Aktualizujte řešení: $x_{k+1} = x_k + \alpha_k p_k$.
   3. Aktualizujte reziduum: $r_{k+1} = r_k - \alpha_k A p_k$.
   4. Vypočtěte další směr pro vyhledávání: $\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$ a $p_{k+1} = r_{k+1} + \beta_{k+1} p_k$.

Podmíněnost Matice a Konvergence Iterativních Řešičů

- **Podmíněnost Matice A:**
  - Měří citlivost matice na změny ve vstupních datech.
  - Vyšší podmíněnost ($\kappa(A)$) značí větší citlivost matice.
- **Souvislost s Konvergencí:**
  - Vyšší podmíněnost může ztížit konvergenci iterativních řešičů.
  - Metody jako CG nebo GMRES mohou být citlivé na špatně podmíněné matice.
  - Předpodmínění (např., Jacobiho nebo Gauss-Seidelovo) se používá k zlepšení konvergence iterativních metod.

Definice Čísla Podmíněnosti $\kappa(A)$:

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

nebo u SPD matice

$$\kappa_2(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$$

kde $\lambda_{\max}(A)$ a $\lambda_{\min}(A)$ jsou největší a nejmenší vlastní čísla matice A.

Předpodmínění v Iteračních Metodách

- **Předpodmínění:**
  - Představuje modifikaci matice nebo systému rovnic k zlepšení konvergence iterativních metod.
  - Cílem je snížit číslo podmíněnosti celkové matice soustavy, které mohou bránit rychlé konvergenci.
  - Ideální je A^{-1}, ale kvůli výpočetní ceně se volí levná náhrada.
- **Jacobiho Předpodmínění:**
  - Předpodmínění pomocí inverze hlavní diagonály matice.
  - Algoritmus: $M_J^{-1}Ax = M_J^{-1}b$
- **Gauss-Seidel Předpodmínění:**
  - Předpodmínění pomocí části matice pod hlavní diagonálou.
  - Algoritmus: $M_{GS}^{-1}Ax = M_{GS}^{-1}b$
- **A jiné . . . ,** např. ILU.

Metoda GMRES

Popis:
Metoda Generalized Minimal Residual (GMRES) je iterativní algoritmus používaný k řešení velkých, řídkých, nesymetrických systémů lineárních rovnic.

Algoritmus:
Najděte x, které minimalizuje $\|b - Ax\|_2$ ve Krylovově podprostoru $\mathcal{K}_m(A, r_0)$.

Metoda GMRES

GMRES Algorithm:

- Initialize: Choose initial guess x_0, set $r_0 = b - Ax_0$, and normalize $v_1 = r_0/\|r_0\|_2$.
- For $k = 1$ to max_iter:
  - Arnoldi process: Generate $k+1$-dimensional Krylov subspace using Arnoldi orthogonalization.
  - Solve least squares problem: Find y_k minimizing $\|H_k y - \beta e_1\|_2$, where H_k is the upper Hessenberg matrix.
  - Update approximation: $x_k = x_0 + V_k y_k$, where V_k is the matrix containing Arnoldi basis vectors.
  - Update residual: $r_k = b - Ax_k$.
  - Check convergence: If $\|r_k\|_2 < \epsilon$, exit.