1 Iterační metody pro řešení soustav lineárních rovnic

Přímé metody řešení soustav lineárních rovnic (LU, LDLT, Choleského faktorizace atd.) vyžadují $\mathcal{O}(n^3)$ operací a velikost soustavy, kterou jimi dokážeme vyřešit je značně omezená. V případě husté matice $\mathsf{A} \in \mathbb{R}^{n\times n}$ se přibližná velikost řešitelné soustavy historicky vyvíjela takto:

- 1950: $n = 20$ (Wilkinson)
- 1965: $n = 200$ (Forsythe a Moler)
- 1980: $n = 2000$ (LINPACK)
- 1995: $n = 20000$ (LAPACK)
- 2010: $n = 200000$ (HDSS)

Pracujeme-li s řídkými maticemi, jsme schopni řešit i systémy s mnohem větší dimenzí (miliony, desítky milionů neznámých) – zejména, pokud je řešič schopen pracovat paralelně. V případě použití přímého řešiče na řídkou matici však může dojít k jejímu zaplnění. Např. využitím prvního řádku následující matice k vynulování prvního sloupce dojde k zaplnění všech ostatních prvků v matici:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & 0 & 0 & 0 \\ \times & 0 & \times & 0 & 0 \\ \times & 0 & 0 & \times & 0 \\ \times & 0 & 0 & 0 & \times \end{bmatrix} \rightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}$$

Tento problém lze částečně řešit vhodnou pivotizací.

Mezi další nevýhody přímých řešičů patří:

- Známe-li již přibližné řešení soustavy, nedokážeme tuto znalost využít ke snížení celkového počtu operací a zkrácení doby výpočtu.
- Naopak, pokud nám postačuje znalost pouze přibližného řešení, nemůžeme výpočet pomocí přímého řešiče ukončit předčasně.

Alternativou k přímým řešičům jsou iterační řešiče, které generují posloupnost přibližných řešení $\{\boldsymbol{x}^k\}$ a pracují téměř výhradně s násobením matice-vektor, které má náročnost $\mathcal{O}(n^2)$. Důležitou vlastností každé iterační metody je rychlost konvergence posloupnosti $\{\boldsymbol{x}^k\}$ k řešení. Může se totiž stát, že pro některé matice A iterační metoda konverguje velmi pomalu nebo vůbec.

1.1 Lineární iterační metody

Prvním typem iteračních metod, kterým se budeme zabývat, jsou tzv. lineární iterační metody. Ty hledají posloupnost řešení soustavy $\mathsf{A}\boldsymbol{x} = \boldsymbol{b}$ ve tvaru

$$\boldsymbol{x}^{k+1} := \mathsf{M}\boldsymbol{x}^k + \mathsf{N}\boldsymbol{b}, \tag{1.1}$$

kde M a N jsou nějaké matice odpovídajících rozměrů[1].

Definice Lineární iterační metodu nazveme konzistentní, řeší-li rovnici $\mathsf{M}\boldsymbol{x} + \mathsf{N}\boldsymbol{b} = \boldsymbol{x}$ právě jeden vektor $\boldsymbol{x} = \mathsf{A}^{-1}\boldsymbol{b}$. Je možné ukázat, že metoda je konzistentní právě tehdy, je-li splněno $\mathsf{M} = \mathsf{I} - \mathsf{NA}$.

Definice Iterační metodu nazveme konvergentní platí-li $\boldsymbol{x}^k \to \boldsymbol{x} = \mathsf{A}^{-1}\boldsymbol{b}$ pro $k \to \infty$. Je možné ukázat, že metoda je konvergentní, právě tehdy, je-li splněno $\|\mathsf{M}\| < 1$, kde $\|\mathsf{M}\| = \max_{\boldsymbol{v}\in\mathbb{R}^n} \frac{\|\mathsf{M}\boldsymbol{v}\|_v}{\|\boldsymbol{v}\|_v}$ je maticová norma indukovaná vektorovou normou.

Při odvozování následujících iteračních metod budeme využívat rozkladu matice A na součet dolní trojúhelníkové, diagonální a horní trojúhelníhové matice, tedy $\mathsf{A} = \mathsf{L} + \mathsf{D} + \mathsf{U}$ (pozor, nepleťte si matice $\mathsf{L}, \mathsf{D}, \mathsf{U}$ se stejně nazvanými maticemi, které se vyskytovaly u přímých řešičů). Např. matici

$$\mathsf{A} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

rozložíme na

$$\mathsf{L} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \mathsf{D} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \mathsf{U} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$

1.1.1 Jacobiho metoda

Vyjděme z rovnice $\mathsf{A}\boldsymbol{x} = \boldsymbol{b}$. Dosazením $\mathsf{A} = \mathsf{L} + \mathsf{D} + \mathsf{U}$ dostaneme

$$(\mathsf{L} + \mathsf{D} + \mathsf{U})\boldsymbol{x} = \boldsymbol{b}.$$

Roznásobme a přezávorkujme výraz na levé straně

$$\mathsf{D}\boldsymbol{x} + (\mathsf{L} + \mathsf{U})\boldsymbol{x} = \boldsymbol{b}$$

Jacobiho metodu odvodíme tak, že přidáme indexy $k+1$ a k k příslušným vektorům $\boldsymbol{x}$

$$\mathsf{D}\boldsymbol{x}^{k+1} + (\mathsf{L} + \mathsf{U})\boldsymbol{x}^k = \boldsymbol{b}.$$

Osamostatněním $\boldsymbol{x}^{k+1}$ dostaneme předpis pro $k+1$ aproximaci vektoru $\boldsymbol{x}$

$$\boldsymbol{x}^{k+1} := \mathsf{D}^{-1}(\boldsymbol{b} - (\mathsf{L} + \mathsf{U})\boldsymbol{x}^k) = \underbrace{-\mathsf{D}^{-1}(\mathsf{L} + \mathsf{U})}_{=\mathsf{M}}\boldsymbol{x}^k + \underbrace{\mathsf{D}^{-1}}_{=\mathsf{N}}\boldsymbol{x}^k.$$

[1] Index k označuje číslo aktuální iterace.

Jednotlivé složky vektoru $\boldsymbol{x}^{k+1}$ můžeme vyjádřit jako

$$(\boldsymbol{x}^{k+1})_i := \frac{1}{a_{i,i}} \left(b_i - \sum_{j=1, j\neq i}^{n} a_{i,j} (\boldsymbol{x}^k)_j \right) = \tag{1.2}$$

$$= \frac{1}{a_{i,i}} \left(b_i - \sum_{j=1}^{i-1} a_{i,j} (\boldsymbol{x}^k)_j - \sum_{j=i+1}^{n} a_{i,j} (\boldsymbol{x}^k)_j \right) \tag{1.3}$$

Pro snadnější a přehlednější zápis budeme i-tý prvek vektoru $\boldsymbol{x}^{k+1}$ také značit jako x_i^{k+1} (horní index tedy označuje číslo iterace, dolní index značí pořadí prvku ve vektoru).

Konzistence metody vyplývá z jejího odvození, můžeme však ještě ověřit, že $\mathsf{M} = \mathsf{I} - \mathsf{NA}$. V případě Jacobiho metody je $\mathsf{M} = -\mathsf{D}^{-1}(\mathsf{L}+\mathsf{U})$ a $\mathsf{N} = \mathsf{D}^{-1}$ (viz výše). Platí tedy

$$\begin{aligned} \mathsf{I} - \mathsf{NA} &= \mathsf{I} - \mathsf{D}^{-1}\mathsf{A} = \mathsf{I} - \mathsf{D}^{-1}(\mathsf{L}+\mathsf{D}+\mathsf{U}) = \\ &= \mathsf{I} - \mathsf{D}^{-1}(\mathsf{L}+\mathsf{U}) - \mathsf{D}^{-1}\mathsf{D} = -\mathsf{D}^{-1}(\mathsf{L}+\mathsf{U}) = \mathsf{M}. \end{aligned}$$

Metoda je tedy konzistentní.

Lze dokázat, že metoda je konvergentní právě tehdy, když $\|\mathsf{M}\| = \|\mathsf{D}^{-1}(\mathsf{L}+\mathsf{U})\| < 1$. To splňují např. striktně diagonálně dominantní matice (tedy matice, pro které platí $\forall i : |a_{i,i}| > \sum_{j=1, j\neq i}^{n} |a_{i,j}|$). Konvergenci metody pro diagonálně dominantní matice se dá poměrně snadno dokázat. Vyjděme ze vztahu pro i-tý prvek aproximovaného vektoru v iteraci $k+1$:

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

Jelikož je metoda konzistentní, musí tuto rovnost splňovat i prvky vektoru přesného řešení $\boldsymbol{x}$:

$$x_i = \frac{1}{a_{i,i}} \left(b_i - \sum_{j=1, j\neq i}^{n} a_{i,j} x_j \right).$$

Odečteme-li od první rovnosti druhou, dostaneme

$$\underbrace{x^{k+1} - x_i}_{=e_i^{k+1}} = -\frac{1}{a_{i,i}} \sum_{j=1, j\neq i}^{n} a_{i,j} (\underbrace{x_j^k - x_j}_{=e_i^k}),$$

kde $\boldsymbol{e}^k$ je vektor chyby v k-tém kroce. Chybu v kroce $k+1$ můžeme tedy odhadnout pomocí vlastností striktně diagonálně dominantní matice:

$$|e_i^{k+1}| \leq \frac{1}{|a_{i,i}|} \sum_{j=1, j\neq i}^{n} |a_{i,j}||e_j^k| \leq \frac{1}{|a_{i,i}|} \sum_{j=1, j\neq i}^{n} |a_{i,j}| \max_{j=1,\ldots,n, j\neq i} |e_j^k| =$$

$$= \max_{j=1,\ldots,n, j\neq i} |e_j^k| \underbrace{\frac{1}{|a_{i,i}|} \sum_{j=1, j\neq i}^{n} |a_{i,j}|}_{<1} < \max_{j=1,\ldots,n, j\neq i} |e_j^k|$$

Každý prvek vektoru chyby v kroce $k+1$ je tedy v absolutní hodnotě menší než maximální prvek vektoru chyby v předchozím kroce. Vektor chyby tedy konverguje k nulovému vektoru.

1.1.2 Gaussova-Seidelova metoda

Všimněme si, že při výpočtu x_i^{k+1} využíváme v sumě $\sum_{j=1}^{i-1} a_{i,j} x_j^k$ ve výrazu (1.3) pouze prvky $x_1^k, \ldots, x_{i-1}^k$. Tyto prvky tedy můžeme nahradit již vypočtenými prvky aktuálními iterace $x_1^{k+1}, \ldots, x_{i-1}^{k+1}$. Dostaneme tak předpis Gaussovy-Seidelovy metody:

$$x_i^{k+1} := \frac{1}{a_{i,i}} \left(b_i - \sum_{j=1}^{i-1} a_{i,j} x_j^{k+1} - \sum_{j=i+1}^{n} a_{i,j} x_j^k \right) \tag{1.4}$$

Podobně jako v předchozím případě můžeme metodu odvodit, nahradíme-li v soustavě $\mathsf{A}\boldsymbol{x} = \boldsymbol{b}$ matici A součtem $\mathsf{L} + \mathsf{D} + \mathsf{U}$. Tentokrát ovšem vektor s indexem $k+1$ ponecháme u součtu $\mathsf{L} + \mathsf{D}$

$$(\mathsf{L} + \mathsf{D})\boldsymbol{x}^{k+1} = \boldsymbol{b} - \mathsf{U}\boldsymbol{x}^k, \tag{1.5}$$

tedy

$$\boldsymbol{x}^{k+1} = \underbrace{-(\mathsf{L} + \mathsf{D})^{-1}\mathsf{U}}_{=\mathsf{M}} \boldsymbol{x}^k + \underbrace{(\mathsf{L} + \mathsf{D})^{-1}}_{=\mathsf{N}} \boldsymbol{b}.$$

Vztah mezi maticovým zápisem a zápisem po prvcích (1.4) je nejlépe vidět na rovnosti (1.5). Jedná se o soustavu rovnic s dolní trojúhelníkovou maticí $\mathsf{L} + \mathsf{D}$, vektorem pravé strany $\boldsymbol{b} - \mathsf{U}\boldsymbol{x}^k$ a neznámým vektorem $\boldsymbol{x}^{k+1}$. Všimněte si, že výraz (1.4) pak přesně odpovídá algoritmu pro dopřednou substituci pro řešení takovéto soustavy.

Podobně jako v případě Jacobiho metody můžeme ověřit, zda je metoda konzistentní porovnáním $\mathsf{I} - \mathsf{NA}$ a M.

$$\mathsf{I} - \mathsf{NA} = \mathsf{I} - (\mathsf{L} + \mathsf{D})^{-1}\mathsf{A} = \mathsf{I} - (\mathsf{L} + \mathsf{D})^{-1}(\mathsf{L} + \mathsf{D} + \mathsf{U}) =$$
$$= \mathsf{I} - (\mathsf{L} + \mathsf{D})^{-1}(\mathsf{L} + \mathsf{D}) - (\mathsf{L} + \mathsf{D})^{-1}\mathsf{U} = -(\mathsf{L} + \mathsf{D})^{-1}\mathsf{U} = \mathsf{M}.$$

Metoda je tedy konzistentní.

Metoda je konvergentní, právě když $\|(\mathsf{L} + \mathsf{D})^{-1}\mathsf{U}\| < 1$.

1.1.3 Richardsonova metoda

Iterace Richardsonovy metody je dána předpisem

$$\boldsymbol{x}^{k+1} := \boldsymbol{x}^k + \omega \boldsymbol{r}^k,$$

kde $\omega \in \mathbb{R}_+$ a $\boldsymbol{r}^k = \boldsymbol{b} - \mathsf{A}\boldsymbol{x}^k$ je reziduum, které určuje, jak dobře je splněna původní rovnice. Vztah mezi reziduem a chybou $\boldsymbol{e}^k = \boldsymbol{x}^k - \boldsymbol{x}$ lze odvodit přenásobením definice chyby maticí A

$$\mathsf{A}\boldsymbol{e}^k = \mathsf{A}\boldsymbol{x}^k - \mathsf{A}\boldsymbol{x} = \mathsf{A}\boldsymbol{x}^k - \boldsymbol{b} = -\boldsymbol{r}^k.$$

Studujme konvergenci metody pro symetrickou pozitivně definitní matici A. V takovém případě vlastní čísla λ_i a vlastní vektory $\boldsymbol{v}_i$ (tedy skaláry vektory, pro které platí $\mathsf{A}\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i, \|\boldsymbol{v}\| = 1$) splňují

$$0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$$

a

$$\operatorname{span}\{\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n\} = \mathbb{R}^n.$$

Zde span značí lineární obal. Jinými slovy, vlastní vektory tvoří bázi $\mathbb{R}^n$.

Díky předchozímu poznatku můžeme reziduum vyjádřit jako lineární kombinaci prvků báze tvořené vlastními vektory, tedy $\boldsymbol{r}^{k+1} = \sum_{i=1}^n \alpha_i^{k+1} \boldsymbol{v}_i$. Studujme nyní, jak se chová reziduum (a tedy i chyba) v jednotlivých iteracích. Na základě toho se později pokusíme odvodit optimální hodnotu ω pro co nejrychlejší konvergenci.

$$\begin{aligned}
\sum_{i=1}^n \underline{\alpha_i^{k+1}} \boldsymbol{v}_i = \boldsymbol{r}^{k+1} = \boldsymbol{b} - \mathsf{A}\boldsymbol{x}^{k+1} = \underbrace{\boldsymbol{b} - \mathsf{A}(\boldsymbol{x}^k}_{=\boldsymbol{r}^k} + \omega \boldsymbol{r}^k) = \\
= \boldsymbol{r}^k - \mathsf{A}\omega\boldsymbol{r}^k = (\mathsf{I} - \omega\mathsf{A}) \underbrace{\boldsymbol{r}^k}_{=\sum_{i=1}^n \alpha_i^k \boldsymbol{v}_i} = (\mathsf{I} - \omega\mathsf{A}) \sum_{i=1}^n \alpha_i^k \boldsymbol{v}_i = \\
= \sum_{i=1}^n \alpha_i^k \boldsymbol{v}_i - \sum_{i=1}^n \alpha_i^k \omega \underbrace{\mathsf{A}\boldsymbol{v}_i}_{=\lambda_i \boldsymbol{v}_i} = \sum_{i=1}^n \alpha_i^k \boldsymbol{v}_i - \sum_{i=1}^n \alpha_i^k \omega \lambda_i \boldsymbol{v}_i = \\
= \sum_{i=1}^n \underline{(1 - \omega\lambda_i)\alpha_i^k} \boldsymbol{v}_i.
\end{aligned}$$

Všimněme si, že jsme vyjádřili koeficienty rozvoje rezidua $\boldsymbol{r}^{k+1}$ v bázi $\{\boldsymbol{v}_i\}_{i=1}^n$ pomocí násobků koeficientů v předchozím kroce (viz podtržené části předchozího výrazu). Koeficienty se tedy budou zmenšovat (a jednotlivé složky vektoru rezidua budou konvergovat k nule) právě tehdy, když $|1 - \omega\lambda_i| < 1$ pro všechna $i = 1, 2, \ldots, n$. Rychlost konvergence bude záviset na největší hodnotě $|1 - \omega\lambda_i|$. Shrněme tento poznatek do následující věty.

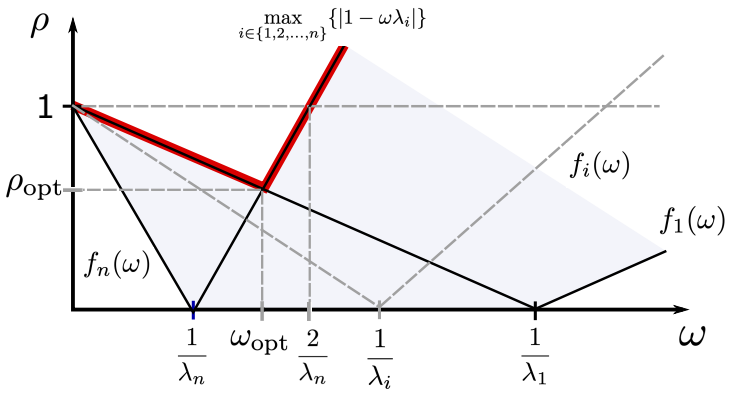

Figure 1.1: Konvergenční faktor Richardsonovy metody v závislosti na ω.

Věta Richardsonova metoda konverguje, právě když $\forall i \in \{1, 2, \dots, n\} : |1 - \omega\lambda_i| < 1$. Konvergenční faktor $\rho = \max_{i\in\{1,2,\dots,n\}}\{|1 - \omega\lambda_i|\}$ určuje rychlost konvergence: $\|\boldsymbol{r}^{k+1}\| \leq \rho\|\boldsymbol{r}^k\|$.

Čím menší bude konvergenční faktor $\rho = \max_{i\in\{1,2,\dots,n\}}\{|1-\omega\lambda_i|\}$, tím rychleji bude metoda konvergovat. Vzhledem k tomu, že vlastní čísla matice A jsou daná, můžeme konvergenční faktor ovlivnit pouze vhodnou volbou parametru ω. Odvození ideální hodnoty ω ilustrujme na Obrázku 1.1. Jsou na něm znázorněny funkce $f_1(\omega) = |1 - \omega\lambda_1|$ a $f_n(\omega) = |1 - \omega\lambda_n|$. Protože směrnice funkcí $f_i(\omega) = |1 - \omega\lambda_i|$ jsou určeny vlastními čísly matice A a ta jsou seřazena od nejmenšího po největší, budou grafy všech funkcí $f_i, i = 2, \dots, n-1$, ležet "mezi" grafy f_1 a f_n (v šedě vyznačené oblasti). Funkci

$$\max_{i\in\{1,2,\dots,n\}}\{|1 - \omega\lambda_i|\}$$

tedy můžeme vykreslit jako červeně zvýrazněnou lomenou čáru tvořenou částí funkce f_1 a částí funkce f_n. Z grafu této funkce tedy rovnou můžeme odvodit:

1. Interval, ve kterém musí ω ležet. Aby metoda konvergovala, musí platit $\rho = \max_{i\in\{1,2,\dots,n\}}\{|1 - \omega\lambda_i| < 1$. Červená funkce tedy musí ležet pod zakreslenou konstantní funkcí $\rho = 1$. Levý krajní bod intervalu je 0, pravý určíme jako průsečík příslušné části funkce f_n s konstantní funkcí 1:

   $$-(1 - \omega\lambda_n) = 1 \quad \Rightarrow \quad \omega = \frac{2}{\lambda_n}.$$

   Metoda tedy konverguje pro $\omega \in (0, 2/\lambda_n)$.

2. Optimální ω je bod, ve kterém červeně vyznačená funkce dosahuje minima. Tento bod dostaneme jako průsečík funkcí f_1 a f_n:

   $$1 - \omega_{\text{opt}}\lambda_1 = -(1 - \omega_{\text{opt}}\lambda_n) \quad \Rightarrow \quad \omega_{\text{opt}} = \frac{2}{\lambda_1 + \lambda_n}.$$

Zjistili jsme tedy, že nejlepší konvergence dosáhneme, zvolíme-li $\omega_{\text{opt}} = \frac{2}{\lambda_1+\lambda_n}$. Konvergenční faktor bude v tomto případě

$$\rho_{\text{opt}} = 1 - \omega_{\text{opt}}\lambda_1 = 1 - \frac{2\lambda_1}{\lambda_1+\lambda_n} = \frac{\lambda_1+\lambda_n-2\lambda_1}{\lambda_1+\lambda_n} = \\ = \frac{\lambda_n-\lambda_1}{\lambda_n+\lambda_1}\frac{\frac{1}{\lambda_1}}{\frac{1}{\lambda 1}} = \frac{\frac{\lambda_n}{\lambda_1}-1}{\frac{\lambda_n}{\lambda_1}+1} = \frac{\kappa(\mathsf{A})-1}{\kappa(\mathsf{A})+1},$$

kde $\kappa(\mathsf{A}) = \lambda_n/\lambda_1$ je číslo podmíněnosti matice A.

Můžeme také odvodit, kolik iterací je třeba, abychom dosáhli požadované relativní změny normy rezidua. Hledáme tedy k, pro které platí

$$\frac{\|\boldsymbol{r}^k\|}{\|\boldsymbol{r}^0\|} \le \varepsilon, \qquad \text{tedy} \qquad \|\boldsymbol{r}^k\| \le \varepsilon\|\boldsymbol{r}^0\|$$

Využijme toho, že $\|\boldsymbol{r}^k\| \le \rho_{\text{opt}}^k\|\boldsymbol{r}^0\|$ a přepišme nerovnici na

$$\rho_{\text{opt}}^k\|\boldsymbol{r}^0\| \le \varepsilon\|\boldsymbol{r}^0\|.$$

Vykrácením normy a zlogaritmováním obou stran nerovnice dostanem řešení $k \ge \frac{\log\varepsilon}{\log\rho_{\text{opt}}}$ (nezapomeňte, že protože $\rho_{\text{opt}} \in (0,1)$, je třeba otočit znaménko nerovnosti).

1.2 Gradientní iterační metody

Věta Řešení soustavy $\mathsf{A}\boldsymbol{x} = \boldsymbol{b}$ se symetrickou pozitivně definitní maticí A je ekvivalentní s minimalizací kvadratické formy

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T\mathsf{A}\boldsymbol{x} - \boldsymbol{b}^t\boldsymbol{x}.$$

Důkaz Dokažme nejdříve implikaci $\mathsf{A}\boldsymbol{x} = \boldsymbol{b} \Rightarrow \boldsymbol{x} = \arg\min_{\boldsymbol{v}\in\mathbb{R}^n} f(\boldsymbol{v})$. Podívejme se, jak se změní funkční hodnota f, posuneme-li se z bodu $\boldsymbol{x}$ o nějaký nenulový vektor $\boldsymbol{c}$:

$$f(\boldsymbol{p}) = f(\boldsymbol{x}+\boldsymbol{c}) = \frac{1}{2}(\boldsymbol{x}+\boldsymbol{c})^T\mathsf{A}(\boldsymbol{x}+\boldsymbol{c}) - \boldsymbol{b}(\boldsymbol{x}+\boldsymbol{c}) = \\ = \frac{1}{2}\boldsymbol{x}^T\mathsf{A}\boldsymbol{x} + \boldsymbol{c}^T\underbrace{\mathsf{A}\boldsymbol{x}}_{=\boldsymbol{b}} + \frac{1}{2}\boldsymbol{c}^T\mathsf{A}\boldsymbol{c} - \boldsymbol{b}^T\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{c} = \\ = \underbrace{\frac{1}{2}\boldsymbol{x}^T\mathsf{A}\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x}}_{=f(\boldsymbol{x})} + \underbrace{\boldsymbol{c}^T\boldsymbol{b} - \boldsymbol{b}^T\boldsymbol{c}}_{=0} + \frac{1}{2}\boldsymbol{c}^T\mathsf{A}\boldsymbol{c} = f(\boldsymbol{x}) + \underbrace{\frac{1}{2}\boldsymbol{c}^T\mathsf{A}\boldsymbol{c}}_{>0}.$$

Díky pozitivní definitnosti A je výraz $\boldsymbol{c}^T\mathsf{A}\boldsymbol{c}$ kladný. Posuneme-li se tedy z bodu $\boldsymbol{x}$ v libovolném směru, hodnota funkce f se zvětší. V bodě $\boldsymbol{x}$ tedy nastává minimum.

K důkazu opačné implikace $\boldsymbol{x} = \arg\min_{\boldsymbol{v}\in\mathbb{R}^n} f(\boldsymbol{v}) \Rightarrow \mathsf{A}\boldsymbol{x} = \boldsymbol{b}$ je třeba si uvědomit nutnou podmínku minima funkce $f : \mathbb{R}^n \to \mathbb{R}$, tedy nulovost gradientu:

$$\boldsymbol{x} = \arg\min_{\boldsymbol{v}\in\mathbb{R}^n} f(\boldsymbol{v}) \Rightarrow \nabla f(\boldsymbol{x}) = \boldsymbol{o}. \tag{1.6}$$

Lze ukázat, že pro gradient funkce f platí

$$\nabla f(\boldsymbol{x}) = \left[\frac{\partial f}{\partial x_1}(\boldsymbol{x}), \ldots, \frac{\partial f}{\partial x_n}(\boldsymbol{x})\right]^T = \frac{1}{2}\mathsf{A}^T\boldsymbol{x} + \frac{1}{2}\mathsf{A}\boldsymbol{x} - b = \mathsf{A}\boldsymbol{x} - \boldsymbol{b}.$$

Z podmínky (1.6) tedy vyplývá $\mathsf{A}\boldsymbol{x} - \boldsymbol{b} = \boldsymbol{o}$. □

2 LDMT, LDLT a Choleského dekompozice

2.1 LDMT (LDU) dekompozice

Uvažujme rozklad obecné čtvercové matice $\mathsf{A} \in \mathbb{R}^{n\times n}$

$$\mathsf{A} = \mathsf{L}\mathsf{D}\mathsf{M}^T,$$

kde $\mathsf{L} \in \mathbb{R}^{n\times n}$ a $\mathsf{M} \in \mathbb{R}^{n\times n}$ jsou dolní trojúhelníkové matice s jedničkami na diagonálách a $\mathsf{D} \in \mathbb{R}^{n\times n}$ je diagonální matice. Matici tedy rozkládáme do následujícího tvaru ($\times$ značí nenulové prvky matice):

$$\mathsf{A} = \mathsf{L}\mathsf{D}\mathsf{M}^T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \times & 1 & 0 & 0 & 0 \\ \times & \times & 1 & 0 & 0 \\ \times & \times & \times & 1 & 0 \\ \times & \times & \times & \times & 1 \end{bmatrix} \begin{bmatrix} \times & 0 & 0 & 0 & 0 \\ 0 & \times & 0 & 0 & 0 \\ 0 & 0 & \times & 0 & 0 \\ 0 & 0 & 0 & \times & 0 \\ 0 & 0 & 0 & 0 & \times \end{bmatrix} \begin{bmatrix} 1 & \times & \times & \times & \times \\ 0 & 1 & \times & \times & \times \\ 0 & 0 & 1 & \times & \times \\ 0 & 0 & 0 & 1 & \times \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Při odvozování vhodného algoritmu budeme vycházet z předpokladu, že již známe $j-1$ sloupců matice L, řádků matice M (tedy sloupců matice M^T) a diagonálních prvků $d_1, d_2, \ldots, d_{j-1}$ matice D, $1 \le j \le n$. Chceme odvodit vzorce pro j-tý sloupec matice L (tedy prvky $\mathsf{L}(j+1:n, j)$), j-tý řádek matice M (tedy prvky $\mathsf{M}(j, j+1:n)$ neboli $\mathsf{M}^T(j+1:n, j)$) a diagonální prvek d_j.

Sloupec j matice A můžeme vyjádřit pomocí našeho rozkladu jako

$$\mathsf{A}(1:n, j) = (\mathsf{L}\mathsf{D}\mathsf{M}^T)(1:n, j) = \mathsf{L}\mathsf{D}\mathsf{M}^T\boldsymbol{e}_j = \mathsf{L}(\mathsf{D}\mathsf{M}^T\boldsymbol{e}_j) = \mathsf{L}\boldsymbol{v},$$

kde $\boldsymbol{e}_j$ je j-tý vektor standardní báze prostoru $\mathbb{R}^n$ (tedy vektor, který má 1 na j-té pozici a jinde nuly). Součin $\mathsf{D}\mathsf{M}^T\boldsymbol{e}_j$ jsme si označili $\boldsymbol{v}$ a z tvarů matic D, M^T není těžké ověřit, že tento vektor má nenulové prvky pouze na prvních j pozicích. Uvědomme si, že $\boldsymbol{v}$ reprezentuje j-tý sloupec součinu matic $\mathsf{D}\mathsf{M}^T$ a vynásobíme-li jím matici L, získáme j-tý sloupec matice A.

Rozdělme nyní sloupec $\mathsf{A}(1:n, j)$ na dvě části - $\mathsf{A}(1:j, j)$ a $\mathsf{A}(j+1:n, j)$. Pro první část platí

$$\mathsf{A}(1:j,j) = \mathsf{L}(1:j,1:j)\boldsymbol{v}(1:j).$$

Výraz na levé straně známe, submatici $\mathsf{L}(1:j,1:j)$ známe také (z předpokladu známe $j-1$ sloupců L a víme, že $\mathsf{L}(j,j)=1$). Neznámý vektor $\boldsymbol{v}(1:j)$ tedy získáme řešením této soustavy j rovnic o j neznámých s dolní trojúhelníkovou maticí (k řešení můžeme použít efektivní dopřednou substituci).

Po nalezení vektoru $\boldsymbol{v}(1:j)$ určeme příslušné prvky matice M. Vektor $\boldsymbol{v}$ je j-tým sloupcem součinu DM^T. Z tvarů matic D, M^T tedy vyplývá, že pro prvky $\boldsymbol{v}(1:j)$ platí

$$\boldsymbol{v}(1:j) = \mathsf{D}(1:j,1:j)\mathsf{M}^T(1:j,j)$$
$$\mathsf{D}(1:j,1:j)^{-1}\boldsymbol{v}(1:j) = \mathsf{M}^T(1:j,j)$$

Protože matice D je diagonální, k jejímu invertování stačí pouze invertovat diagonální prvky. Po přenásobení inverzní diagonální matice vektorem $\boldsymbol{v}(1:j)$ dostaneme pro $i = 1,\ldots,j-1$ rovnost

$$\frac{1}{d_i}\boldsymbol{v}(i) = \mathsf{M}^T(i,j),$$

tedy $\mathsf{M}(j,i) = \frac{1}{d_i}\boldsymbol{v}(i)$. Zbývá určit neznámý prvek d_j. Protože $\mathsf{M}(j,j)=1$, musí platit

$$1 = \frac{1}{d_j}\boldsymbol{v}(j),$$

tedy $d_j = \boldsymbol{v}(j)$.

Určili jsme tedy j-tý řádek matice M a prvek d_j, zbývá určit j-tý sloupec matice L. K tomu využijeme druhou část sloupce $\mathsf{A}(1:n,j)$. Tu si můžeme vyjádřit jako

$$\mathsf{A}(j+1:n,j) = \mathsf{L}(j+1:n,1:j)\boldsymbol{v}(1:j)$$

Výraz na levé straně opět známe, známe již také vektor $\boldsymbol{v}(1:j)$ a všechny sloupce submatice $\mathsf{L}(j+1:n,1:j)$ kromě posledního, tedy kromě $\mathsf{L}(j+1:n,j)$. Součin na pravé straně si můžeme rozepsat na

$$\mathsf{A}(j+1:n,j) = \mathsf{L}(j+1:n,1:j-1)\boldsymbol{v}(1:j-1) + \mathsf{L}(j+1:n,j)\boldsymbol{v}(j).$$

Osamostatněním neznámého sloupce získáme

$$\mathsf{L}(j+1:n,j) = \frac{\mathsf{A}(j+1:n,j) - \mathsf{L}(j+1:n,1:j-1)\boldsymbol{v}(1:j-1)}{\boldsymbol{v}(j)}.$$

Výsledný algoritmus tedy můžeme zapsat takto:

function LDMT(A)
  $n = \text{size}(\mathsf{A})$
  $\mathsf{L} = \text{eye}(n,n)$, $\mathsf{M} = \text{eye}(n,n)$, $\mathsf{D} = \text{zeros}(n,n)$
  $\boldsymbol{v}(1) = \mathsf{A}(1,1)$
  $\mathsf{D}(1,1) = \boldsymbol{v}(1)$
  $\mathsf{L}(2:n,1) = A(2:n,1)/\boldsymbol{v}(1)$

for $j = 2, \ldots, n$ **do**
Solve $\mathsf{L}(1:j, 1:j)\boldsymbol{v}(1:j) = A(1:j, j)$ for $\boldsymbol{v}(1:j)$
for $i = 1, \ldots, j-1$ **do**
$\mathsf{M}(j, i) = \boldsymbol{v}(i)/\mathsf{D}(i, i)$
end for
$\mathsf{D}(j, j) = \boldsymbol{v}(j)$
$\mathsf{L}(j+1:n, j) = \frac{\mathsf{A}(j+1:n,j) - \mathsf{L}(j+1:n,1:j-1)\boldsymbol{v}(1:j-1)}{\boldsymbol{v}(j)}$
end for
end function

2.2 LDLT dekompozice

Dá se ukázat, že je-li matice A na vstupu algoritmu pro LDMT dekompozici symetrická, platí $\mathsf{L} = \mathsf{M}$. Matici jsme tedy schopni rozložit na součin $\mathsf{A} = \mathsf{LDL}^T$. Vhodnou úpravou předchozího algoritmu jsme schopni snížit jeho náročnost pro symetrické matice na polovinu.

Pro odvození tedy opět vyjděme z předpokladu, že známe $j-1$ sloupců matice L, řádků M a diagonálních prvků matice D. Navíc, protože $\mathsf{M} = \mathsf{L}$, známe i prvky v příslušných *sloupcích* matice M, včetně prvků $\mathsf{M}(j, 1:j-1)$, tedy $\mathsf{M}^T(1:j-1, j)$. Vzpomeňme si, co platilo pro vektor $\boldsymbol{v}(1:j)$:

$$\boldsymbol{v}(1:j) = \mathsf{D}(1:j, 1:j)\mathsf{M}^T(1:j, j) = \mathsf{D}(1:j, 1:j)\mathsf{L}^T(1:j, j)$$

Díky tomu, že D je diagonální matice, můžeme si součin na pravé straně snadno rozepsat na

$$\boldsymbol{v}(1:j) = \begin{bmatrix} \mathsf{D}(1,1)\mathsf{L}^T(1,j) \\ \mathsf{D}(2,2)\mathsf{L}^T(2,j) \\ \vdots \\ \mathsf{D}(j-1,j-1)\mathsf{L}^T(j-1,j) \\ \mathsf{D}(j,j) \end{bmatrix}$$

Díky předpokladům známe všechny prvky vektoru napravo kromě posledního. Získali jsme tedy předpis, který nám říká, jak snadno získat $\boldsymbol{v}(1:j-1)$. Na posledním řádku jsme navíc využili toho, že $\mathsf{L}(j, j) = 1$. Zbývá nalézt $\boldsymbol{v}(j)$. K tomu si vyjádřeme, čemu se rovná $\mathsf{A}(j, j)$:

$$\mathsf{A}(j, j) = \mathsf{L}(j, 1:j)\boldsymbol{v}(1:j)$$

Rozepišme součin napravo opět na známou a neznámou část

$$\mathsf{A}(j, j) = \mathsf{L}(j, 1:j-1)\boldsymbol{v}(1:j-1) + \mathsf{L}(j, j)\boldsymbol{v}(j) = \mathsf{L}(j, 1:j-1)\boldsymbol{v}(1:j-1) + 1\boldsymbol{v}(j)$$

a vyjádřeme neznámý prvek $\boldsymbol{v}(j)$

$$\boldsymbol{v}(j) = \mathsf{A}(j, j) - \mathsf{L}(j, 1:j-1)\boldsymbol{v}(1:j-1).$$

Tímto jsme získali $\boldsymbol{v}(1:j)$ aniž bychom museli řešit soustavu rovnic, čímž jsme původní algoritmus výrazně urychlili. Zbývající část odvození je identická. Výsledný algoritmus pro LDLT rozklad symetrické matice tedy můžeme zapsat takto:

function LDLT(A)
  $n = \text{size}(\mathsf{A})$
  $\mathsf{L} = \text{eye}(n, n)$, $\mathsf{D} = \text{zeros}(n, n)$
  $\boldsymbol{v}(1) = \mathsf{A}(1, 1)$
  $\mathsf{D}(1, 1) = \boldsymbol{v}(1)$
  $\mathsf{L}(2 : n, 1) = A(2 : n, 1)/\boldsymbol{v}(1)$
  for $j = 2, \ldots, n$ **do**
    for $i = 1, \ldots, j - 1$ **do**
      $\boldsymbol{v}(i) = \mathsf{L}(j, i)\mathsf{D}(i, i)$
    end for
    $\boldsymbol{v}(j) = \mathsf{A}(j, j) - \mathsf{L}(j, 1 : j - 1)\boldsymbol{v}(1 : j - 1)$
    $\mathsf{D}(j, j) = \boldsymbol{v}(j)$
    $\mathsf{L}(j + 1 : n, j) = \frac{\mathsf{A}(j+1:n,j)-\mathsf{L}(j+1:n,1:j-1)\boldsymbol{v}(1:j-1)}{\boldsymbol{v}(j)}$
  end for
end function

2.3 Choleského rozklad

Matici A, pro kterou platí $a_{ij} = a_{ji}$ a

$$\forall \boldsymbol{x} \neq 0 : \boldsymbol{x}^T \mathsf{A} \boldsymbol{x} > 0$$

nazýváme symetrickou pozitivně definitní. Tyto matice často vznikají diskretizací fyzikálních systémů. Zopakujme si některé jejich vlastnosti:

1. Každá pozitivně definitní matice má kladné prvky na diagonále ($a_{ii} = \boldsymbol{e}_i^T \mathsf{A} \boldsymbol{e}_i > 0$, nerovnost vyplývá z definice pozitivní definitnosti)

2. Matice je pozitivně definitní, je-li symetrická a má-li všechna vlastní čísla kladná.

3. Každá pozitivně definitní matice může být rozložena na

$$\mathsf{A} = \mathsf{R}^T \mathsf{R},$$

   kde R je horní trojúhelníková matice s kladnými prvky na diagonále.

Dekompozici z bodu 3 nazýváme Choleského rozklad. Pokusme se odvodit jeho základní tvar. Rozdělme matice A a R na čtyři bloky:

$$\mathsf{A} = \begin{bmatrix} \mathsf{A}(1,1) & \mathsf{A}(1,2:n) \\ \mathsf{A}(2:n,1) & \mathsf{A}(2:n,2:n) \end{bmatrix}, \mathsf{R} = \begin{bmatrix} \mathsf{R}(1,1) & \mathsf{R}(1,2:n) \\ 0 & \mathsf{R}(2:n,2:n) \end{bmatrix}$$

Hledaný součin má tvar

$$\begin{bmatrix} \mathsf{A}(1,1) & \mathsf{A}(1,2:n) \\ \mathsf{A}(2:n,1) & \mathsf{A}(2:n,2:n) \end{bmatrix} = \begin{bmatrix} \mathsf{R}(1,1) & 0 \\ \mathsf{R}^T(1,2:n) & \mathsf{R}^T(2:n,2:n) \end{bmatrix} \begin{bmatrix} \mathsf{R}(1,1) & \mathsf{R}(1,2:n) \\ 0 & \mathsf{R}(2:n,2:n) \end{bmatrix} =$$

$$= \begin{bmatrix} \mathsf{R}^2(1,1) & \mathsf{R}(1,1)\mathsf{R}(1,2:n) \\ \mathsf{R}(1,1)\mathsf{R}^T(1,2:n) & \mathsf{R}^T(1,2:n)\mathsf{R}(1,2:n) + \mathsf{R}^T(2:n,2:n)\mathsf{R}(2:n,2:n) \end{bmatrix}$$

Porovnáním bloků na odpovídajících pozicích na levé a pravé straně nejdříve snadno vypočítáme první řádek matice R. Musí platit $\mathsf{R}^2(1,1) = \mathsf{A}(1,1)$, takže $\mathsf{R}(1,1) = \sqrt{\mathsf{A}(1,1)}$ (uvědomme si, že díky pozitivní definitnosti A víme, že diagonální prvek je kladný). Odsud a z $\mathsf{A}(1,2:n) = \mathsf{R}(1,1)\mathsf{R}(1,2:n)$ snadno vyjádříme $\mathsf{R}(1,2:n) = \mathsf{A}(1,2:n)/\mathsf{R}(1,1)$. Zbývá určit blok $\mathsf{R}(2:n,2:n)$. Porovnáním bloků na pozicích (2,2) získáme rovnost

$$\mathsf{A}(2:n,2:n) - \mathsf{R}^T(1,2:n)\mathsf{R}(1,2:n) = \mathsf{R}^T(2:n,2:n)\mathsf{R}(2:n,2:n).$$

Dá se ukázat, že výraz na levé straně je opět symetrická pozitivně definitní matice. Výraz na pravé straně je její Choleského rozklad. K nalezení matice a $\mathsf{R}(2:n,2:n)$ tedy rekurzivně aplikujeme tento postup na levou stranu předchozí rovnosti. V dalším kroce tedy nalezneme řádek $\mathsf{R}(2,2:n)$ a rekurzivně budeme pokračovat, dokud nenalezneme zbývající řádky matice R. Promyslete si, že se algoritmus dá zapsat takto:

function CHOLESKY(A)
  $m = \text{size}(\mathsf{A})$
  $\mathsf{R} = \mathsf{A}$
  for $k = 1, \ldots, m$ **do**
    for $j = k+1, \ldots, m$ **do**
      $\mathsf{R}(j, j:m) = \mathsf{R}(j, j:m) - \mathsf{R}(k, j:m)\mathsf{R}(k,j)/\mathsf{R}(k,k)$
    end for
    $\mathsf{R}(k, k:m) = \frac{\mathsf{R}(k,k:m)}{\sqrt{(\mathsf{R}(k,k))}}$
  end for
end function

3 Výpočetní náročnost

3.1 Gaussova eliminace (LU rozklad)

Zopakujme si algoritmus LU faktorizace:

function LU(A)
  $m = \text{size}(\mathsf{A})$
  $\mathsf{U} = \mathsf{A}$
  for $k = 1, \ldots, m-1$ **do**
    for $j = k+1, \ldots, m$ **do**
      $\mathsf{L}(j,k) = \mathsf{U}(j,k)/\mathsf{U}(k,k)$
      $\mathsf{U}(j, k:m) = \mathsf{U}(j, k:m) - \mathsf{L}(j,k)\mathsf{U}(k, k:m)$
    end for
  end for
end function

Výpočtu dominuje vnitřní smyčka, konkrétně řádek

$$\mathsf{U}(j, k:m) = \mathsf{U}(j, k:m) - \mathsf{L}(j,k)\mathsf{U}(k, k:m) \tag{3.1}$$

Vektor $\mathsf{U}(k, k : m)$ má v k-tém kroce délku $l = m - k + 1$. Na řádku tento vektor přenásobíme skalárem (l násobení) a odečteme od vektoru $\mathsf{U}(j, k : m)$ (l rozdílů). Celkem tedy na tomto řádku provedeme $2l$ operací. Zajímá nás, kolikrát se (3.1) provede v průběhu celého běhu algoritmu a s jak dlouhými vektory při tom pracuje. Rozepišme si postupně počet volání (3.1) a délku l pro jednotlivé kroky k:

$$k = 1 : \text{počet: } m - 1, \quad l = m$$

$$k = 2 : \text{počet: } m - 2, \quad l = m - 1$$

$$\vdots$$

$$k = m - 1 : \text{počet: } 1, \quad l = m - (m - 1) + 1 = 2.$$

Celkový počet operací q je tedy dán výrazem

$$q = 2((m-1)m + (m-2)(m-1) + \ldots + 2 \cdot 3 + 1 \cdot 2),$$

tedy

$$q = \sum_{i=1}^{m} 2(m-i)(m-i+1) = 2\sum_{j=0}^{m-1} j(j+1) = 2\sum_{j=0}^{m-1}(j^2+j) = 2(\sum_{j=0}^{m-1} j^2 + \sum_{j=0}^{m-1} j).$$

K vyčíslení těchto sum můžeme použít vzorce $\sum_{j=0}^{m} j^2 = \frac{1}{6}(2m+1)(m+1)m$ a $\sum_{j=0}^{m} j = \frac{m(m+1)}{2}$. Dostaneme tedy ($m^2$ a m odčítáme, protože naše sumy sčítají pouze po $j = m - 1$):

$$q = 2(\frac{1}{6}(2m+1)(m+1)m - m^2 + \frac{m(m+1)}{2} - m) = 2(\frac{1}{3}m^3 - \frac{1}{3}m)$$

Výpočtu dominuje třetí mocnina, proto můžeme říct, že náročnost Gaussovy eliminace (LU rozkladu), je $\approx \frac{2}{3}m^3$.

3.2 LU rozklad s částečnou a úplnou pivotizací

Určeme nyní náročnost částečné pivotizace při výpočtu LU rozkladu. V každém kroce $k = 1, \ldots, k = m - 1$ hledáme největší pivot ve sloupci délky $m - k + 1$. Celkový počet porovnání je tedy

$$m + (m-1) + (m-2) + \ldots + 2 = \frac{(m+2)(m+1)}{2} = \frac{1}{2}(m^2 + m - 2)$$

Výpočtu dominuje druhá mocnina, můžeme tedy říct, že celkový počet porovnání je $\approx \frac{1}{2}m^2$, tedy náročnost částečné pivotizace je $O(m^2)$. Vzhledem k tomu, že náročnost samotného LU rozkladu je kubická, nepřináší částečná pivotizace významný overhead.

Podobně bychom mohli odvodit, že náročnost úplné pivotizace je $O(m^3)$, což výrazně přispěje k původní náročnosti metody. Z tohoto důvodu se úplná pivotizace často nepoužívá.

3.3 Další algoritmy

Stejným způsobem lze odvodit, že náročnost LDMT rozkladu je $2/3m^3$, LDLT a Choleského rozklad symetrické matice vyžadují poloviční počet operací, tedy $1/3m^3$. Řešení soustavy s dolní nebo horní trojúhelníkovou maticí pomocí dopředné nebo zpětné substituce vyžaduje $1/2m^2$ operací. Pro srovnání výpočet inverzní matice vyžaduje m^3 operací a řešení systému pomocí inverzní matice (násobení inverzní maticí) vyžaduje m^2 operací.

4 Iterační metody

Jak jsme si ukázali v předchozí kapitole, přímé řešiče vyžadují $O(m^3)$ operací, což je pro praktické použití s velkými maticemi příliš mnoho. Proto velikost soustav s plnými maticemi, které jsme schopni řešit pomocí přímých řešičů, nerostla v historii příliš rychle:

- 1950: $m = 20$ (Wilkinson)
- 1965: $m = 200$ (Forsythe, Moler)
- 1980: $m = 2000$ (LINPACK)
- 1995: $m = 20000$ (LAPACK)
- 2010: $m = 200000$ (HDSS)

V případě řídkých systémů jsme sice schopni řešit soustavy s ještě větším počtem neznámých, nicméně použití přímých řešičů může vést k zaplnění systémů. Uvědomme si např., co se stane pokud eliminujeme první sloupec následující matice pomocí prvního řádku:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & 0 & 0 & 0 \\ \times & 0 & \times & 0 & 0 \\ \times & 0 & 0 & \times & 0 \\ \times & 0 & 0 & 0 & \times \end{bmatrix} \Rightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}$$

Alternativou je použít iterační řešiče, ty generují posloupnost přibližných řečení $\{x^{(k)}\}$ a pracují téměř výhradně s násobením matice a vektoru (což vyžaduje $O(m^2)$ operací). Podstatnou vlastností každého iteračního řešiče je tedy rychlost konvergence $\{x^{(k)}\}$ ke správnému řešení. Pokud bychom totiž k řešení dokonvergovali až po m a více iteracích, potřebovali bychom opět minimálně m^3 operací.

4.1 Lineární iterační řešiče

Hledejme posloupnost přibližných řešení rovnice $\mathsf{A}\boldsymbol{x} = \boldsymbol{b}$, kde $\mathsf{A} \in \mathbb{R}^{n\times n}$ je regulární matice, ve tvaru

$$\boldsymbol{x}^{(k+1)} := \mathsf{M}\boldsymbol{x}^{(k)} + \mathsf{N}\boldsymbol{b}$$

(někdy se v literatuře setkáte také s tvarem $\mathsf{R}\boldsymbol{x}^{(k+1)} := \mathsf{W}\boldsymbol{x}^{(k)} + \boldsymbol{b}$). Horní index (k) zde znamená k-tou iteraci.

Řekneme, že metoda je *konzistentní*, pokud rovnici $\boldsymbol{x} := \mathsf{M}\boldsymbol{x} + \mathsf{N}\boldsymbol{b}$ řeší právě jeden vektor $\boldsymbol{x} = \mathsf{A}^{-1}\boldsymbol{b}$. Lze ukázat, že toto platí právě tehdy, když $\mathsf{M} = \mathsf{I} - \mathsf{NA}$.

Řekneme, že metoda je *konvergentní*, platí-li $\boldsymbol{x}^{(k)} \to \boldsymbol{x}$ pro $k \to \infty$. Toto platí právě tehdy, když $\|\mathsf{M}\| < 1$.

Rozložme nyní matici A na součet $\mathsf{A} = \mathsf{L} + \mathsf{D} + \mathsf{U}$. Např. pro matici

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

by rozklad vypadal takto

$$\mathsf{L} = \begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix}, \mathsf{D} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} \mathsf{U} = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

4.1.1 Jacobiho metoda

V případě Jacobiho metody vyjdeme ze zápisu soustavy $\mathsf{A}\boldsymbol{x} = \boldsymbol{b}$ ve tvaru

$$(\mathsf{L} + \mathsf{D} + \mathsf{U})\boldsymbol{x} = \boldsymbol{b}.$$

Roznásobme závorku do následujícího tvaru

$$\mathsf{D}\boldsymbol{x} + (\mathsf{L} + \mathsf{U})\boldsymbol{x} = \boldsymbol{b}$$

a převeďme člen s $(\mathsf{L} + \mathsf{U})$ na pravou stranu

$$\mathsf{D}\boldsymbol{x} = -(\mathsf{L} + \mathsf{U})\boldsymbol{x} + \boldsymbol{b}.$$

Jacobiho iterace pak definujeme takto

$$\mathsf{D}\boldsymbol{x}^{(k+1)} := -(\mathsf{L} + \mathsf{U})\boldsymbol{x}^{(k)} + \boldsymbol{b}.$$

Přenásobením inverzí k D získáme konečný tvar

$$\boldsymbol{x}^{(k+1)} := -\mathsf{D}^{-1}(\mathsf{L} + \mathsf{U})\boldsymbol{x}^{(k)} + \mathsf{D}^{-1}\boldsymbol{b}.$$

Zde $\mathsf{M} = -\mathsf{D}^{-1}(\mathsf{L}+\mathsf{U})$ a $\mathsf{N} = \mathsf{D}^{-1}$. Vzhledem k tomu, že D je diagonální matice a že $(\mathsf{L}+\mathsf{U})\boldsymbol{x}^{(k)}$ je násobením vektoru $\boldsymbol{x}^{(k)}$ původní maticí A bez její diagonály, můžeme napsat předpis pro i-tý prvek vektoru $\boldsymbol{x}^{(k+1)}$ takto:

$$\begin{aligned} (\boldsymbol{x}^{(k+1)})_i &:= \frac{1}{a_{ii}}(b_i - \sum_{j=1, j\neq i}^{n} a_{ij}(\boldsymbol{x}^{(k)})_j) = \\ &= \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}(\boldsymbol{x}^{(k)})_j - \sum_{j=i+1}^{n} a_{ij}(\boldsymbol{x}^{(k)})_j) \end{aligned} \tag{4.1}$$

Konzistence metody vyplývá ze způsobu, jakým jsme ji odvodili, tedy

$$\mathsf{A}\boldsymbol{x} = (\mathsf{L} + \mathsf{D} + \mathsf{U})\boldsymbol{x} = \boldsymbol{b},$$

tedy

$$\boldsymbol{x} = -\mathsf{D}^{-1}(\mathsf{L} + \mathsf{U})\boldsymbol{x} + \mathsf{D}^{-1}\boldsymbol{b}. \tag{4.2}$$

Řeší-li $\boldsymbol{x} = \mathsf{A}^{-1}\boldsymbol{b}$ původní rovnici $\mathsf{A}\boldsymbol{x} = \boldsymbol{b}$, řeší také (4.2). Metoda je tedy konzistentní. Konzistenci můžeme také ověřit zjistěním, zda platí rovnost $\mathsf{M} = \mathsf{I} - \mathsf{NA}$:

$$\begin{aligned}\mathsf{I} - \mathsf{NA} &= \mathsf{I} - \mathsf{D}^{-1}\mathsf{A} = \mathsf{I} - \mathsf{D}^{-1}(\mathsf{L} + \mathsf{D} + \mathsf{U}) \\ &= \mathsf{I} - \mathsf{D}^{-1}\mathsf{D} - \mathsf{D}^{-1}(\mathsf{L} + \mathsf{U}) = -\mathsf{D}^{-1}(\mathsf{L} + \mathsf{U}) = \mathsf{M}.\end{aligned}$$

Jacobiho metoda je konvergentní, platí-li $\|\mathsf{M}\| = \|\mathsf{D}^{-1}(\mathsf{L} + \mathsf{U})\| < 1$, což splňují např. diagonálně dominantní matice, tedy takové matice, pro které platí: $\forall i : |a_{ii}| > \sum_{j \neq i} a_{ij}$.

4.1.2 Gaussova-Seidelova metoda

Všimneme si, že počítáme-li Jacobiho metodou i-tý prvek vektoru vektoru $\boldsymbol{x}^{(k+1)}$, tedy $(\boldsymbol{x}^{(k+1)})_i$, známe již všech $i-1$ předchozích prvků tohoto vektoru. V předpise (4.1) můžeme tedy člen $\sum_{j=1}^{i-1} a_{ij}(\boldsymbol{x}^{(k)})_j$ nahradit členem $\sum_{j=1}^{i-1} a_{ij}(\boldsymbol{x}^{(k+1)})_j$. Touto změnou dostaneme předpis pro Gaussovu-Seidelovu metodu:

$$(\boldsymbol{x}^{(k+1)})_i := \frac{1}{a_{ii}}(b_i - \sum_{j=1}^{i-1} a_{ij}(\boldsymbol{x}^{(k+1)})_j - \sum_{j=i+1}^{n} a_{ij}(\boldsymbol{x}^{(k)})_j), \tag{4.3}$$

což lze zapsat maticově jako

$$\boldsymbol{x}^{(k+1)} := \mathsf{D}^{-1}(\boldsymbol{b} - \mathsf{L}\boldsymbol{x}^{(k+1)} - \mathsf{U}\boldsymbol{x}^{(k)}).$$

Převedením $\boldsymbol{x}^{(k+1)}$ na levou stranu dostaneme

$$(\mathsf{D} + \mathsf{L})\boldsymbol{x}^{(k+1)} := \boldsymbol{b} - \mathsf{U}\boldsymbol{x}^{(k)},$$

odtud

$$\boldsymbol{x}^{(k+1)} := -(\mathsf{D} + \mathsf{L})^{-1}\mathsf{U}\boldsymbol{x}^{(k)} + (\mathsf{D} + \mathsf{L})^{-1}\boldsymbol{b}.$$

Pro Gaussovu-Seidelovu metodu tedy platí $\mathsf{M} = -(\mathsf{D} + \mathsf{L})^{-1}\mathsf{U}, \mathsf{N} = (\mathsf{D} + \mathsf{L})^{-1}$. Ověřme její konzistenci:

$$\begin{aligned}\mathsf{I} - \mathsf{NA} &= \mathsf{I} - (\mathsf{D} + \mathsf{L})^{-1}\mathsf{A} = \mathsf{I} - (\mathsf{D} + \mathsf{L})^{-1}(\mathsf{L} + \mathsf{D} + \mathsf{U}) \\ &= \mathsf{I} - (\mathsf{D} + \mathsf{L})^{-1}(\mathsf{D} + \mathsf{L}) - (\mathsf{D} + \mathsf{L})^{-1}\mathsf{U} = -(\mathsf{D} + \mathsf{L})^{-1}\mathsf{U} = \mathsf{M}.\end{aligned}$$

Metoda je tedy konzistentní. Gaussova-Seidelova metoda je konvergentní, právě když $\|\mathsf{M}\| = \| - (\mathsf{D} + \mathsf{L})^{-1}\mathsf{U}\| < 1$.

4.1.3 Richardsonova metoda

Předpis pro $k+1$. krok Richardsonovy metody je dán

$$\boldsymbol{x}^{k+1} := \boldsymbol{x}^{(k)} + \omega \boldsymbol{r}^{(k)},$$

kde $\boldsymbol{r}^{(k)} = \boldsymbol{b} - \mathsf{A}\boldsymbol{x}^{(k)}$ je tzv. reziduum a $\omega > 0$.

Metoda je konzistentní pro libovolné ω ($\boldsymbol{x} = \boldsymbol{x} + \omega(\boldsymbol{b} - \mathsf{A}\boldsymbol{x})$), $\boldsymbol{x} = \boldsymbol{x} + \omega\boldsymbol{0}$).

Studujme nyní konvergenci metody pro A symetrickou pozitivně definitní matici. V takovém případě vlastní čísla λ_i a vlastní vektory a $\boldsymbol{v}_i$ (tedy λ_i a $\boldsymbol{v}_i$, pro které platí $\mathsf{A}\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i$) splňují

$$0 < \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \ldots \leq \lambda_n$$

a $\{\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \ldots, \boldsymbol{v}_n\}$ tvoří ortonormální bázi prostoru $\mathbb{R}^n$. Každý prvek toho prostoru tedy můžeme vyjádřit jako lineární kombinaci vektorů $\boldsymbol{v}_i$.

Studujme nyní, jak se šíří reziduum $\boldsymbol{r}^{(k+1)}$. Nejdříve si vyjádřeme $\boldsymbol{r}^{(k+1)}$ a $\boldsymbol{r}^{(k)}$ v bázi dané vektory $\boldsymbol{v}_i$.

$$\boldsymbol{r}^{(k+1)} = \sum_{i=1}^{n} \alpha_i^{(k+1)} \boldsymbol{v}_i, \quad \boldsymbol{r}^{(k)} = \sum_{i=1}^{n} \alpha_i^{(k)} \boldsymbol{v}_i$$

a dále postupujme takto:

$$\begin{aligned}
\boldsymbol{r}^{(k+1)} &= \boldsymbol{b} - \mathsf{A}\boldsymbol{x}^{(k+1)} = \boldsymbol{b} - \mathsf{A}(\boldsymbol{x}^{(k)} + \omega \boldsymbol{r}^{(k)}) = (\boldsymbol{b} - \mathsf{A}\boldsymbol{x}^{(k)}) - \omega \mathsf{A}\boldsymbol{r}^{(k)}) \\
&= \boldsymbol{r}^{(k)} - \omega \mathsf{A}\boldsymbol{r}^{(k)} = (\mathsf{I} - \omega\mathsf{A})\boldsymbol{r}^{(k)} = (\mathsf{I} - \omega\mathsf{A}) \sum_{i=1}^{n} \alpha_i^{(k)} \boldsymbol{v}_i \\
&= \sum_{i=1}^{n} \alpha_i^{(k)} \boldsymbol{v}_i - \sum_{i=1}^{n} \omega\lambda_i \alpha_i^{(k)} \boldsymbol{v}_i = \sum_{i=1}^{n} (\alpha_i^{(k)} \boldsymbol{v}_i - \omega\lambda_i \alpha_i^{(k)} \boldsymbol{v}_i) \\
&= \sum_{i=1}^{n} (1 - \omega\lambda_i) \alpha_i^{(k)} \boldsymbol{v}_i.
\end{aligned}$$

Všimneme si tedy, že jsme dostali rovnost

$$\boldsymbol{r}^{(k+1)} = \sum_{i=1}^{n} \alpha_i^{(k+1)} \boldsymbol{v}_i = \sum_{i=1}^{n} (1 - \omega\lambda_i) \alpha_i^{(k)} \boldsymbol{v}_i.$$

Vyjádřili jsme tedy koeficienty $\alpha_i^{(k+1)}$ pomocí koeficientů v předchozím kroce, tedy $\alpha_i^{(k)}$, přenásobených konstantami $(1-\omega\lambda_i)$. Konvergence rezidua k nulovému vektoru tedy bude zajištěna, právě když $\forall i : |1 - \omega\lambda_i| < 1$. Platí $\|\boldsymbol{r}^{(k+1)}\| \leq \varrho \|\boldsymbol{r}^{(k+1)}\|$, kde ϱ je konvergenční faktor $\varrho = \max_{i \in \{1,\ldots,n\}} |1 - \omega\lambda_i|$.

Jednoduchou úvahou můžeme odvodit vhodné hodnoty parametru ω.