

UNIVERZITA PALACKÉHO V OLMOUCI
PŘÍRODOVĚDECKÁ FAKULTA
KATEDRA MATEMATICKÉ ANALÝZY A APLIKACÍ MATEMATIKY

BAKALÁŘSKÁ PRÁCE

Iterační metody pro řešení systémů lineárních
rovníc



Vedoucí bakalářské práce:
RNDr. Jitka Machalová, Ph.D.
Rok odevzdání: 2011

Vypracovala:
Jana Prokopová
ME, III. ročník

Prohlášení

Prohlašuji, že jsem vytvořila tuto bakalářskou práci samostatně za vedení RNDr. Jitky Machalové, Ph.D. a že jsem v seznamu použité literatury uvedla všechny zdroje použité při zpracování práce.

V Olomouci dne 6.12. 2011

Poděkování

Ráda bych na tomto místě poděkovala vedoucí bakalářské práce RNDr. Jitce Machalové, Ph.D. za trpělivou a obětavou spolupráci, odborné vedení, cenné rady a čas, který mi věnovala při konzultacích. Dále si zaslouží poděkování moje rodina za podporu během studia.

Obsah

Použité značení	4
Úvod	5
1 Základní pojmy	7
2 Iterační metody	14
2.1 Jacobiova metoda	18
2.2 Gaussova–Seidelova iterační metoda	30
2.3 Relaxační metoda	42
3 Analýza metod	57
Závěr	61

Použité značení

\mathbb{N}	množina přirozených čísel
\mathbb{R}	množina reálných čísel
\mathbb{R}^n	vektorový prostor uspořádaných n -tic reálných čísel
\mathbb{R}^*	rozšířená množina reálných čísel
\mathbb{M}_n	třída všech čtvercových matic řádu n s reálnými prvky
\mathbf{O}	nulová matice
\mathbf{o}	nulový sloupcový vektor
$\rho(\mathbf{T})$	spektrální poloměr matice \mathbf{T}
\mathbf{I}	jednotková matice
λ	vlastní čísla matice
$ x $	absolutní hodnota čísla x
$ \mathbf{A} $	determinant matice \mathbf{A}
\mathbf{A}^T	transponovaná matice k matici \mathbf{A}
\mathbf{A}^{-1}	inverzní matice k matici \mathbf{A}

Úvod

V průběhu studia jsem se seznámila s teorií matic pouze okrajově, a protože mne zaujala, rozhodla jsem se ji lépe pochopit v rámci své bakalářské práce. Po bližším zkoumání jsem zjistila, že tato teorie je velice obsáhlá. Tudíž jsem se musela zaměřit pouze na některou její část. V lineární algebře jsme se zabývali většinou pouze lineárními rovnicemi s menším počtem neznámých a jejich aplikací na maticové řešení. Právě v tuto chvíli mne napadlo: „Jak bychom řešili velký počet lineárních rovnic s mnoha neznámými?“. Snažila jsem se nalézt odpověď na tuto otázku a dospěla jsem k tomu, že nejvhodnější bude použít Iterační metody. Protože já jsem se s nimi během studia nesetkala a nebylo pro mne jednoduché je nastudovat, je tato práce věnována těm, kteří mají jen základní povědomí o lineární algebře, kteří o iteračních metodách nic neví a rádi by se s nimi touto formou seznámili.

Iterační metody se používají k nalezení řešení soustav n lineárních rovnic o n neznámých, kde n je velké. Při řešení soustav lineárních rovnic pomocí Iteračních metod nedostáváme přesné řešení soustavy. Počítáme posloupnost vektorů $\{\mathbf{x}^k\}$, kde $k \in \mathbb{N}$, která konverguje, při splnění určitých předpokladů, k přesnému řešení \mathbf{x}^* . Můžeme provést pouze konečný počet kroků dané iterační metody, a proto se tedy můžeme většinou jen přiblížit k přesnému řešení \mathbf{x}^* a to s určitou přesností. Tu budeme označovat ϵ . Iteračními metodami se snažíme tedy pouze nalézt dostatečně přesnou aproximaci přesného řešení.

První kapitola mé bakalářské práce obsahuje základní definice a věty, které sice nepatří k primárnímu obsahu práce, ale jsou důležité k jejímu pochopení. Tato kapitola je pro snadnější pochopení uvedených tvrzení doplněna příklady.

Druhá kapitola se skládá ze tří podkapitol. Na počátku této kapitoly uvádíme základní myšlenku iteračních metod a tvrzení, která platí pro všechny následně uvedené metody. Podkapitoly nás seznamují postupně s Jacobiovou, Gaussovou–Seidelovou a Relaxační metodou. Každá je obohacena o praktický příklad, který ukončujeme symbolem ♠, a m-file, který lze použít pro výpočet jednotlivých metod. M-fily jsou skripty sestavené v matematickém softwaru Matlab

a budeme je ukončovat symbolem ★. Všechny m-fily jsou uvedeny na příloženém CD.

V poslední kapitole nazvané Analýza metod budeme vytvořené m-fily aplikovat na různé soustavy lineárních rovnic a zkoumat změny hledané aproximace \mathbf{x}^k při změnách vstupních parametrů.

1 Základní pojmy

V této kapitole si uvedeme zásadní definice a věty, jejichž znalost je nezbytná k pochopení této tematiky. Tyto pojmy budeme užívat v následujících kapitolách. Důraz bude kladen především na vysvětlení pojmů systému lineárních rovnic, maticové a vektorové normy a dalších.

V celé této práci budeme uvažovat pouze reálné prvky matic a vektorů. Dále symbolem \mathbb{M}_n budeme značit třídu všech čtvercových matic řádu n s reálnými prvky.

Definice 1.1.

1. Matice $\mathbf{A} \in \mathbb{M}_n$ se nazývá regulární, je-li $\det(\mathbf{A}) \neq 0$.
2. Matice $\mathbf{A} \in \mathbb{M}_n$ se nazývá singulární, je-li $\det(\mathbf{A}) = 0$.

Uvažujme systém lineárních rovnic

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \quad (1)$$

kde \mathbf{A} je reálná čtvercová matice řádu n , tj. $\mathbf{A} \in \mathbb{M}_n$, $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$ je vektor neznámých, $\mathbf{b} \in \mathbb{R}^n$ je reálný vektor. Matice $\mathbf{A} = (a_{ij})_{i,j=1}^n$ a vektor \mathbf{b} jsou tedy ve tvaru

$$\mathbf{A} = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nn} \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}. \quad (2)$$

Je-li matice \mathbf{A} regulární, má systém $\mathbf{A}\mathbf{x} = \mathbf{b}$ jediné řešení \mathbf{x}^* tvaru

$$\mathbf{x}^* = \mathbf{A}^{-1}\mathbf{b}.$$

Abychom se mohli dále zabývat iteračními metodami, musíme si vysvětlit následující pojmy, které budeme používat v dalších kapitolách.

Definice 1.2. Jestliže $\mathbf{A} \in \mathbb{M}_n$, pak \mathbf{A} se nazývá symetrická matice, platí-li $\mathbf{A}^T = \mathbf{A}$.

Definice 1.3. Reálná symetrická matice $\mathbf{A} \in \mathbb{M}_n$ se nazývá pozitivně definitní, platí-li pro libovolný reálný vektor $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{x} \neq \mathbf{o}$, že $\mathbf{x}^T \mathbf{A} \mathbf{x} > 0$.

Věta 1.1. (Sylvestr) Necht' je dána matice $\mathbf{A} = (a_{ij}) \in \mathbb{M}_n$ a necht'

$$\mathbf{A}_i = \begin{pmatrix} a_{11} & \dots & a_{1i} \\ \vdots & \ddots & \vdots \\ a_{i1} & \dots & a_{ii} \end{pmatrix}, \quad \forall i = 1, \dots, n.$$

Pak matice \mathbf{A} je pozitivně definitní, právě když

$$\det(\mathbf{A}_i) > 0, \quad \forall i = 1, \dots, n.$$

Důkaz : viz [9], str. 154.

Definice 1.4. Vektorová norma na \mathbb{R}^n je funkce $\|\cdot\| : \mathbb{R}^n \rightarrow \mathbb{R}$ s následujícími vlastnostmi

1. $\|\mathbf{x}\| \geq 0$, $\forall \mathbf{x} \in \mathbb{R}^n$
2. $\|\mathbf{x}\| = 0 \Leftrightarrow \mathbf{x} = \mathbf{o}$,
3. $\|\alpha \mathbf{x}\| = |\alpha| \|\mathbf{x}\|$, $\forall \alpha \in \mathbb{R}, \forall \mathbf{x} \in \mathbb{R}^n$
4. $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$.

Příkladem vektorových norem jsou

1. $\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$, oktaedrická norma
2. $\|\mathbf{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|$, krychlová norma
3. $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}$, eukleidovská norma.

Příklad 1.1. Vypočítejte pro vektor $\mathbf{x} = (5, 3, -2, 4, 0, -1)^T$ vektorové normy $\|\mathbf{x}\|_1, \|\mathbf{x}\|_\infty$ a $\|\mathbf{x}\|_2$.

Řešení

Oktaedrická norma je dána vztahem $\|\mathbf{x}\|_1 = \sum_{i=1}^n |x_i|$. Tedy pro daný vektor \mathbf{x} dostáváme $\|\mathbf{x}\|_1 = |5| + |3| + |-2| + |4| + |0| + |-1| = 15$.

Krychlová norma je $\|\mathbf{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|$, tj. $\|\mathbf{x}\|_\infty = \max\{5, 3, 2, 4, 0, 1\} = 5$.

Pro eukleidovskou normu platí vztah $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^n |x_i|^2}$, který vypočítáme takto

$$\|\mathbf{x}\|_2 = \sqrt{|5|^2 + |3|^2 + |-2|^2 + |4|^2 + 0 + |-1|^2} = \sqrt{55} = 7.4162.$$

Normy daného vektoru jsou $\|\mathbf{x}\|_1 = 15$, $\|\mathbf{x}\|_\infty = 5$ a $\|\mathbf{x}\|_2 = 7.4162$.



Nyní si uvedeme definici konvergence posloupnosti vektorů $\{\mathbf{x}^k\}$, kde $\mathbf{x}^k \in \mathbb{R}^n$.

Definice 1.5. Posloupnost $\{\mathbf{x}^k\}_{k=1}^\infty$ vektorů z \mathbb{R}^n konverguje k vektoru $\mathbf{x} \in \mathbb{R}^n$ vzhledem k normě $\|\cdot\|$, jestliže pro libovolné $\varepsilon \in \mathbb{R}$, $\varepsilon > 0$ existuje index $N = N(\varepsilon)$, kde $N \in \mathbb{N}$, pro který platí

$$\|\mathbf{x}^k - \mathbf{x}\| < \varepsilon, \quad \forall k \geq N(\varepsilon).$$

Definice 1.6. Maticová norma na množině \mathbb{M}_n je funkce $\|\cdot\| : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ s těmito vlastnostmi

1. $\|\mathbf{A}\| \geq 0, \quad \forall \mathbf{A} \in \mathbb{M}_n$
2. $\|\mathbf{A}\| = 0 \Leftrightarrow \mathbf{A}$ je nulová matice
3. $\|\alpha \mathbf{A}\| = |\alpha| \|\mathbf{A}\|, \quad \forall \alpha \in \mathbb{R}, \forall \mathbf{A} \in \mathbb{M}_n$
4. $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|, \quad \forall \mathbf{A}, \mathbf{B} \in \mathbb{M}_n$
5. $\|\mathbf{AB}\| \leq \|\mathbf{A}\| \|\mathbf{B}\|, \quad \forall \mathbf{A}, \mathbf{B} \in \mathbb{M}_n$.

V některých případech je užitečné požadovat, aby maticové normy měly vlastnost nazývanou souhlasnost. Tato vlastnost určuje závislost normy matice a normy vektoru.

Definice 1.7. Řekneme, že maticová norma $\|\cdot\|$ je souhlasná s danou vektorovou normou $\|\cdot\|_\varphi$, jestliže

$$\|\mathbf{Ax}\|_\varphi \leq \|\mathbf{A}\| \|\mathbf{x}\|_\varphi, \forall \mathbf{x} \in \mathbb{R}^n, \forall \varphi \in \mathbb{R}^*.$$

Věta 1.2. *Nechť $\|\cdot\|_\varphi$ je vektorová norma na \mathbb{R}^n . Pak číslo*

$$\|\mathbf{A}\|_\varphi = \max_{\|\mathbf{x}\|_\varphi=1} \|\mathbf{Ax}\|_\varphi$$

je maticová norma souhlasná s danou vektorovou normou $\|\cdot\|_\varphi$. Tato norma se nazývá přidružená k dané vektorové normě.

Důkaz : viz [2], str. 16.

Věta 1.3. *Nechť maticová norma $\|\cdot\|$ je souhlasná s danou vektorovou normou $\|\cdot\|_\varphi$. Pak pro všechna vlastní čísla λ matice \mathbf{A} platí*

$$|\lambda| \leq \|\mathbf{A}\|.$$

Důkaz : viz [2], str. 18.

Definice 1.8. Nechť $\lambda_1, \dots, \lambda_n$ jsou vlastní čísla matice $\mathbf{A} \in \mathbb{M}_n$. Číslo

$$\rho(\mathbf{A}) = \max_{1 \leq k \leq n} |\lambda_k|$$

se nazývá **spektrální poloměr** matice \mathbf{A} .

Následující věta o přidružených maticových normách je převzata z [2].

Věta 1.4. *Nechť $\mathbf{A} \in \mathbb{M}_n$. Přidružené maticové normy k vektorovým normám $\|\cdot\|_1, \|\cdot\|_\infty, \|\cdot\|_2$ jsou dány vztahy*

$$1. \|\mathbf{A}\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|, \quad \text{sloupcová norma}$$

$$2. \|\mathbf{A}\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|, \quad \text{řádková norma}$$

$$3. \|\mathbf{A}\|_2 = \sqrt{\rho(\mathbf{A}^T \mathbf{A})}, \quad \text{spektrální norma, kde } \rho(\mathbf{A}^T \mathbf{A}) \text{ je spektrální polo-} \\ \text{měr matice } \mathbf{A}^T \mathbf{A}.$$

Příklad 1.2. Vypočítejte pro matici

$$\mathbf{A} = \begin{pmatrix} 1 & 3 & 1 & -2 & 2 \\ 2 & 4 & 3 & -2 & 1 \\ 1 & 2 & -1 & 5 & 3 \\ 3 & 4 & 1 & 3 & 4 \\ 2 & 3 & -1 & 4 & 2 \end{pmatrix}$$

maticové normy $\|\mathbf{A}\|_1$, $\|\mathbf{A}\|_\infty$, $\|\mathbf{A}\|_2$.

Řešení

Sloupcovou normu matice \mathbf{A} vypočítáme z $\|\mathbf{A}\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^n |a_{ij}|$. V Matlabu bychom ji spočítali pomocí příkazu `norm(A, 1)`. Při ručním výpočtu sčítáme absolutní hodnoty prvků matice \mathbf{A} ve sloupcích takto

$$\sum_{i=1}^5 |a_{i1}| = |1| + |2| + |1| + |3| + |2| = 9,$$

$$\sum_{i=1}^5 |a_{i2}| = |3| + |4| + |2| + |4| + |3| = 16,$$

$$\sum_{i=1}^5 |a_{i3}| = |1| + |3| + |-1| + |1| + |-1| = 7,$$

$$\sum_{i=1}^5 |a_{i4}| = |-2| + |-2| + |5| + |3| + |4| = 16,$$

$$\sum_{i=1}^5 |a_{i5}| = |2| + |1| + |3| + |4| + |2| = 12.$$

Výsledkem je tedy $\|\mathbf{A}\|_1 = \max\{9, 16, 7, 16, 12\} = 16$.

Pro řádkovou normu platí vztah $\|\mathbf{A}\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^n |a_{ij}|$. Pokud bychom ji chtěli spočítat Matlabem použijeme příkaz $norm(A, inf)$. Nyní tedy sčítáme absolutní hodnoty prvků matice v řádcích, tj.

$$\sum_{j=1}^5 |a_{1j}| = |1| + |3| + |1| + |-2| + |2| = 9,$$

$$\sum_{j=1}^5 |a_{2j}| = |2| + |4| + |3| + |-2| + |1| = 12,$$

$$\sum_{j=1}^5 |a_{3j}| = |1| + |2| + |-1| + |5| + |3| = 12,$$

$$\sum_{j=1}^5 |a_{4j}| = |3| + |4| + |1| + |3| + |4| = 15,$$

$$\sum_{j=1}^5 |a_{5j}| = |2| + |3| + |-1| + |4| + |2| = 12.$$

Řádková norma matice \mathbf{A} je $\|\mathbf{A}\|_\infty = \max\{9, 12, 12, 15, 12\} = 15$.

Spektrální norma je vyjádřena vztahem $\|\mathbf{A}\|_2 = \sqrt{\rho(\mathbf{A}^T \mathbf{A})}$. Druhou mocninu spektrální normy můžeme vypočítat pomocí Matlabu příkazem $norm(\mathbf{A}' * \mathbf{A})$, kde transponovanou matici získáme z matice \mathbf{A} pomocí \mathbf{A}' . Nebo spektrální poloměr můžeme určit ručním výpočtem. Pro ten potřebujeme nejdříve transponovat původní matici a pak ji vynásobit s původní maticí, tj.

$$\mathbf{A}^T \mathbf{A} = \begin{pmatrix} 1 & 2 & 1 & 3 & 2 \\ 3 & 4 & 2 & 4 & 3 \\ 1 & 3 & -1 & 1 & -1 \\ -2 & -2 & 5 & 3 & 4 \\ 2 & 1 & 3 & 4 & 2 \end{pmatrix} \begin{pmatrix} 1 & 3 & 1 & -2 & 2 \\ 2 & 4 & 3 & -2 & 1 \\ 1 & 2 & -1 & 5 & 3 \\ 3 & 4 & 1 & 3 & 4 \\ 2 & 3 & -1 & 4 & 2 \end{pmatrix} =$$

$$= \begin{pmatrix} 19 & 31 & 7 & 16 & 23 \\ 31 & 54 & 14 & 20 & 38 \\ 7 & 14 & 13 & -14 & 4 \\ 16 & 20 & -14 & 58 & 29 \\ 23 & 38 & 4 & 29 & 34 \end{pmatrix}.$$

Pro získání spektrálního poloměru musíme dále vypočítat vlastní čísla této nové matice. Ty získáme v Matlabu pomocí příkazu $\text{eig}(\mathbf{A}' * \mathbf{A})$. Vlastními čísly jsou $\lambda_1 = 0.4220$, $\lambda_2 = 1.4474$, $\lambda_3 = 2.4806$, $\lambda_4 = 50.3804$ a $\lambda_5 = 123.2696$. Spektrální poloměr je tedy roven $\rho(\mathbf{A}^T \mathbf{A}) = 123.2696$. Tuto hodnotu dosadíme do vztahu pro spektrální normu a získáme $\|\mathbf{A}\|_2 = \sqrt{123.2696} = 11.1027$.

Přidružené maticové normy k vektorovým normám $\|\cdot\|_1, \|\cdot\|_\infty, \|\cdot\|_2$ jsou pro tuto matici rovny $\|\mathbf{A}\|_1 = 16$, $\|\mathbf{A}\|_\infty = 15$ a $\|\mathbf{A}\|_2 = 11.1027$.



Poznámka 1.1. Matlabovský příkaz *norm* můžeme užívat pro výpočet norem matic i vektorů. V případě matic máme následující čtyři možnosti využití

1. $\text{norm}(A, 1)$ tímto příkazem získáme sloupcovou normu matice \mathbf{A} ,
2. $\text{norm}(A, 2) = \text{norm}(A)$ příkazy, jejichž výsledek vyjádří druhou mocninu spektrální normy matice \mathbf{A} ,
3. $\text{norm}(A, \text{inf})$ je příkaz pro řádkovou normu matice \mathbf{A} ,
4. $\text{norm}(A, 'fro')$ pomocí tohoto příkazu vypočítáme Frobeniovu normu.

Pro vektory lze užít následující příkazy

1. $\text{norm}(x, p)$, kde $1 \leq p \leq \infty$, dosazením za p získáme různé normy vektoru \mathbf{x} ,
 - (a) $\text{norm}(x, 2) = \text{norm}(A)$ příkazy, pomocí kterých vypočítáme eukleidovskou normu vektoru \mathbf{x} ,
 - (b) $\text{norm}(x, \text{inf})$ je krychlová norma vektoru \mathbf{x} ,
2. $\text{norm}(x, -\text{inf})$ nalezneme v Matlabu $\text{min}(\text{abs}(x))$.

2 Iterační metody

Metody pro řešení systému lineárních rovnic

$$\mathbf{Ax} = \mathbf{b},$$

kde matice $\mathbf{A} \in \mathbb{M}_n$ a vektor $\mathbf{b} \in \mathbb{R}^n$, se rozdělují do dvou skupin. Těmi jsou metody přímé a iterační.

Přímé metody řeší soustavy lineárních rovnic předem známým konečným počtem kroků. V ideálním případě bychom se dostali po ukončení výpočtu k přesnému řešení. Tento ideální případ ale většinou nenastává z důvodu vzniku zaokrouhlovacích chyb během výpočtu. Jedním z možných způsobů výpočtu je eliminace neznámých. Základní algoritmus tohoto typu popisuje Gaussova eliminační metoda. Pokud bychom měli n neznámých, tj. n složek hledaného vektoru \mathbf{x} , pak k nalezení řešení povede $n - 1$ kroků. Jednu neznámou vyjádříme z některé rovnice a dosadíme ji do ostatních rovnic. Takto postupně snižujeme počet neznámých v lineárních rovnicích až zůstane pouze jedna rovnice o jedné neznámé. Tu vypočítáme a následně hodnotu této neznámé dosadíme do rovnice, ze které jsme si vyjádřili předchozí neznámou, a vypočítáme další neznámou. Takto postupně dosazujeme hodnoty vypočítaných neznámých do rovnic až nakonec dostáváme hodnoty všech neznámých a získáváme přesné řešení dané soustavy.

Iterační metody užíváme zejména v případech, kdy řešíme soustavu n lineárních rovnic o n neznámých, přičemž počet neznámých n je velký. Přímé metody řešení těchto rovnic by byly velice zdlouhavé a snadno bychom se mohli dopustit numerické chyby. Iterační metody řeší soustavy lineárních rovnic pomocí postupného přibližování k přesnému řešení. Pomocí nich konstruujeme posloupnost vektorů $\{\mathbf{x}^k\}$ konvergující za určitých podmínek k přesnému řešení soustavy. Iterační metody umožňují tedy nalézt řešení soustav lineárních rovnic pomocí konvergentních procesů, které skončí až bude pro posloupnost $\{\mathbf{x}^k\}$ splněna podmínka zastavovacího kritéria. Iterační metody tedy dávají vždy pouze jistou aproximaci řešení. Je to způsobeno tím, že můžeme realizovat jen konečný počet kroků vý-

počtu.

Některé z vlastností iteračních metod jsou

1. V každé iteraci známe aproximaci řešení \mathbf{x}^k . Iteracemi postupně vypočítáme posloupnost vektorů aproximací $\{\mathbf{x}^k\}$ takovou, že pokud existuje $\lim_{k \rightarrow \infty} \mathbf{x}^k$, pak posloupnost $\{\mathbf{x}^k\}$ konverguje k přesnému řešení \mathbf{x}^* . Aproximaci \mathbf{x}^k označíme za dostatečně přesnou, pokud je splněno kritérium pro ukončení výpočtu (viz poznámka 2.2), a výpočet ukončíme.
2. Iterační metody jsou méně citlivé na zaokrouhlovací chyby než metody přímé. Každá iterace se považuje za počáteční, a proto zaokrouhlovací chyby z předchozích operací vymizí, v případě že posloupnost \mathbf{x}^k konverguje k přesnému řešení.

V další části této kapitoly si popíšeme teoretickou část iteračních metod, uvedeme konkrétní iterační metody a na příkladech si vysvětlíme postup jejich aplikace.

Vyjádříme systém $\mathbf{Ax} = \mathbf{b}$, kde $\mathbf{A} \in \mathbb{M}_n$ a $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$, v ekvivalentním tvaru

$$\mathbf{x} = \mathbf{T}\mathbf{x} + \mathbf{g}, \quad \mathbf{T} \in \mathbb{M}_n.$$

Je-li $\mathbf{x}^* = (\mathbf{I} - \mathbf{T})^{-1}\mathbf{g}$ přesným řešením systému rovnic $\mathbf{x} = \mathbf{T}\mathbf{x} + \mathbf{g}$, pak je také přesným řešením ekvivalentní soustavy rovnic $\mathbf{Ax} = \mathbf{b}$. To platí i opačně, tj. pokud je vektor $\mathbf{x}^* = \mathbf{A}^{-1}\mathbf{b}$ přesným řešením systému rovnic $\mathbf{Ax} = \mathbf{b}$, pak je \mathbf{x}^* také řešením $\mathbf{x} = \mathbf{T}\mathbf{x} + \mathbf{g}$. Je to z důvodu ekvivalentního převodu systému rovnic $\mathbf{Ax} = \mathbf{b}$ na systém $\mathbf{x} = \mathbf{T}\mathbf{x} + \mathbf{g}$. Dále nechť $\mathbf{x}^0 \in \mathbb{R}^n$ je libovolnou počáteční aproximací. Posloupnost $\{\mathbf{x}^k\}$ generovanou vztahem

$$\mathbf{x}^{k+1} = \mathbf{T}\mathbf{x}^k + \mathbf{g}, \quad k = 0, 1, \dots \quad (3)$$

nazveme **iterační** a matici \mathbf{T} pojmenujeme iterační maticí.

Hlavním úkolem bude vyřešit následující otázky.

1. Jakým způsobem převést $\mathbf{Ax} = \mathbf{b}$ na ekvivalentní systém $\mathbf{x} = \mathbf{T}\mathbf{x} + \mathbf{g}$?
2. Za jakých předpokladů bude posloupnost $\{\mathbf{x}^k\}$ generovaná vztahem (3) konvergovat pro libovolnou počáteční aproximaci k přesnému řešení \mathbf{x}^* ?

Věta 2.1. Posloupnost $\{\mathbf{x}^k\}$ generovaná vztahem

$$\mathbf{x}^{k+1} = \mathbf{T}\mathbf{x}^k + \mathbf{g},$$

konverguje k přesnému řešení \mathbf{x}^* soustavy $\mathbf{A}\mathbf{x} = \mathbf{b}$ pro každou volbu počáteční aproximaci $\mathbf{x}^0 \in \mathbb{R}^n$, právě když spektrální poloměr iterační matice \mathbf{T} je menší než jedna, tj. $\rho(\mathbf{T}) < 1$, přičemž $\lim_{k \rightarrow \infty} \mathbf{x}^k = \mathbf{x}^*$, $\mathbf{x}^* = \mathbf{T}\mathbf{x}^* + \mathbf{g}$.

Důkaz viz [7] str. 98.

Důsledek 2.1. Je-li některá z přidružených norem matice \mathbf{T} menší než jedna, pak posloupnost $\{\mathbf{x}^k\}$ generovaná iteračním procesem (3) konverguje k přesnému řešení $\mathbf{x}^* = (\mathbf{I} - \mathbf{T})^{-1}\mathbf{g}$ nezávisle na volbě počáteční aproximace $\mathbf{x}^0 \in \mathbb{R}^n$. Dále platí

$$\|\mathbf{x}^* - \mathbf{x}^k\| \leq \|\mathbf{T}\|^k \|\mathbf{x}^* - \mathbf{x}^0\|, \quad (4)$$

$$\|\mathbf{x}^* - \mathbf{x}^k\| \leq \frac{\|\mathbf{T}\|^k}{1 - \|\mathbf{T}\|} \|\mathbf{x}^1 - \mathbf{x}^0\|. \quad (5)$$

Důkaz viz [2] str. 138.

Tento důsledek plyne z věty 1.3 a definice 1.8. Tato věta formuluje podmínku $|\lambda| \leq \|\mathbf{T}\|$, která platí pro všechna vlastní čísla matice \mathbf{T} . Definice vyjadřuje, že spektrální poloměr je roven největšímu vlastnímu číslu v absolutní hodnotě. Z toho vyplývá $\rho(\mathbf{T}) \leq \|\mathbf{T}\|$. To znamená, že můžeme podmínku $\rho(\mathbf{T}) < 1$ ve větě 2.1 nahradit podmínkou $\|\mathbf{T}\| < 1$.

Jestliže zvolíme počáteční aproximaci $\mathbf{x}^0 = \mathbf{g}$, dostáváme ze vztahu (5)

$$\|\mathbf{x}^* - \mathbf{x}^k\| \leq \frac{\|\mathbf{T}\|^{k+1}}{1 - \|\mathbf{T}\|} \|\mathbf{g}\|, \quad (6)$$

což si odvodíme následovně.

Užitím vztahu (3) lze psát

$$\|\mathbf{x}^1 - \mathbf{x}^0\| = \|\mathbf{T}\mathbf{x}^0 + \mathbf{g} - \mathbf{x}^0\|.$$

Je-li $\mathbf{x}^0 = \mathbf{g}$, pak

$$\|\mathbf{x}^1 - \mathbf{x}^0\| = \|\mathbf{T}\mathbf{g} + \mathbf{g} - \mathbf{g}\| = \|\mathbf{T}\mathbf{g}\| \leq \|\mathbf{T}\| \cdot \|\mathbf{g}\|.$$

Po dosazení do pravé strany nerovnice (5) získáváme nerovnici tvaru (6)

$$\|\mathbf{x}^* - \mathbf{x}^k\| \leq \frac{\|\mathbf{T}\|^k}{1 - \|\mathbf{T}\|} \|\mathbf{x}^1 - \mathbf{x}^0\| \leq \frac{\|\mathbf{T}\|^k}{1 - \|\mathbf{T}\|} \|\mathbf{T}\| \cdot \|\mathbf{g}\| = \frac{\|\mathbf{T}\|^{k+1}}{1 - \|\mathbf{T}\|} \|\mathbf{g}\|.$$

Pomocí vzorce (6) lze určit počet iterací potřebný k nalezení aproximace řešení dané soustavy lineárních rovnic s námi určenou přesností, což bude prezentováno v příkladech 2.2, 2.6 a 2.10.

Poznámka 2.1. Rychlost konvergence iteračních metod ovlivňuje spektrální poloměr $\rho(\mathbf{T})$. Proto jakákoliv úprava, která zmenší spektrální poloměr této matice, povede ke zrychlení této konvergence.

Poznámka 2.2. Jak již bylo řečeno, iterační metody řeší soustavy lineárních rovnic, v případě jejich konvergence, pomocí postupného přibližování k přesnému řešení. Nedostáváme tedy přesné řešení soustavy, a proto musíme určit, která $(k+1)$ -ní iterace \mathbf{x}^{k+1} je dostatečně přesnou aproximací přesného řešení. Pro její zjištění potřebujeme mít stanovenou požadovanou přesnost ϵ , pro kterou platí vztah $\|\mathbf{x}^k - \mathbf{x}^*\| < \epsilon$. Pokud bude počáteční aproximace rovna vektoru \mathbf{g} můžeme počet iterací k vypočítat užitím vztahu (6). V obecném případě, tj. při obecné volbě vektoru \mathbf{x}^0 , ale neověřujeme požadovanou přesnost aproximace \mathbf{x}^{k+1} z odchylky rozdílu $\|\mathbf{x}^k - \mathbf{x}^*\| < \epsilon$, protože neznáme přesné řešení \mathbf{x}^* , nýbrž užíváme některého ze zastavovacích kritérií, např. tvaru

1. $\frac{\|\mathbf{x}^{k+1} - \mathbf{x}^k\|}{\|\mathbf{x}^k\|} < \epsilon$,
2. $\|\mathbf{r}^{k+1}\| \leq \epsilon (\|\mathbf{A}\| \|\mathbf{x}^{k+1}\| + \|\mathbf{b}\|)$, kde $\mathbf{r}^{k+1} = \mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}$,

maticová norma je přidružená dané vektorové normě.

2.1 Jacobiova metoda

Metoda, kterou se budeme zabývat jako první se nazývá Jacobiova. Je známá pod více názvy, např. Metoda současné opravy, jak uvádí [6]. Původní soustavu lineárních rovnic $\mathbf{Ax} = \mathbf{b}$ lze vyjádřit v ekvivalentním tvaru

$$\mathbf{x} = \mathbf{T}_J \mathbf{x} + \mathbf{g}_J. \quad (7)$$

Tento převod provedeme následujícím postupem.

Nejprve matici \mathbf{A} , kde $\mathbf{A} = (a_{ij}) \in \mathbb{M}_n$, zapíšeme ve tvaru

$$\mathbf{A} = \mathbf{D} - \mathbf{L} - \mathbf{U}, \quad (8)$$

kde

$$\mathbf{D} = \begin{pmatrix} a_{11} & & 0 \\ & \ddots & \\ 0 & & a_{nn} \end{pmatrix}, \quad (9)$$

$$\mathbf{L} = \begin{pmatrix} 0 & & & 0 \\ -a_{21} & \ddots & & \\ \vdots & \ddots & \ddots & \\ -a_{n1} & \dots & -a_{n,n-1} & 0 \end{pmatrix}, \quad (10)$$

$$\mathbf{U} = \begin{pmatrix} 0 & -a_{12} & \dots & -a_{1n} \\ & \ddots & \ddots & \vdots \\ & & \ddots & -a_{n-1,n} \\ 0 & & & 0 \end{pmatrix}. \quad (11)$$

tedy matice \mathbf{D} je diagonální a tvoří ji prvky na hlavní diagonále matice \mathbf{A} . Matice \mathbf{L} je ostře dolní trojúhelníková s nulami na hlavní diagonále a skládá se z prvků matice \mathbf{A} s opačnými znaménky, které se nacházejí pod hlavní diagonálou. Podobně, matice \mathbf{U} je ostře horní trojúhelníková matice s nulami na hlavní diagonále, jejíž prvky jsme získali z matice \mathbf{A} . Konkrétně to jsou prvky s obrácenými znaménky nalézající se v horní trojúhelníkové části.

Následně rovnici $\mathbf{Ax} = \mathbf{b}$ zapíšeme ve tvaru $(\mathbf{D} - \mathbf{L} - \mathbf{U})\mathbf{x} = \mathbf{b}$ a upravíme ji na ekvivalentní rovnici

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

Za předpokladu, že $a_{ii} \neq 0, \forall i = 1, \dots, n$, je matice \mathbf{D} regulární a z předchozí rovnice dostáváme

$$\mathbf{x} = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{x} + \mathbf{D}^{-1}\mathbf{b}.$$

Tento vztah nazýváme maticový tvar **Jacobiovy iterační metody**. Označíme-li $\mathbf{T}_J = \mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})$ a $\mathbf{g}_J = \mathbf{D}^{-1}\mathbf{b}$, je tato metoda tvaru

$$\mathbf{x}^{k+1} = \mathbf{T}_J\mathbf{x}^k + \mathbf{g}_J, \quad (12)$$

kde $\mathbf{T}_J = (t_{ij}) \in \mathbb{M}_n$ je **Jacobiova iterační matice**, $t_{ij} = -\frac{a_{ij}}{a_{ii}}$ pro $i \neq j$, $t_{ii} = 0$ pro $\forall i, j = 1, \dots, n$. Matice \mathbf{T}_J má tedy nulové diagonální prvky. Pro sloupcový vektor $\mathbf{g}_J = (g_1, \dots, g_n)^T$ je $g_i = \frac{b_i}{a_{ii}}$ pro $\forall i = 1, \dots, n$. Matice \mathbf{T}_J a vektor \mathbf{g}_J jsou tedy tvaru

$$\mathbf{T}_J = \begin{pmatrix} 0 & -\frac{a_{12}}{a_{11}} & \dots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & & -\frac{a_{2n}}{a_{22}} \\ \vdots & & \ddots & \vdots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & \dots & 0 \end{pmatrix}, \quad \mathbf{g}_J = \begin{pmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ \vdots \\ \frac{b_n}{a_{nn}} \end{pmatrix}. \quad (13)$$

Jacobiovu iterační metodu lze zapsat také ve složkách, kde $\mathbf{x}^k = (x_1^k, \dots, x_n^k)^T$, následovně

$$x_i^{k+1} = - \sum_{j=1, j \neq i}^n \frac{a_{ij}}{a_{ii}} x_j^k + \frac{b_i}{a_{ii}}, \quad i = 1, \dots, n; \quad k = 0, 1, 2, \dots$$

Postupnými výpočty nám z první rovnice vyjde x_1^{k+1} , z druhé x_2^{k+1} a obecně z i -té rovnice vypočteme x_i^{k+1} až z n -té rovnice x_n^{k+1} .

Věta 2.2. *Posloupnost $\{\mathbf{x}^k\}$ generovaná vztahem (12) konverguje pro každou počáteční aproximaci $\mathbf{x}^0 \in \mathbb{R}^n$ právě tehdy, když $\rho(\mathbf{T}_J) < 1$.*

Tuto větu jsme uvedli již jako větu 2.1 v předchozí kapitole a zde jsme ji upravili pro Jacobiovu iterační metodu.

Následující věta zaručuje konvergenci Jacobiovu iterační metody pro níže uvedené speciální typy matic.

Věta 2.3. a) *Silné řádkové sumační kritérium*

Nechť matice \mathbf{A} je ryze řádkově diagonálně dominantní, tj.

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

V tomto případě Jacobiova iterační metoda konverguje pro každou počáteční aproximaci $\mathbf{x}^0 \in \mathbb{R}^n$.

b) *Silné sloupcové sumační kritérium*

Nechť matice \mathbf{A} je ryze sloupcově diagonálně dominantní, tj.

$$|a_{kk}| > \sum_{i=1, i \neq k}^n |a_{ik}|, \quad k = 1, \dots, n.$$

Pak Jacobiova iterační metoda konverguje nezávisle na volbě počáteční aproximace $\mathbf{x}^0 \in \mathbb{R}^n$.

Důkaz viz [2] str. 142.

Příklad 2.1. Řešte soustavu lineárních rovnic

$$\begin{aligned}x_1 + x_3 &= -2 \\-x_1 + 3x_2 + 4x_3 &= 3 \\-4x_1 + x_2 + 2x_3 &= 4\end{aligned}$$

Jacobiovou iterační metodou.

Řešení

Přesným řešením tohoto systému lineárních rovnic je vektor $\mathbf{x}^* = (-1, 2, -1)^T$,

což lze jednoduše ověřit dosazením do dané soustavy. Původní soustavu rovnic můžeme přepsat do maticového tvaru Jacobiovy iterační metody

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 \\ \frac{1}{3} & 0 & -\frac{4}{3} \\ \frac{4}{2} & -\frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} -2 \\ \frac{3}{3} \\ \frac{4}{2} \end{pmatrix}.$$

Matice \mathbf{T}_J má tedy tvar

$$\mathbf{T}_J = \begin{pmatrix} 0 & 0 & -1 \\ \frac{1}{3} & 0 & -\frac{4}{3} \\ 2 & -\frac{1}{2} & 0 \end{pmatrix}.$$

Matice dané soustavy lineárních rovnic není ryze řádkově ani sloupcově diagonálně dominantní, a proto o konvergenci Jacobiova iteračního procesu nelze rozhodnout užitím věty 2.3. Proto na základě věty 2.2 vypočítáme spektrální poloměr matice \mathbf{T}_J . Pokud bude menší než jedna, je posloupnost $\{\mathbf{x}^k\}$ generovaná Jacobiovou iterační metodou konvergentní a konverguje k přesnému řešení. K výpočtu spektrálního poloměru potřebujeme nejdříve znát vlastní čísla matice \mathbf{T}_J . Ty určíme buď matlabovským příkazem $\text{eig}(\mathbf{T}_J)$, nebo z následujícího determinantu

$$\begin{aligned} \det(\mathbf{T}_J - \lambda \mathbf{I}) &= \begin{vmatrix} -\lambda & 0 & -1 \\ \frac{1}{3} & -\lambda & -\frac{4}{3} \\ 2 & -\frac{1}{2} & -\lambda \end{vmatrix} = -\lambda^3 + 0 + \frac{1}{6} - \left(2\lambda - \frac{2}{3}\lambda + 0\right) = \\ &= -\lambda^3 - \frac{4}{3}\lambda + \frac{1}{6}. \end{aligned} \tag{14}$$

Neznámá vlastní čísla matice \mathbf{T}_J vypočítáme jako kořeny polynomu (14) matlabovským příkazem $\text{roots}([-1 \ 0 \ -\frac{4}{3} \ \frac{1}{6}])$, přičemž čísla v hranaté závorce jsou indexy tohoto polynomu. Vlastní čísla iterační matice \mathbf{T}_J tedy jsou $\lambda_1 = 0.1236$ a $\lambda_{2,3} = -0.0618 \pm 1.1597i$. Spektrální poloměr matice \mathbf{T}_J je dle definice 1.8 roven $\rho(\mathbf{T}_J) = \max\{|-0.0618 \pm 1.1597i|, 0.1236\}$, tedy $\rho(\mathbf{T}_J) = 1.1613 > 1$. Absolutní hodnotu komplexního čísla $|-0.0618 \pm 1.1597i|$ jsem určila matlabovským příkazem $\text{abs}(-0.0618 + 1.1597i)$. Z toho vyplývá dle věty 2.2, že posloupnost $\{\mathbf{x}^k\}$

generovaná Jacobiovou iterační metodou je divergentní. I přesto, že je tato posloupnost divergentní, uvedeme si první čtyři iterace, abychom si předvedli jejich výpočet. Iterační proces Jacobiovy metody je v tomto případě tvaru

$$\begin{pmatrix} x_1^{k+1} \\ x_2^{k+1} \\ x_3^{k+1} \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 \\ \frac{1}{3} & 0 & -\frac{4}{3} \\ 2 & -\frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} x_1^k \\ x_2^k \\ x_3^k \end{pmatrix} + \begin{pmatrix} -2 \\ 1 \\ 2 \end{pmatrix}.$$

Dosadíme-li například za počáteční aproximaci $\mathbf{x}^0 = (0, 0, 0)^T$ dostáváme první iteraci $\mathbf{x}^1 = (-2, 1, 2)^T$. Pro výpočet \mathbf{x}^2 uijeme již vypočítanou první iteraci

$$\begin{pmatrix} x_1^2 \\ x_2^2 \\ x_3^2 \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 \\ \frac{1}{3} & 0 & -\frac{4}{3} \\ 2 & -\frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} -2 \\ 1 \\ 2 \end{pmatrix} + \begin{pmatrix} -2 \\ 1 \\ 2 \end{pmatrix}.$$

Hodnota druhé iterace je $\mathbf{x}^2 = (-4, -\frac{7}{3}, -\frac{5}{2})^T$. Pro výpočet \mathbf{x}^3 uijeme \mathbf{x}^2

$$\begin{pmatrix} x_1^3 \\ x_2^3 \\ x_3^3 \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 \\ \frac{1}{3} & 0 & -\frac{4}{3} \\ 2 & -\frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} -4 \\ -\frac{7}{3} \\ -\frac{5}{2} \end{pmatrix} + \begin{pmatrix} -2 \\ 1 \\ 2 \end{pmatrix}.$$

Výslednou hodnotou je sloupcový vektor $\mathbf{x}^3 = (\frac{1}{2}, 3, -\frac{29}{6})^T$. Ke čtvrté iteraci \mathbf{x}^4 se dostáváme následovně

$$\begin{pmatrix} x_1^4 \\ x_2^4 \\ x_3^4 \end{pmatrix} = \begin{pmatrix} 0 & 0 & -1 \\ \frac{1}{3} & 0 & -\frac{4}{3} \\ 2 & -\frac{1}{2} & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{2} \\ 3 \\ -\frac{29}{6} \end{pmatrix} + \begin{pmatrix} -2 \\ 1 \\ 2 \end{pmatrix},$$

a tedy $\mathbf{x}^4 = (\frac{17}{6}, \frac{137}{18}, \frac{3}{2})^T$.

Velká odlišnost výsledků jednotlivých iterací a přesného řešení $\mathbf{x}^* = (-1, 2, -1)^T$ koresponduje s divergencí posloupnosti $\{\mathbf{x}^k\}$ generované Jacobiovou iterační metodou.



V následujícím příkladě uvedeme soustavu lineárních rovnic, pro kterou posloupnost $\{\mathbf{x}^k\}$ generovaná Jacobiovou metodou konverguje.

Příklad 2.2. Řešte soustavu lineárních rovnic

$$6x_1 + 3x_2 + 2x_3 = 19$$

$$x_1 + 6x_2 + 3x_3 = 14$$

$$x_1 + x_2 + 6x_3 = 15$$

Jacobiovou iterační metodou a určete počet iterací potřebných k nalezení aproximace řešení dané soustavy s přesností 10^{-3} .

Řešení

Příklad budeme řešit totožným postupem jako předchozí. Přesným řešením je vektor $\mathbf{x}^* = (2, 1, 2)^T$. Maticově zapíšeme soustavu rovnic následovně

$$\begin{pmatrix} 6 & 3 & 2 \\ 1 & 6 & 3 \\ 1 & 1 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 19 \\ 14 \\ 15 \end{pmatrix}.$$

Jacobiova iterační metoda je tvaru

$$\begin{pmatrix} x_1^{k+1} \\ x_2^{k+1} \\ x_3^{k+1} \end{pmatrix} = \begin{pmatrix} 0 & -\frac{1}{2} & -\frac{1}{3} \\ -\frac{1}{6} & 0 & -\frac{1}{2} \\ -\frac{1}{6} & -\frac{1}{6} & 0 \end{pmatrix} \begin{pmatrix} x_1^k \\ x_2^k \\ x_3^k \end{pmatrix} + \begin{pmatrix} \frac{19}{6} \\ \frac{14}{6} \\ \frac{15}{6} \end{pmatrix}.$$

Matice \mathbf{T}_J má tedy tvar

$$\mathbf{T}_J = \begin{pmatrix} 0 & -\frac{1}{2} & -\frac{1}{3} \\ -\frac{1}{6} & 0 & -\frac{1}{2} \\ -\frac{1}{6} & -\frac{1}{6} & 0 \end{pmatrix}.$$

Vektor \mathbf{g}_J je tvaru

$$\mathbf{g}_J = \begin{pmatrix} \frac{19}{6} \\ \frac{14}{6} \\ \frac{15}{6} \end{pmatrix}.$$

Matice dané soustavy lineárních rovnic je ryze řádkově diagonálně dominantní, a proto je dle věty 2.3 posloupnost $\{\mathbf{x}^k\}$ generovaná Jacobiovou iterační metodou konvergentní.

Za počáteční aproximaci \mathbf{x}^0 zvolíme vektor $\mathbf{g}_J = \left(\frac{19}{6}, \frac{7}{3}, \frac{5}{2}\right)^T$, abychom vyhověli podmínkám pro následné uplatnění i ve vztahu (6), a dostáváme

$$\begin{pmatrix} x_1^1 \\ x_2^1 \\ x_3^1 \end{pmatrix} = \begin{pmatrix} 0 & -\frac{1}{2} & -\frac{1}{3} \\ -\frac{1}{6} & 0 & -\frac{1}{2} \\ -\frac{1}{6} & -\frac{1}{6} & 0 \end{pmatrix} \begin{pmatrix} \frac{19}{6} \\ \frac{7}{3} \\ \frac{5}{2} \end{pmatrix} + \begin{pmatrix} \frac{19}{6} \\ \frac{7}{3} \\ \frac{5}{2} \end{pmatrix}.$$

Výpočtem získáme $\mathbf{x}^1 = \left(\frac{7}{6}, \frac{5}{9}, \frac{19}{12}\right)^T = (1.1667; 0.5556; 1.5833)^T$. Pro výpočet iterace \mathbf{x}^2 uijeme sloupcový vektor \mathbf{x}^1

$$\begin{pmatrix} x_1^2 \\ x_2^2 \\ x_3^2 \end{pmatrix} = \begin{pmatrix} 0 & -\frac{1}{2} & -\frac{1}{3} \\ -\frac{1}{6} & 0 & -\frac{1}{2} \\ -\frac{1}{6} & -\frac{1}{6} & 0 \end{pmatrix} \begin{pmatrix} \frac{7}{6} \\ \frac{5}{9} \\ \frac{19}{12} \end{pmatrix} + \begin{pmatrix} \frac{19}{6} \\ \frac{7}{3} \\ \frac{5}{2} \end{pmatrix}.$$

Druhá iterace je tvaru $\mathbf{x}^2 = \left(\frac{85}{36}, \frac{97}{72}, \frac{239}{108}\right)^T = (2.3611; 1.3472; 2.2130)^T$. Třetí iteraci \mathbf{x}^3 vypočítáme za použití \mathbf{x}^2

$$\begin{pmatrix} x_1^3 \\ x_2^3 \\ x_3^3 \end{pmatrix} = \begin{pmatrix} 0 & -\frac{1}{2} & -\frac{1}{3} \\ -\frac{1}{6} & 0 & -\frac{1}{2} \\ -\frac{1}{6} & -\frac{1}{6} & 0 \end{pmatrix} \begin{pmatrix} \frac{85}{36} \\ \frac{97}{72} \\ \frac{239}{108} \end{pmatrix} + \begin{pmatrix} \frac{19}{6} \\ \frac{7}{3} \\ \frac{5}{2} \end{pmatrix}$$

a dostáváme $\mathbf{x}^3 = \left(\frac{2275}{1296}, \frac{5}{6}, \frac{271}{144}\right)^T = (1.7554; 0.8333; 1.8819)^T$. Z důvodu názornosti výpočtů, jsme si uvedli první tři iterace. Zdoluhavost a obtížnost následných ručních výpočtů je důvodem provedení právě pouze těchto první tří iterací. Hledanou aproximaci přesného řešení nalezneme užitím m-filu 2.1.

Pro určení počtu iterací k potřebného k nalezení aproximace \mathbf{x}^k s přesností ϵ uijeme vztahu (6). Nejdříve vypočítáme normu matice \mathbf{T}_J a vektoru \mathbf{g}_J . Řádková norma matice \mathbf{T}_J je $\|\mathbf{T}_J\|_\infty = \max\left\{\frac{5}{6}, \frac{4}{6}, \frac{2}{6}\right\} = \frac{5}{6}$. Krychlová norma vektoru \mathbf{g}_J je $\|\mathbf{g}_J\|_\infty = \max\left\{\frac{19}{6}, \frac{14}{6}, \frac{15}{6}\right\} = \frac{19}{6}$. Nyní opět položíme počáteční aproximaci \mathbf{x}^0 rovnu vektoru $\mathbf{g}_J = \left(\frac{19}{6}, \frac{14}{6}, \frac{15}{6}\right)^T$, protože je to nejvhodnější volba počáteční aproximace pro pozdější uplatnění vztahu (6). Jestliže je $\frac{\|\mathbf{T}_J\|^{k+1}}{1-\|\mathbf{T}_J\|} \|\mathbf{g}_J\| < \epsilon$, pak dle

vztahu (6) je $\|\mathbf{x}^* - \mathbf{x}^k\| < \epsilon$, tj. \mathbf{x}^k je požadovaná aproximace přesného řešení, tj. chyba je menší než 10^{-3} . Hodnotu k pro dané ϵ vypočítáme ze vztahu

$$\frac{\|\mathbf{T}_J\|^{k+1}}{1 - \|\mathbf{T}_J\|} \|\mathbf{g}_J\| < \epsilon.$$

Do vzorce dosadíme hodnoty již vypočítaných norem $\|\mathbf{T}_J\|$ a $\|\mathbf{g}_J\|$ a hledáme $k \in \mathbb{N}$ takové, aby platilo

$$\frac{\left(\frac{5}{6}\right)^{k+1} 19}{1 - \frac{5}{6} 6} < 10^{-3},$$

po úpravě

$$\left(\frac{5}{6}\right)^{k+1} 19 < 10^{-3}.$$

Nerovnici zlogaritmujeme a upravíme, tj.

$$(k+1) \log \frac{5}{6} < \log(10^{-3}) - \log 19,$$

$$k > \frac{-3 - \log 19}{\log \frac{5}{6}} - 1.$$

Následně získáme hledanou hodnotu k

$$k > 53.0375.$$

Pro chybu k -té aproximace s přesností $\epsilon = 10^{-3}$ je tedy nutné provést alespoň $k = 54$ iterací. První tři iterace sice nejsou dostačujícími k vyslovení závěru, ale použitím věty 2.3 jsme zjistili, že posloupnost konverguje k přesnému řešení.



V následující části této podkapitoly uvedeme m-file naprogramovaný v matematickém softwaru Matlab, který jsem sama vytvořila. M-file umí ověřit, zda si odpovídají rozměry vstupních parametrů. Pokud ne, ukončí výpočet a vypíše

chybové hlášení. Pouze v případě, kdy bude vstupní parametr počáteční aproximace prázdný, umí použít vypočítaný vektor \mathbf{g}_J . Zastavovací kritérium použité v m-filu je převzato z prvního bodu poznámky 2.2. Přičemž maximální počet provedených iterací bude 10000. Toto číslo je dostatečně velké a zabraňuje případnému zacyklení m-filu. Pokud bude soustava lineárních rovnic konvergovat, pak program vypíše pro Jacobiovu iterační metodu počet provedených iterací k se zadanou přesností ϵ a poslední provedenou iteraci. Jinak vypíše divergenci metody. Tento počet iterací k bude odlišný od počtu iterací k uvedených v příkladu 2.2. Pro výpočet k v příkladech uvedených ručním výpočtem používáme vztah (6) a musíme tedy volit \mathbf{x}^0 rovno \mathbf{g}_J proto, abychom mohli odhadnout potřebný počet iterací. M-file je vytvořen pro obecnější aplikace, můžeme v něm tedy volit různé počáteční aproximace a přesnosti ϵ .

M-file 2.1.

```
function [x,k]=JACOBY(A,b,x0,eps)
% Jacobiova metoda pro řešení systému lineárních rovnic
% vstupní data:  A   matice tvořená koeficienty soustavy rovnic
%               b   vektor pravých stran dané soustavy rovnic
%               x0  počáteční aproximace
%               eps požadovaná přesnost
% výstupní data: x   hledaná iterace
%               k   počet provedených iterací
% rozměr vstupních hodnot
n=size(A,1);
n1=size(b,1);
n2=size(x0,1);
% maximální počet provedených iterací
K=10000;
% ověření rozměru A a b, kladnosti epsilon
if(n==n1 && eps>0)
    % výpočet matice T a vektoru g
    a=diag(A);
    D=diag(a);
```

```

L=(tril(A)-D)*(-1);
U=(triu(A)-D)*(-1);
X=L+U;
T=D\X; % výpočet \ inverzní matice
g=D\b;
spekt=max(abs(eig(T)));
if (spekt>=1)
    disp('Info: Zvolte jinou metodu.');
```

error('Chyba: Posloupnost diverguje!');

```

end
% ověření neprázdnosti vektoru x0
if (isempty(x0)==1)
    x=g; % pokud je prázdný, pak načte vektor g
else
    x=x0; % neprázdný, použije vektor ze vstupu
    if (n~=n2)
        error('Chyba: Zkontrolujte zda odpovídá rozměr vektoru x0
                a vektoru b!')
```

end

```

end
% výpočet iterací, nejvíc provede K iterací
for k1=0:K
    z0=x;
    for i=1:n
        w=0;
        for j=1:n
            w=w+(T(i,j)*x(j));
            if j==n
                x(i)=w+g(i);
            end
        end
    end
end

```

```

        end
    end
    z1=x;
    odchylka=norm(z1-z0)/norm(z0);
    if (odchylka<eps)
        disp('Info:Konec výpočtu, byla dosažena požadovaná
                přesnost.');
```

k=k1+1;

```

        break;
    end
end
else
    error('Chyba: Zkontrolujte zda odpovídá stupeň matice A veli-
        kosti vektoru b, nebo zda je kladné epsilon!')
```

end

end



V následujícím příkladě aplikujeme m-file 2.1 na příklad 2.2. Středník na konci řádku přikazuje Matlabu, aby znovu nevypisoval na obrazovku daný řádek. Vektor počáteční aproximace \mathbf{x}^0 se rovná množině prázdné, protože m-file je naprogramovaný tak, aby prověřil neprázdnost \mathbf{x}^0 . Pokud zjistí, že je \mathbf{x}^0 prázdné, pak zvolí počáteční aproximaci $\mathbf{x}^0 = \mathbf{g}_J$.

Příklad 2.3. Matice \mathbf{A} je dle věty 2.3 ryze řádkově diagonálně dominantní, a proto je zaručena konvergence posloupnosti \mathbf{x}^k generované Jacobiovou iterační metodou. Nyní uijeme pro výpočet této soustavy m-file. Zadáme do Matlabu vstupní data a tento m-file zavoláme.

Řešení

```
>> A=[6 3 2;1 6 3;1 1 6];
```

```
>> b=[19;14;15];
>> x0=[ ];
>> eps=0.0010;
>> [x,k]=JACOBY(A,b,x0,eps)
```

Info: Konec výpočtu, byla dosažena požadovaná přesnost.

```
x =
    1.9997    1.0001    2.0000
k =
    6
```

K nalezení aproximace řešení dané soustavy s přesností $\epsilon = 0.0010$ stačilo provést 6 iterací, přičemž $\mathbf{x}^6 = (1.9997, 1.0001, 2.0000)^T$.



M-file 2.1 lze použít pro řešení libovolného n rozměrného systému lineárních rovnic o n neznámých. Z důvodu přehlednosti práce ale budeme m-file používat pouze pro systémy $n = 5$ a menší. V příkladu 2.4 ho aplikujeme na matici $\mathbf{A} \in \mathbb{M}_4$.

Příklad 2.4. Řešte systém $\mathbf{Ax} = \mathbf{b}$ s počáteční aproximací \mathbf{x}^0 tvaru

$$\mathbf{A} = \begin{pmatrix} 445 & 56 & 0 & 15 \\ 45 & 916 & 77 & 0 \\ 111 & 66 & 1423 & 5 \\ 35 & 45 & 48 & 6615 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 976 \\ 929 \\ -1125 \\ 13297 \end{pmatrix}, \quad \mathbf{x}^0 = \begin{pmatrix} 55 \\ 42 \\ -12 \\ 125 \end{pmatrix},$$

Jacobiovou iterační metodou s přesností $\epsilon = 0.1000$.

Řešení

Konvergence tohoto systému je zaručena, neboť \mathbf{A} je ryze diagonálně dominantní. Nyní do Matlabu napíšeme vstupní parametry a zavoláme m-file následovně

```
>> A=[445 56 0 15;45 916 77 0;111 66 1423 5;35 45 48 6615];
>> b=[976;929;-1125;13297];
>> x0=[55;42;-12;125];
>> eps=0.1000;
```

```
>> [x,k]=JACOBY(A,b,x0,eps)
```

Info: Konec výpočtu, byla dosažena požadovaná přesnost.

```
x =
```

```
2.0013 0.9988 -1.0000 2.0000
```

```
k =
```

```
3
```

Přesným řešením tohoto systému je vektor $\mathbf{x}^* = (2, 1, -1, 2)^T$. Již ve třetím kroku Jacobiova iteračního procesu našel m-file dostatečně přesnou aproximaci $\mathbf{x}^3 = (2.0013, 0.9988, -1.0000, 2.0000)^T$ s danou přesností $\epsilon = 0.1000$.



2.2 Gaussova–Seidelova iterační metoda

Gaussova-Seidelova iterační metoda je modifikací Jacobiovy metody. Její uživatelé mohou znát Gaussovu-Seidelovu iterační metodu také pod názvem Metoda postupných oprav. Tento název charakterizuje tu skutečnost, že při výpočtech se každá již vypočítaná složka x_i^{k+1} vektoru \mathbf{x}^{k+1} posloupnosti $\{\mathbf{x}^k\}$ ihned použije v dalším výpočtu složky x_{i+1}^{k+1} .

Na konci této kapitoly si vysvětlíme, která z metod je pro některé typy matic vhodnější, a rozdíl mezi rychlostí jejich konvergence.

Při výpočtu složky x_i^{k+1} , $1 < i \leq n$, používá Gaussova–Seidelova metoda již vypočtené složky $x_1^{k+1}, \dots, x_{i-1}^{k+1}$ a složky x_{i+1}^k, \dots, x_n^k z předchozí iterace. Gaussova–Seidelova iterační metoda může být tedy zapsána ve tvaru

$$\begin{aligned} a_{11}x_1^{k+1} + a_{12}x_2^k + \dots + a_{1n}x_n^k &= b_1 \\ a_{21}x_1^{k+1} + a_{22}x_2^{k+1} + \dots + a_{2n}x_n^k &= b_2 \\ &\vdots \\ a_{n1}x_1^{k+1} + a_{n2}x_2^{k+1} + \dots + a_{nn}x_n^{k+1} &= b_n, \end{aligned}$$

nebo po složkách

$$x_i^{k+1} = - \sum_{j=1}^{i-1} \frac{a_{ij}}{a_{ii}} x_j^{k+1} - \sum_{j=i+1}^n \frac{a_{ij}}{a_{ii}} x_j^k + \frac{b_i}{a_{ii}}, \quad i = 1, \dots, n. \quad (15)$$

Vztah (15) se často používá v praxi.

Následující postup si uvedeme pro snadnější pochopení odvození maticového zápisu Gaussovy-Seidelovy metody tvaru $\mathbf{x} = \mathbf{T}_G \mathbf{x} + \mathbf{g}_G$. Nejdříve upravíme daný systém lineárních rovnic

$$\mathbf{A} \mathbf{x} = \mathbf{b}$$

tak, že za matici \mathbf{A} dosadíme dle vztahu (8) a rovnicí upravíme následujícím způsobem

$$\begin{aligned} (\mathbf{D} - \mathbf{L} - \mathbf{U}) \mathbf{x} &= \mathbf{b}, \\ (\mathbf{D} - \mathbf{L}) \mathbf{x} &= \mathbf{U} \mathbf{x} + \mathbf{b}, \end{aligned} \quad (16)$$

kde matice \mathbf{D} je dána vztahem (9). Matice \mathbf{L} a \mathbf{U} jsou dány vztahy (10) a (11). Za předpokladu, že $a_{ii} \neq 0, i = 1, \dots, n$, je matice $\mathbf{D} - \mathbf{L}$ regulární, maticový rozdíl je tvaru

$$\mathbf{D} - \mathbf{L} = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ a_{21} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ a_{n1} & \dots & a_{n,n-1} & a_{nn} \end{pmatrix},$$

a následnou úpravou vztahu (16) získáme vztah

$$\mathbf{x} = (\mathbf{D} - \mathbf{L})^{-1} \mathbf{U} \mathbf{x} + (\mathbf{D} - \mathbf{L})^{-1} \mathbf{b}. \quad (17)$$

Označením matice $\mathbf{T}_G = (\mathbf{D} - \mathbf{L})^{-1} \mathbf{U}$ a vektoru $\mathbf{g}_G = (\mathbf{D} - \mathbf{L})^{-1} \mathbf{b}$ získáváme předpis $\mathbf{x} = \mathbf{T}_G \mathbf{x} + \mathbf{g}_G$. Iterační vztah Gaussovy-Seidelovy metody je tvaru

$$\mathbf{x}^{k+1} = \mathbf{T}_G \mathbf{x}^k + \mathbf{g}_G.$$

Věta 2.4. *Posloupnost $\{\mathbf{x}^k\}$ generovaná Gaussovou-Seidelovou iterační metodou $\mathbf{x}^{k+1} = \mathbf{T}_G \mathbf{x}^k + \mathbf{g}_G$ konverguje pro každou počáteční aproximaci $\mathbf{x}^0 \in \mathbb{R}^n$ právě tehdy, když $\rho(\mathbf{T}_G) < 1$.*

Tuto větu jsme uváděli již dříve jako větu 2.1, zde je pouze aplikována na Gaussovu–Seidelovu iterační metodu.

Věta 2.5. *Je-li matice \mathbf{A} symetrická a pozitivně definitní, Gaussova–Seidelova iterační metoda konverguje nezávisle na volbě počáteční aproximace.*

Důkaz viz [6] str. 468.

Věta 2.6. *Pokud jsou splněny předpoklady a) nebo b) věty 2.3, pak Gaussova–Seidelova metoda konverguje pro každou počáteční aproximaci $\mathbf{x}^0 \in \mathbb{R}^n$.*

Důkaz viz [2] str. 144.

Po vyslovení těchto vět vznikají dvě otázky

1. Konverguje Jacobiova a Gaussova–Seidelova metoda i pro jiné než ryze řádkově diagonálně dominantní matice?
2. V případě konvergence obou metod, která z nich konverguje rychleji?

Z globálního hlediska jsou obory konvergence Jacobiovy a Gaussovy–Seidelovy metody různé a jen částečně se překrývají. Porovnat je můžeme jen v některých situacích. Jednu z nich nám dává např. **Steinova–Rosenbergova věta**, kterou lze najít např. v [2].

Věta 2.7. *Nechť pro prvky matice $\mathbf{A} = (a_{ij})_{ij=1}^n$, platí $a_{ij} \leq 0$ pro všechna $i \neq j$ a $a_{ii} > 0$, $i = 1, \dots, n$. Pak platí jedno z následujících tvrzení*

1. $\rho(\mathbf{T}_J) = \rho(\mathbf{T}_G) = 0$,
2. $\rho(\mathbf{T}_J) = \rho(\mathbf{T}_G) = 1$,
3. $0 < \rho(\mathbf{T}_G) < \rho(\mathbf{T}_J) < 1$,
4. $1 < \rho(\mathbf{T}_J) < \rho(\mathbf{T}_G)$.

Konverguje-li Jacobiova i Gaussova–Seidelova iterační metoda pro matice z věty 2.7, pak Gaussova–Seidelova iterační metoda konverguje rychleji, což plyne ze třetího bodu věty 2.7. Věta dále říká, že pro matice z věty 2.7 platí, pokud se spektrální poloměry iteračních matic \mathbf{T}_J a \mathbf{T}_G rovnají, pak jsou rovny nule nebo jedné. Z posledního bodu této věty plyne, pokud obě metody pro matice z této věty divergují, pak Jacobiova metoda diverguje rychleji.

V následujícím příkladě si názorně odvodíme jak obecně vznikne matice \mathbf{T}_G a vektor \mathbf{g}_G pro $n = 3$ a pak je uplatníme v příkladě 2.6.

Příklad 2.5. Vyjádřete obecné odvození Gaussovy–Seidelovy iterační matice \mathbf{T}_G a vektoru \mathbf{g}_G pro matici $\mathbf{A} \in \mathbb{M}_3$ a vektor $\mathbf{b} \in \mathbb{R}^3$.

Řešení

Matice \mathbf{T}_G je dána vztahem $\mathbf{T}_G = (\mathbf{D} - \mathbf{L})^{-1}\mathbf{U}$, přičemž matici $(\mathbf{D} - \mathbf{L})^{-1}$ vy počítáme následovně

$$\begin{aligned} (\mathbf{D} - \mathbf{L})^{-1} &= \begin{pmatrix} a_{11} & 0 & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{pmatrix}^{-1} = \\ &= \frac{1}{a_{11}a_{22}a_{33}} \begin{pmatrix} a_{22}a_{33} & -a_{21}a_{33} & a_{21}a_{32} - a_{22}a_{31} \\ 0 & a_{11}a_{33} & a_{11}a_{22} \\ 0 & 0 & a_{11} \end{pmatrix}^T = \\ &= \begin{pmatrix} \frac{a_{22}a_{33}}{a_{11}a_{22}a_{33}} & 0 & 0 \\ -\frac{a_{21}a_{33}}{a_{11}a_{22}a_{33}} & \frac{a_{11}a_{33}}{a_{11}a_{22}a_{33}} & 0 \\ \frac{a_{21}a_{32} - a_{22}a_{31}}{a_{11}a_{22}a_{33}} & -\frac{a_{11}a_{32}}{a_{11}a_{22}a_{33}} & \frac{a_{11}a_{22}}{a_{11}a_{22}a_{33}} \end{pmatrix} = \\ &= \begin{pmatrix} \frac{1}{a_{11}} & 0 & 0 \\ -\frac{a_{21}}{a_{11}a_{22}} & \frac{1}{a_{22}} & 0 \\ \frac{a_{21}a_{32} - a_{22}a_{31}}{a_{11}a_{22}a_{33}} & -\frac{a_{32}}{a_{22}a_{33}} & \frac{1}{a_{33}} \end{pmatrix}. \end{aligned}$$

Matice \mathbf{U} je horní trojúhelníková tvaru

$$\mathbf{U} = \begin{pmatrix} 0 & -a_{12} & -a_{13} \\ 0 & 0 & -a_{23} \\ 0 & 0 & 0 \end{pmatrix}.$$

Matici \mathbf{T}_G vypočítáme maticovým násobením matic $(\mathbf{D} - \mathbf{L})^{-1}$ a \mathbf{U}

$$\begin{aligned} \mathbf{T}_G &= \begin{pmatrix} \frac{1}{a_{11}} & 0 & 0 \\ -\frac{a_{21}}{a_{11}a_{22}} & \frac{1}{a_{22}} & 0 \\ \frac{a_{21}a_{32}-a_{22}a_{31}}{a_{11}a_{22}a_{33}} & -\frac{a_{32}}{a_{22}a_{33}} & \frac{1}{a_{33}} \end{pmatrix} \begin{pmatrix} 0 & -a_{12} & -a_{13} \\ 0 & 0 & -a_{23} \\ 0 & 0 & 0 \end{pmatrix} = \\ &= \begin{pmatrix} 0 & -\frac{a_{12}}{a_{11}} & -\frac{a_{13}}{a_{11}} \\ 0 & \frac{a_{12}a_{21}}{a_{11}a_{22}} & \frac{a_{13}a_{21}-a_{11}a_{23}}{a_{11}a_{22}} \\ 0 & \frac{a_{12}a_{22}a_{31}-a_{12}a_{21}a_{32}}{a_{11}a_{22}a_{33}} & \frac{a_{11}a_{23}a_{32}+a_{13}a_{22}a_{31}-a_{13}a_{21}a_{32}}{a_{11}a_{22}a_{33}} \end{pmatrix}. \end{aligned} \quad (18)$$

Pro výpočet čísla k , které vyjadřuje počet iterací nutných k dosažení aproximace \mathbf{x}^k s požadovanou přesností ϵ , pomocí vztahu (6) potřebujeme vypočítat i normu vektoru \mathbf{g}_G . Proto je nutné odvodit i vektor \mathbf{g}_G . Platí $\mathbf{g}_G = (\mathbf{D} - \mathbf{L})^{-1}\mathbf{b}$, pak \mathbf{g}_G vypočítáme

$$\begin{aligned} \mathbf{g}_G &= \begin{pmatrix} \frac{1}{a_{11}} & 0 & 0 \\ -\frac{a_{21}}{a_{11}a_{22}} & \frac{1}{a_{22}} & 0 \\ \frac{a_{21}a_{32}-a_{22}a_{31}}{a_{11}a_{22}a_{33}} & -\frac{a_{32}}{a_{22}a_{33}} & \frac{1}{a_{33}} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \\ &= \begin{pmatrix} \frac{b_1}{a_{11}} \\ -\frac{a_{21}b_1-a_{11}b_2}{a_{11}a_{22}} \\ \frac{a_{21}a_{32}b_1-a_{22}a_{31}b_1-a_{11}a_{32}b_2+a_{11}a_{22}b_3}{a_{11}a_{22}a_{33}} \end{pmatrix}. \end{aligned} \quad (19)$$



Nyní máme odvozenou matici \mathbf{T}_G a vektor \mathbf{g}_G , ale pouze pro systémy stupně $n = 3$. V případě větších či menších lineárních soustav musíme použít pro výpočet \mathbf{T}_G a \mathbf{g}_G odpovídající části vztahu (17), nebo m-file uvedený na konci této podkapitoly. V následujícím příkladu použijeme vztahy (18) a (19) pro matici \mathbf{T}_G a vektor \mathbf{g}_G .

Příklad 2.6. Řešte soustavu lineárních rovnic

$$\begin{aligned} 5x_1 + x_2 + 2x_3 &= 4 \\ -2x_1 + 4x_2 + x_3 &= -3 \\ x_1 + x_2 + 4x_3 &= 2 \end{aligned}$$

Gaussovou–Seidelovou iterační metodou a určete počet iterací k nutných k dosažení dostatečného přiblížení poslední provedené aproximace \mathbf{x}^k k přesnému řešení s přesností $\epsilon = 10^{-4}$.

Řešení

Gaussova–Seidelova metoda bude konvergovat pro každou počáteční aproximaci, protože matice tohoto systému je ryze řádkově diagonálně dominantní. Přesným řešením je vektor $\mathbf{x}^* = \left(\frac{13}{18}, -\frac{1}{2}, \frac{4}{9}\right)^T = (0.7220, -0.5000, 0.4400)^T$. Tuto soustavu lineárních rovnic můžeme přepsat do maticového tvaru

$$\begin{pmatrix} 5 & 1 & 2 \\ -2 & 4 & 1 \\ 1 & 1 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 4 \\ -3 \\ 2 \end{pmatrix}.$$

Gaussova–Seidelova iterační metoda lze po složkách vyjádřit takto

$$\begin{aligned} x_1^{k+1} &= \frac{1}{5} (4 - x_2^k - 2x_3^k), \\ x_2^{k+1} &= \frac{1}{4} (-3 + 2x_1^{k+1} - x_3^k), \\ x_3^{k+1} &= \frac{1}{4} (2 - x_1^{k+1} - x_2^{k+1}). \end{aligned}$$

Protože je matice $\mathbf{T}_{\mathbf{G}}$ stupně $n = 3$, můžeme dosadit do vztahu (18) a vypočítáme matici

$$\mathbf{T}_{\mathbf{G}} = \begin{pmatrix} 0 & -\frac{1}{5} & -\frac{2}{5} \\ 0 & -\frac{1}{10} & -\frac{9}{20} \\ 0 & \frac{3}{40} & \frac{17}{80} \end{pmatrix}.$$

Pokud budeme znát hodnotu norem matice $\mathbf{T}_{\mathbf{G}}$ a vektoru $\mathbf{g}_{\mathbf{G}}$, budeme moci vypočítat pomocí vztahu (6) počet iterací k potřebný k nalezení aproximace \mathbf{x}^k s přesností ϵ . Řádková norma matice $\mathbf{T}_{\mathbf{G}}$ je $\|\mathbf{T}_{\mathbf{G}}\|_{\infty} = \max\{\frac{3}{5}, \frac{11}{20}, \frac{23}{80}\} = \frac{3}{5}$. Norma matice $\mathbf{T}_{\mathbf{G}}$ je tedy menší než jedna, což znovu potvrzuje konvergenci posloupnosti $\{\mathbf{x}^k\}$. Pro stanovení velikosti normy vektoru $\mathbf{g}_{\mathbf{G}}$ musíme nejdříve

tento vektor vypočítat dle vztahu (19)

$$\mathbf{g}_G = \begin{pmatrix} \frac{4}{5} \\ -\frac{7}{20} \\ \frac{31}{80} \end{pmatrix}.$$

Norma vektoru \mathbf{g}_G je $\|\mathbf{g}_G\|_\infty = \max\{\frac{4}{5}, \frac{7}{20}, \frac{31}{80}\} = \frac{4}{5}$. Položme počáteční aproximaci $\mathbf{x}^0 = \mathbf{g}_G = (\frac{4}{5}, -\frac{7}{20}, \frac{31}{80})^T$. Po provedení stejných úvah jako v příkladu 2.2 dostáváme ze vztahu (6) tuto nerovnici

$$\|\mathbf{x}^* - \mathbf{x}^k\| \leq \frac{\|\mathbf{T}\|^{k+1}}{1 - \|\mathbf{T}\|} \|\mathbf{g}\| < \epsilon.$$

Do ní dosadíme a provádíme výpočet

$$\frac{(\frac{3}{5})^{k+1} 4}{1 - \frac{3}{5} 5} < 10^{-3}.$$

Tuto nerovnici zlogaritmujeme a provedeme úpravy, abychom vyjádřili hledanou hodnotu k

$$(k+1) \log\left(\frac{3}{5}\right) < \log 10^{-3} - \log 2,$$

$$k > \frac{-3 - \log 2}{\log(\frac{3}{5})} - 1,$$

$$k > 13.8796.$$

Pro nalezení aproximace \mathbf{x}^k přesného řešení \mathbf{x}^* s přesností menší než 10^{-3} je nutné provést alespoň $k = 14$ iterací, protože $k \in \mathbb{N}$. Abychom si předvedli, jakým postupem se provádí výpočet, uvedeme si první 2 iterace.

Nalezení aproximace \mathbf{x}^k Gaussovy-Seidelovy iterační metody s přesností ϵ můžeme provést užitím následujícího m-filu naprogramovaného v Matlabu. Ručně je počítat nebudeme z důvodu zdlouhavosti a obtížnosti výpočtů a uvedeme pouze první 2 iterace. Jako počáteční aproximaci označíme již dříve užívaný vektor

$\mathbf{x}^0 = \mathbf{g}_G = (0.8000, -0.3500, 0.3875)^T$. První iteraci vypočítáme

$$x_1^1 = \frac{1}{5} (4 + 0.3500 - 2 \cdot 0.3875) = 0.7150,$$

$$x_2^1 = \frac{1}{4} (-3 + 2 \cdot 0.7150 - 0.3875) = -0.4894,$$

$$x_3^1 = \frac{1}{4} (2 - 0.7150 + 0.4894) = 0.4436.$$

Její hodnota je $\mathbf{x}^1 = (0.7150, -0.4894, 0.4436)^T$. Pro výpočet první složky druhé iterace použijeme pouze \mathbf{x}^1 . Složku x_2^2 vypočítáme pomocí x_1^1 a x_3^1 a poslední složku x_3^2 získáme užitím x_1^1 a x_2^1 , tj.

$$x_1^2 = \frac{1}{5} (4 + 0.4894 - 2 \cdot 0.4436) = 0.7204,$$

$$x_2^2 = \frac{1}{4} (-3 + 2 \cdot 0.7204 - 0.4436) = -0.6808,$$

$$x_3^2 = \frac{1}{4} (2 - 0.7204 + 0.6808) = 0.4901.$$

Druhá iterace je rovna $\mathbf{x}^2 = (0.7204, -0.6808, 0.4901)^T$.

Provedené výpočty prokázaly, že se obě iterace pohybují blízko přesného řešení a pokud bychom provedli další iterace, pak bychom se pravděpodobně přiblížili k přesnému řešení s požadovanou přesností ϵ .



M-file 2.2 jsem naprogramovala sama v matematickém softwaru Matlab. Tento m-file řeší soustavy n lineárních rovnic o n neznámých Gaussovou–Seidelovou metodou. Má stejné vlastnosti jako předchozí m-file, tj. ověřuje rozměry vstupních parametrů, také pokud bude vstupní parametr počáteční aproximace prázdný, zvolí za počáteční aproximaci \mathbf{x}^0 vektor \mathbf{g}_G . I zde je použito zastavovací kritérium z prvního bodu poznámky 2.2. Maximální počet provedených iterací bude 10000. Tato hodnota byla vybrána z důvodu zabránění případnému zacyklení m-filu. V případě, že bude soustava lineárních rovnic konvergovat, pak program vypíše

počet provedených iterací k se zadanou přesností ϵ a poslední provedenou iterací. Jinak vypíše hlášení o divergenci metody. Počet iterací k v m-filu 2.2 bude ze stejného důvodu jako v m-file 2.1 odlišný od počtu iterací k v příkladu 2.6. M-file je naprogramován tak, abychom mohli volit různé počáteční aproximace a odchylky ϵ .

M-file 2.2.

```
function [x,k]=GAUSSOVA_SEIDELOVA(A,b,x0,eps)
% Gaussova-Seidelova metoda pro řešení systému lineárních rovnic
% vstupní data:  A   matice tvořená koeficienty soustavy rovnic
%               b   vektor pravých stran dané soustavy rovnic
%               x0  počáteční aproximace
%               eps požadovaná přesnost
% výstupní data: x   hledaná iterace
%               k   počet provedených iterací
% zjištění rozměru vstupních hodnot
n=size(A,1);
n1=size(b,1);
n2=size(x0,1);
% maximální počet provedených iterací
K=10000;
% ověření rozměru A a b, kladnosti epsilon
if (n==n1 && eps>0)
    % mezivýpočty matic D, L a U
    a=diag(A);
    D=diag(a);
    L=(tril(A)-D)*(-1);
    U=(triu(A)-D)*(-1);
    X=D-L;
    % výpočet matice T a vektoru g
    T=X\U; % \ počítá inverzní matici
    g=X\b;
    spekt=max(abs(eig(T)));
```

```

if (spekt>=1)
    disp('Info: Zvolte jinou metodu.');
```

error('Chyba: Posloupnost je divergentní!');

```

end
% ověření neprázdnosti vektoru x0
if (isempty(x0)==1)
    x=g; % jestliže je prázdný, načte vektor g
else
    x=x0; % jinak načte zadaný vektor
    if (n~=n2)
        error('Chyba: Zkontrolujte zda odpovídá rozměr počáteční
            aproximace x0 a vektoru b!')
```

end

```

end
% výpočet iterací, maximální počet iterací K
for k1=0:K
    z0=x;
    for i=1:n
        w=0; % umožňuje sčítání násobených prvků při iteracích
        for j=1:n
            w=w+T(i,j)*x(j);
            if j==n
                x(i)=w+g(i);
            end
        end
    end
    end
    % přepsání jednoho prvku v x a naplnění vektoru z
    z1=x;
    odchylka=norm(z1-z0)/norm(z0);
    if (odchylka<eps)

```



```

        disp('Info:Konec výpočtu, byla dosažena požadovaná
              přesnost. ');
        k=k1+1;
        break;
    end
end
else
    error('Chyba:Zkontrolujte zda odpovídá stupeň matice A rozměru
          vektoru b, nebo zda je kladné epsilon!')
end
end
end

```



M-file 2.2 použijeme pro získání hledané aproximace \mathbf{x}^k s přesností $\epsilon = 10^{-4}$ z příkladu 2.6. Počáteční aproximace \mathbf{x}^0 je prázdná a m-file ji tedy položí rovnu vektoru \mathbf{g} .

Příklad 2.7. Do Matlabu vložíme vstupní data a zavoláme m-file následovně.

Řešení

```

>> A=[5 1 2;-2 4 1;1 1 4];
>> b=[4;-3;2];
>> x0=[ ];
>> eps=0.0001;
>> [x,k]=GAUSSOVA_SEIDELOVA(A,b,x0,eps)
Info: Konec výpočtu, byla dosažena požadovaná přesnost.
x =
    0.7222 -0.5000  0.4444
k =
    6

```

K nalezení aproximace s přesností $\epsilon = 0.0001$ m-file provedl šest iterací. Poslední je tvaru $\mathbf{x}^6 = (0.7222, -0.5000, 0.4444)^T$.



V příkladě 2.8 uijeme oba zatím uvedené m-fily a ukážeme si, že i pokud Jacobiova metoda diverguje může Gaussova–Seidelova metoda konvergovat. Samozřejmě tomu může být i naopak, tj. Jacobiova metoda bude konvergovat a Gaussova–Seidelova metoda divergovat.

Příklad 2.8. Řešte systém $\mathbf{Ax} = \mathbf{b}$ se zadanou počáteční aproximací \mathbf{x}^0

$$\mathbf{A} = \begin{pmatrix} 5 & 1 & -1 & 1 \\ 25 & 43 & 0 & 0 \\ 82 & 99 & 16 & 1 \\ 22 & 42 & 3 & 68 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 3 \\ -18 \\ -18 \\ -88 \end{pmatrix}, \quad \mathbf{x}^0 = \begin{pmatrix} 5 \\ 19 \\ 48 \\ 1 \end{pmatrix},$$

Jacobiovou a Gaussovou–Seidelovou iterační metodou s přesností $\epsilon = 10^{-3}$.

Řešení

Do Matlabu zadáme známé hodnoty a zavoláme oba m-fily.

```
>> A=[5 1 -1 1;25 43 0 0;82 99 16 1;22 42 3 68];
```

```
>> b=[3;-18;-18;-88];
```

```
>> x0=[5;19;48;1];
```

```
>> eps=0.0010;
```

```
>> [x,k]=JACOBY(A,b,x0,eps)
```

```
Info: Zvolte jinou metodu.
```

```
??? Error using ==> JACOBY at 39
```

```
Chyba: Posloupnost je divergující!
```

```
>> [x,k]=GAUSSOVA_SEIDELOVA(A,b,x0,eps)
```

```
Info: Konec výpočtu, byla dosažena požadovaná přesnost.
```

```
x =
```

```
1.0001 -1.0001 -0.0002 -1.0000
```

$\mathbf{k} =$

11

Přesným řešením je $\mathbf{x}^* = (1, -1, 0, -1)^T$. Jacobiova metoda diverguje, ale Gaussova–Seidelova metoda konverguje. K výpočtu dostatečně přesné aproximace bylo nutné provést 11 iterací.



Více příkladů, ve kterých budeme aplikovat m-fily a sledovat jejich výsledky, si uvedeme v poslední kapitole.

2.3 Relaxační metoda

Poslední uvedenou iterační metodou bude **Relaxační metoda**. Jde o modifikaci Gaussovy-Seidelovy metody a představuje obohacení iteračního vzorce o parametr ω . Základní hypotézou pro Relaxační metodu je existence třídy matic \mathbf{T}_ω závisících na parametru ω , kde $\omega > 0$. Pro tyto matice \mathbf{T}_ω má odpovídající iterační proces rychlejší konvergenci než Gaussova-Seidelova metoda. Tento parametr ω se nazývá **relaxační**. Následně se budeme pokoušet vybrat parametr ω tak, aby spektrální poloměr $\rho(\mathbf{T}_\omega)$ byl co nejmenším číslem. Čím více se spektrální poloměr $\rho(\mathbf{T}_\omega)$ blíží nule, tím rychleji konverguje relaxační metoda. Budeme se tedy konkrétně zabývat třídou matic tvaru

$$\mathbf{T}_\omega = (\mathbf{D} - \omega\mathbf{L})^{-1}[(1 - \omega)\mathbf{D} + \omega\mathbf{U}]. \quad (20)$$

Pro $\omega > 0$ mohou nastat tyto situace

$0 < \omega < 1$ iterační metody se nazývají **metodami dolní relaxace**,

$\omega = 1$ Relaxační metoda totožná s Gaussovou-Seidelovou metodou,

$1 < \omega$ iterační metody se nazývají **metodami horní relaxace**, nebo **SOR**.

Parametr ω v prvním případě se užívá pokud Gaussova-Seidelova metoda nekonverguje. Parametr $1 < \omega$ se aplikuje pro zrychlení konvergence Gaussovy-Seidelovy metody.

Maticový zápis Relaxační metody je

$$\mathbf{x}^{k+1} = (\mathbf{D} - \omega\mathbf{L})^{-1}[(1 - \omega)\mathbf{D} + \omega\mathbf{U}]\mathbf{x}^k + \omega(\mathbf{D} - \omega\mathbf{L})^{-1}\mathbf{b}. \quad (21)$$

Vztah (21) můžeme přepsat při označení matice \mathbf{T}_ω vztahem (20) a vektoru $\mathbf{g}_\omega = \omega(\mathbf{D} - \omega\mathbf{L})^{-1}\mathbf{b}$ na rovnici

$$\mathbf{x}^{k+1} = \mathbf{T}_\omega\mathbf{x}^k + \mathbf{g}_\omega. \quad (22)$$

Po jednotlivých složkách lze Relaxační metodu zapsat

$$x_i^{k+1} = (1 - \omega)x_i^k + \frac{\omega}{a_{ii}} \left[b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i+1}^n a_{ij}x_j^k \right]. \quad (23)$$

Vztah (22) lze využít pro jakkoliv rozsáhlý systém, my si v následujícím příkladě uvedeme názorné odvození matice \mathbf{T}_ω a vektoru \mathbf{g}_ω pro systém tří lineárních rovnic o třech neznámých. Pro rozsáhlejší nebo menší systémy můžeme použít vztah (23) a nebo m-file uvedený na konci této podkapitoly.

Příklad 2.9. Uveďte obecné odvození tvaru matice \mathbf{T}_ω a vektoru \mathbf{g}_ω ze vztahu (22) pro systém $\mathbf{Ax} = \mathbf{b}$, kde matice $\mathbf{A} \in \mathbb{M}_3$ a vektor $\mathbf{b} \in \mathbb{R}^3$.

Řešení

Ze vztahu $\mathbf{T}_\omega = (\mathbf{D} - \omega\mathbf{L})^{-1}[(1 - \omega)\mathbf{D} + \omega\mathbf{U}]$ pro iterační matici této metody odvodíme nejdříve inverzi rozdílů matic $(\mathbf{D} - \omega\mathbf{L})^{-1}$

$$\begin{aligned} (\mathbf{D} - \omega\mathbf{L})^{-1} &= \begin{pmatrix} a_{11} & 0 & 0 \\ a_{21}\omega & a_{22} & 0 \\ a_{31}\omega & a_{32}\omega & a_{33} \end{pmatrix}^{-1} = \\ &= \frac{1}{a_{11}a_{22}a_{33}} \begin{pmatrix} a_{22}a_{33} & -a_{21}a_{33}\omega & a_{21}a_{32}\omega^2 - a_{22}a_{31}\omega \\ 0 & a_{11}a_{33} & -a_{11}a_{32}\omega \\ 0 & 0 & a_{11}a_{22} \end{pmatrix}^T = \end{aligned}$$

$$= \begin{pmatrix} \frac{a_{22}a_{33}}{a_{11}a_{22}a_{33}} & 0 & 0 \\ -\frac{a_{21}a_{33}\omega}{a_{11}a_{22}a_{33}} & \frac{a_{11}a_{33}}{a_{11}a_{22}a_{33}} & 0 \\ \frac{a_{21}a_{32}\omega^2 - a_{22}a_{31}\omega}{a_{11}a_{22}a_{33}} & -\frac{a_{11}a_{32}\omega}{a_{11}a_{22}a_{33}} & \frac{a_{11}a_{22}}{a_{11}a_{22}a_{33}} \end{pmatrix} \quad (24)$$

a následně také

$$[(1-\omega)\mathbf{D} + \omega\mathbf{U}] = \begin{pmatrix} (1-\omega)a_{11} & -a_{12}\omega & -a_{13}\omega \\ 0 & (1-\omega)a_{22} & -a_{23}\omega \\ 0 & 0 & (1-\omega)a_{33} \end{pmatrix}. \quad (25)$$

Maticovým násobením matic (24) a (25) vypočítáme matici \mathbf{T}_ω takto

$$\begin{aligned} \mathbf{T}_\omega &= \begin{pmatrix} \frac{a_{22}a_{33}}{a_{11}a_{22}a_{33}} & 0 & 0 \\ -\frac{a_{21}a_{33}\omega}{a_{11}a_{22}a_{33}} & \frac{a_{11}a_{33}}{a_{11}a_{22}a_{33}} & 0 \\ \frac{a_{21}a_{32}\omega^2 - a_{22}a_{31}\omega}{a_{11}a_{22}a_{33}} & -\frac{a_{11}a_{32}\omega}{a_{11}a_{22}a_{33}} & \frac{a_{11}a_{22}}{a_{11}a_{22}a_{33}} \end{pmatrix} \times \\ &\times \begin{pmatrix} (1-\omega)a_{11} & -a_{12}\omega & -a_{13}\omega \\ 0 & (1-\omega)a_{22} & -a_{23}\omega \\ 0 & 0 & (1-\omega)a_{33} \end{pmatrix} = \\ &= \begin{pmatrix} (1-\omega) & -\frac{a_{21}\omega(1-\omega)}{a_{22}} & \frac{a_{21}a_{32}\omega^2(1-\omega) - a_{22}a_{31}\omega(1-\omega)}{a_{22}a_{33}} \\ -\frac{a_{12}\omega}{a_{11}} & \frac{a_{12}a_{21}\omega^2 + a_{11}a_{22}(1-\omega)}{a_{11}a_{22}} & \frac{a_{12}a_{22}a_{31}\omega^2 - a_{12}a_{21}a_{32}\omega^3 - a_{11}a_{22}a_{32}\omega(1-\omega)}{a_{11}a_{22}a_{33}} \\ -\frac{a_{13}\omega}{a_{11}} & \frac{a_{13}a_{21}\omega^2 - a_{11}a_{23}\omega}{a_{11}a_{22}} & \frac{a_{13}\omega^2(a_{22}a_{31} - a_{21}a_{32}\omega) + a_{11}(a_{23}a_{32}\omega^2 + a_{22}a_{33}(1-\omega))}{a_{11}a_{22}a_{33}} \end{pmatrix}^T. \end{aligned}$$

Pro výpočet vektoru \mathbf{g}_ω musíme vynásobit $\omega(\mathbf{D} - \omega\mathbf{L})^{-1}$ s vektorem $(b_1, b_2, b_3)^T$

$$\begin{aligned} \mathbf{g}_\omega &= \omega \begin{pmatrix} \frac{a_{22}a_{33}}{a_{11}a_{22}a_{33}} & 0 & 0 \\ -\frac{a_{21}a_{33}\omega}{a_{11}a_{22}a_{33}} & \frac{a_{11}a_{33}}{a_{11}a_{22}a_{33}} & 0 \\ \frac{a_{21}a_{32}\omega^2 - a_{22}a_{31}\omega}{a_{11}a_{22}a_{33}} & -\frac{a_{11}a_{32}\omega}{a_{11}a_{22}a_{33}} & \frac{a_{11}a_{22}}{a_{11}a_{22}a_{33}} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \\ &= \begin{pmatrix} \frac{b_1\omega}{a_{11}} \\ -\frac{a_{21}b_1\omega^2 + a_{11}b_2\omega}{a_{11}a_{22}} \\ \frac{a_{21}a_{32}b_1\omega^3 - a_{22}a_{31}b_1\omega^2 - a_{11}a_{32}b_2\omega^2 + a_{11}a_{22}b_3\omega}{a_{11}a_{22}a_{33}} \end{pmatrix}. \quad (26) \end{aligned}$$

♠

Nyní má smysl zabývat se dvěma otázkami týkajícími se konvergence posloupnosti $\{\mathbf{x}^k\}$ generované vztahem (22).

1. Pro které hodnoty parametru ω bude posloupnost $\{\mathbf{x}^k\}$ konvergovat?
2. Jaká je optimální volba parametru ω , k tomu aby byla konvergence posloupnosti $\{\mathbf{x}^k\}$ co nejrychlejší?

Odpověď na první otázku nám dává následující věta.

Věta 2.8. *Nechť $a_{ii} \neq 0$, $i = 1, \dots, n$. Pak*

$$\rho(\mathbf{T}_\omega) \geq |\omega - 1|, \quad (27)$$

kde \mathbf{T}_ω je tvaru (20).

Důkaz viz [2] str. 148.

Poznámka 2.3. Z věty 2.8 vyplývá, jaký interval hodnot ω můžeme uvažovat. Pro spektrální poloměr matice $\rho(\mathbf{T}_\omega)$ musí platit vztah $\rho(\mathbf{T}_\omega) < 1$ z věty 2.1, aby posloupnost \mathbf{x}^k konvergovala k přesnému řešení \mathbf{x}^* . Víme také, že pro spektrální poloměr $\rho(\mathbf{T}_\omega)$ platí vztah (27). Spojením těchto dvou vztahů vzniká nerovnice $|\omega - 1| \leq \rho(\mathbf{T}_\omega) < 1$. Z ní vyplývá, že pokud má posloupnost \mathbf{x}^k konvergovat, pak pro relaxační parametr ω musí platit $|\omega - 1| < 1$. Hodnoty ω , které vyhovují této nerovnici, jsou $\omega \in (0, 2)$. Z hlediska konvergence budeme tedy uvažovat relaxační parametr ω pouze v intervalu $(0, 2)$.

Postačující podmínky pro konvergenci relaxační metody udávají věty 2.9 a 2.10. Obě následující věty jsou citací z [2] str. 149 a str. 151.

Věta 2.9. *Pro pozitivně definitní matici \mathbf{A} platí $\rho(\mathbf{T}_\omega) < 1$ pro $\forall \omega \in (0, 2)$.*

Relaxační parametr pro matice z věty 2.10 je optimální z hlediska nejrychlejší konvergence posloupnosti $\{\mathbf{x}^k\}$.

Věta 2.10. *Nechť matice \mathbf{A} je třídiagonální pozitivně definitní. Pak pro spektrální poloměry platí $\rho(\mathbf{T}_\mathbf{G}) = \rho^2(\mathbf{T}_\mathbf{J}) < 1$ a optimální hodnota relaxačního parametru je*

$$\omega = \omega_{opt} = \frac{2}{1 + \sqrt{1 - \rho^2(\mathbf{T}_\mathbf{J})}}. \quad (28)$$

Při splnění těchto předpokladů je $\rho(\mathbf{T}_\omega) = |1 - \omega|$.

Příklad 2.10. Řešte soustavu lineárních rovnic

$$3x_1 + x_2 = 9$$

$$2x_1 + 3x_2 + x_3 = 14$$

$$2x_2 + 3x_3 = 9$$

Relaxační metodou. Dále vypočítejte počet iterací k nutných k nalezení dostatečného přiblížení poslední provedené aproximace \mathbf{x}^k k přesnému řešení \mathbf{x}^* s přesností $\epsilon = 10^{-5}$ pomocí Relaxační metody.

Řešení

Přesným řešením této soustavy je $\mathbf{x}^* = (2, 3, 1)^T$. Matice tohoto systému splňuje předpoklady věty 2.10, protože je třídiagonální a pozitivně definitní. První vlastnost poznáme již z koeficientů soustavy lineárních rovnic. Druhou si dokážeme podle Sylvestrova pravidla. Pokud tedy pro matici

$$\mathbf{A} = \begin{pmatrix} 3 & 1 & 0 \\ 2 & 3 & 1 \\ 0 & 2 & 3 \end{pmatrix},$$

platí

$$\det(\mathbf{A}_1) = |3| = 3 > 0,$$

$$\det(\mathbf{A}_2) = \begin{vmatrix} 3 & 1 \\ 2 & 3 \end{vmatrix} = 7 > 0, \quad \det(\mathbf{A}_3) = \begin{vmatrix} 3 & 1 & 0 \\ 2 & 3 & 1 \\ 0 & 2 & 3 \end{vmatrix} = 15 > 0,$$

pak dle Sylvestrova pravidla je matice \mathbf{A} pozitivně definitní. Pro spektrální poloměry iteračních matic \mathbf{T}_J a \mathbf{T}_G platí $\rho(\mathbf{T}_G) = (\rho(\mathbf{T}_J))^2 < 1$.

Optimální hodnota ω je

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - (\rho(\mathbf{T}_J))^2}}.$$

Spektrální poloměr iterační matice $\mathbf{T}_{\omega_{opt}}$ pro optimální hodnotu ω vypočítáme vztahem

$$\rho(\mathbf{T}_{\omega_{opt}}) = |1 - \omega_{opt}|. \quad (29)$$

Pro určení hodnoty spektrálního poloměru $\rho(\mathbf{T}_{\omega_{opt}})$ potřebujeme nejdříve znát spektrální poloměr Jacobiovy metody. Ten určíme pomocí vlastních čísel z Jacobiovy iterační matice

$$\mathbf{T}_{\mathbf{J}} = \begin{pmatrix} 0 & -\frac{1}{3} & 0 \\ -\frac{2}{3} & 0 & -\frac{1}{3} \\ 0 & -\frac{2}{3} & 0 \end{pmatrix}.$$

Vlastní čísla této matice jsou $\lambda_1 = -\frac{2}{3}$, $\lambda_2 = 0$, $\lambda_3 = \frac{2}{3}$. Spektrální poloměry jsou $\rho(\mathbf{T}_{\mathbf{J}}) = \frac{2}{3}$ a $\rho(\mathbf{T}_{\mathbf{G}}) = \frac{4}{9}$. Optimální hodnotou relaxačního parametru tedy je

$$\omega_{opt} = \frac{2}{1 + \sqrt{1 - \frac{4}{9}}} = 1.0550.$$

Po dosažení této hodnoty do vztahu (29), bude hodnota spektrálního poloměru $\rho(\mathbf{T}_{\omega_{opt}}) = |1 - 1.0550| = 0.0550$. Užitím věty 2.10 jsme tedy dokázali, že posloupnost \mathbf{x}^k bude konvergovat k přesnému řešení.

Pro relaxační metodu ze vztahu (23) platí

$$\begin{aligned} x_1^{k+1} &= -0.0550x_1^k + 0.3517(9 - x_2^k), \\ x_2^{k+1} &= -0.0550x_2^k + 0.3517(14 - 2x_1^{k+1} - x_3^k), \\ x_3^{k+1} &= -0.0550x_3^k + 0.3517(9 - 2x_2^{k+1}). \end{aligned}$$

Abychom mohli následně vypočítat počet iterací k nutných pro dosažení aproximace řešení soustavy lineárních rovnic pomocí Relaxační metody s přesností ϵ , musíme nyní získat matici \mathbf{T}_{ω} , vektor \mathbf{g}_{ω} a z nich vypočítat jejich normu matice a vektoru. Iterační matici \mathbf{T}_{ω} získáme z odvozeného vztahu v příkladu 2.9, protože její stupeň je $n = 3$

$$\mathbf{T}_{\omega} = \begin{pmatrix} -\frac{11}{200} & -\frac{3517}{10000} & 0 \\ \frac{387}{10000} & \frac{1923}{10000} & -\frac{3517}{10000} \\ -\frac{17}{625} & -\frac{1353}{10000} & \frac{1923}{10000} \end{pmatrix}.$$

Norma matice \mathbf{T}_{ω} se rovná $\|\mathbf{T}_{\omega}\|_{\infty} = \max\{0.4067, 0.5827, 0.3548\} = 0.5827$.

Vektor \mathbf{g}_ω Relaxační metody ze vztahu (26) je

$$\mathbf{g}_\omega = \begin{pmatrix} 3 \\ \frac{25567}{10000} \\ \frac{6009}{5000} \end{pmatrix}.$$

Normou vektoru \mathbf{g}_ω je $\|\mathbf{g}_\omega\|_\infty = \max\{3, 2.5567, 1.2018\} = 3$. Za počáteční aproximaci dosadíme tedy $\mathbf{x}^0 = \mathbf{g}_\omega = (3, 2.5567, 1.2018)^T$, abychom vyhověli podmínkám i pro následné uplatnění ve vztahu (6), a dostáváme

$$\begin{aligned} x_1^1 &= -0.0550 \cdot 3 + 0.3517 \cdot (9 - 2.5567), \\ x_2^1 &= -0.0550 \cdot 2.5567 + 0.3517 \cdot (14 - 2 \cdot 2.1011 - 1.2018), \\ x_3^1 &= -0.0550 \cdot 1.2018 + 0.3517 \cdot (9 - 2 \cdot 2.8826). \end{aligned}$$

Hodnoty první iterace jsou $\mathbf{x}^1 = (2.1011, 2.8826, 1.0716)^T$. Druhou iteraci vypočítáme následovně

$$\begin{aligned} x_1^2 &= -0.0550 \cdot 2.1011 + 0.3517 \cdot (9 - 2.8826), \\ x_2^2 &= -0.0550 \cdot 2.8826 + 0.3517 \cdot (14 - 2 \cdot 2.0359 - 1.0716), \\ x_3^2 &= -0.0550 \cdot 1.0716 + 0.3517 \cdot (9 - 2 \cdot 2.9563). \end{aligned}$$

Vektoru druhé iterace odpovídají hodnoty $\mathbf{x}^2 = (2.0359, 2.9563, 1.0269)^T$. Pro názornost jsme uvedli pouze tyto dvě iterace a následující můžeme získat přes m-file uvedený za tímto příkladem.

Po provedení obdobných úvah jako v příkladu 2.2 dostáváme ze vztahu (6) tuto nerovnici

$$\|\mathbf{x}^* - \mathbf{x}^k\| \leq \frac{\|\mathbf{T}_\omega\|^{k+1}}{1 - \|\mathbf{T}_\omega\|} \|\mathbf{g}_\omega\| < \epsilon,$$

do ní dosadíme odpovídající hodnoty a vzniká následující nerovnice

$$\frac{\left(\frac{5827}{10000}\right)^{k+1}}{1 - \frac{5827}{10000}} 3 < 10^{-5}.$$

Provedeme potřebné úpravy, abychom mohli vyjádřit hodnotu k , a získáme vztah

$$(k + 1) \log \left(\frac{5827}{10000} \right) < \log(10^{-5}) - \log \frac{71889}{10000}$$

$$k > \frac{-5 - \log \frac{71889}{10000}}{\log \left(\frac{5827}{10000} \right)} - 1.$$

Zjistili jsme tedy že $k > 23.9693$ a je tedy potřebné provést alespoň $k = 24$ iterací, abychom dosáhli dostatečného přiblížení poslední provedené aproximace \mathbf{x}^k k přesnému řešení s přesností ϵ .



I poslední uvedený m-file 2.3 jsem naprogramovala sama v Matlabu. Tento m-file řeší soustavy n lineárních rovnic o n neznámých Relaxační metodou s parametrem ω , který musí náležet intervalu $(0, 2)$. M-file 2.3 má obdobné vlastnosti jako předchozí dva m-fily, tj. ověřuje rozměry vstupních parametrů, v případě prázdnosti počáteční aproximace, platí také $\mathbf{x}^0 = \mathbf{g}_\omega$. Je v něm znovu aplikováno zastavovací kritérium z prvního bodu poznámky 2.2 a maximální počet iterací je 10000. Také platí, že program vypíše počet provedených iterací k se zadanou přesností ϵ a poslední realizovanou iteraci, pokud bude soustava lineárních rovnic konvergovat. Jinak zastaví výpočet a vypíše hlášení o divergenci metody. I v m-filu 2.3 je počet iterací k odlišný od počtu iterací k v příkladech počítaných ručně. Je tomu tak z důvodu použití jiného zastavovacího kritéria. V m-filu 2.3 můžeme zadávat různé počáteční aproximace, relaxační parametry a odchylky ϵ .

M-file 2.3.

```
function [x,k]=RELAXACNI(A,b,x0,w,eps)
% Relaxační metoda pro řešení systému lineárních rovnic
% vstupní data:  A   matice tvořená koeficienty soustavy rovnic
%               b   vektor pravých stran dané soustavy rovnic
%               x0  počáteční aproximace
%               w   relaxační parametr
%               eps požadovaná přesnost epsilon
% výstupní data: x   hledaná iterace
%               k   počet provedených iterací
```

```

% určení rozměru vstupních hodnot
n=size(A,1);
n1=size(b,1);
n2=size(x0,1);
% maximální počet provedených iterací
K=10000;
% ověření rozměrů A a b, kladnosti eps, zda w náleží (0,2)
if (n==n1 && eps>0 && w > 0 && w < 2)
    % mezivýpočty, kterými získáme matice D, L a U
    a=diag(A);
    D=diag(a);
    L=(tril(A)-D)*(-1);
    U=(triu(A)-D)*(-1);
    X=D-w*L;
    % výpočet Tw,g pro Relaxační metodu
    Tw=X\((1-w)*D+w*U); % \ inverzní matice
    g=X\ (w*b);
    spekt=max(abs(eig(Tw)));
    if (spekt>=1)
        disp('Info: Zvolte jinou metodu.');
```

error('Chyba: Posloupnost je divergující!');

```

    end
    % ověření zda je neprázdná počáteční aproximace x0
    if (isempty(x0)==1)
        % počáteční aproximace je prázdná, pro výpočet uijeme g
        x=g;
    else
        x=x0; % načte vektor ze vstupu
        if (n~=n2)
```

```

        error('Chyba: Zkontrolujte zda odpovídá rozměr počáteční
              aproximace x0 a vektoru b!')
    end
end
% Výpočet iterací, maximální počet iterací K
for k1=0:K
    z0=x;
    for i=1:n
        % pro sčítání vypočítaných prvků při násobení
        h=0;
        for j=1:n
            h=h+Tw(i,j)*x(j); % výpočet x pro 1 řádek
            if j==n
                x(i)=h+g(i);
            end
        end
    end
    z1=x;
    odchylka=norm(z1-z0)/norm(z0);
    if (odchylka<eps)
        disp('Info: Konec výpočtu, byla dosažena požadovaná
              přesnost. ');
        k=k1+1;
        break;
    end
end
else
error('Chyba: Zkontrolujte zda odpovídá stupeň matice A a rozměr
      vektoru b, zda omega náleží intervalu (0,2), nebo zda je
      kladné epsilon!')

```

```
end
end
```



Nyní budeme řešit příklad 2.10 pomocí m-filu 2.2. Abychom si ale předvedli užitečnost Relaxační metody, užijeme také m-file 2.3. Přesnost s jakou hledáme aproximaci \mathbf{x}^k zvolíme $\epsilon = 10^{-5}$ a hodnotu relaxačního parametru zvolíme $\omega = 1.0550$.

Příklad 2.11. Z příkladu 2.10 víme, že spektrální poloměr $\rho(\mathbf{T}_G) < 1$. Pak dle věty 2.4 platí, že Gaussova–Seidelova metoda pro tento systém konverguje. Konvergenci Relaxační metody jsme si ověřili v příkladu 2.10. Nyní do Matlabu vložíme potřebné vstupní parametry a zavoláme m-fily.

Řešení

```
>> A=[3 1 0;2 3 1;0 2 3];
>> b=[9;14;9];
>> x0=[ ];
>> w=1.0550;
>> eps=0.00001;
>> [x,k]=GAUSSOVA_SEIDELOVA(A,b,x0,eps)
Info: Konec výpočtu, byla dosažena požadovaná přesnost.
x =
    2.0000    3.0000    1.0000
k =
    11

>> [x,k]=RELAXACNI(A,b,x0,w,eps)
Info: Konec výpočtu, byla dosažena požadovaná přesnost.
x =
    2.0000    3.0000    1.0000
```

$\mathbf{k} =$

9

Relaxační metoda konverguje rychleji než Gaussova–Seidelova, protože relaxační parametr ω je větší než jedna.



Z důvodu přehlednosti nebudeme m-file používat pro větší systémy než stupně $n = 5$, ale lze je užít na jakkoliv rozsáhlý systém n lineárních rovnic o n neznámých. V příkladu 2.12 používáme m-file 2.3 pro soustavu stupně $n = 4$.

Příklad 2.12. Řešte systém $\mathbf{Ax} = \mathbf{b}$ tvaru

$$\mathbf{A} = \begin{pmatrix} 402 & -25 & 10 & 63 \\ 84 & 286 & 73 & 10 \\ 19 & 29 & 347 & 49 \\ 198 & 84 & 76 & 951 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 386 \\ -277 \\ 4559 \\ -9345 \end{pmatrix}, \quad \mathbf{x}^0 = \begin{pmatrix} 147 \\ 258 \\ 201 \\ 159 \end{pmatrix}$$

s počáteční aproximací \mathbf{x}^0 Relaxační metodou. Přesnost hledané aproximace \mathbf{x}^k zvolíme $\epsilon = 10^{-12}$. Jak se bude Relaxační metoda chovat, pokud budeme používat relaxační parametr pro dolní a horní relaxaci?

Řešení

Zadáme do Matlabu potřebné vstupní parametry. Protože nemáme zadaný přesný parametr ω , zkusíme chování metody postupně pro $\omega_1 = 0.0100, \omega_2 = 0.5000, \omega_3 = 0.9000, \omega_4 = 1.3000, \omega_5 = 1.7000$ a $\omega_6 = 2.1000$.

```
>> A=[402 -25 10 63;84 286 73 10;19 29 347 49;198 84 76 951];  
>> b=[386;-277;4559;-9345];  
>> x0=[147;258;201;159];  
>> eps=1.0000e-012;
```

```
>> w=0.0100;  
>> [x,k]=RELAXACNI(A,b,x0,w,eps)
```

Info: Konec výpočtu, byla dosažena požadovaná přesnost.

```

x =
    2.0000 -5.0000 15.0000 -11.0000
k =
    2836

>> w=0.5000;
>> [x,k]=RELAXACNI(A,b,x0,w,eps)
Info: Konec výpočtu, byla dosažena požadovaná přesnost.
x =
    2.0000 -5.0000 15.0000 -11.0000
k =
    49

>> w=0.9000;
>> [x,k]=RELAXACNI(A,b,x0,w,eps)
Info: Konec výpočtu, byla dosažena požadovaná přesnost.
x =
    2.0000 -5.0000 15.0000 -11.0000
k =
    18

>> w=1.3000;
>> [x,k]=RELAXACNI(A,b,x0,w,eps)
Info: Konec výpočtu, byla dosažena požadovaná přesnost.
x =
    2.0000 -5.0000 15.0000 -11.0000
k =
    34

```

```
>> w=1.7000;
>> [x,k]=RELAXACNI(A,b,x0,w,eps)
Info: Konec výpočtu, byla dosažena požadovaná přesnost.
x =
    2.0000 -5.0000 15.0000 -11.0000
k =
    2681
```

```
>> w=2.1000;
>> [x,k]=RELAXACNI(A,b,x0,w,eps)
??? Error using ==> RELAXACNI at 78
Chyba: Zkontrolujte zda odpovídá stupeň matice A a rozměr vektoru b, zda omega náleží intervalu (0,2), nebo zda je kladné epsilon!
```

Přesným řešením tohoto systému je $\mathbf{x}^* = (2, -5, 15, -11)^T$. Relaxační metoda tohoto systému pro ω blízké nulovým hodnotám konverguje s danou přesností, ale až při velkém počtu provedených iterací. Pro ω blízké 1 konverguje rychleji. Naopak čím více se relaxační parametr blíží hodnotě 2, tím je nutné provést více iterací pro dosažení hledané aproximace \mathbf{x}^k s přesností ϵ . Také jsme si ověřili, že tento m-file vypíše chybové hlášení pro $\omega > 2$.



Relaxační metoda, jak již bylo řečeno, může pro některé soustavy lineárních rovnic konvergovat i v případě, že Gaussova–Seidelova metoda diverguje. Tento případ charakterizuje následující příklad.

Příklad 2.13. Řešte následující systém $\mathbf{Ax} = \mathbf{b}$ s danou počáteční aproximací \mathbf{x}^0

$$\mathbf{A} = \begin{pmatrix} 2 & 3 & -3 & 2 \\ 1 & 1 & 1 & 1 \\ 3 & 3 & 2 & 1 \\ 1 & -1 & 1 & 2 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 6 \\ 2 \\ 2 \\ 0 \end{pmatrix}, \quad \mathbf{x}^0 = \begin{pmatrix} 20 \\ 9 \\ -24 \\ 4 \end{pmatrix}$$

Gaussovou–Seidelovou a Relaxační metodou s přesností $\epsilon = 10^{-4}$. Pro Relaxační metodu použijte parametr $\omega = 0.1517$.

Řešení

Přesným řešením je $\mathbf{x}^* = (-2, 2, 0, 2)^T$. Nyní zadáme vstupní parametry do Matlabu a zavoláme m-fily.

```
>> A=[2 3 -3 2;1 1 1 1;3 3 2 1;1 -1 1 2];
>> b=[6;2;2;0];
>> x0=[20;9;-24;4];
>> eps=0.0001;
>> [x,k]=GAUSSOVA_SEIDELOVA(A,b,x0,eps)
Info: Zvolte jinou metodu.
??? Error using ==> GAUSSOVA_SEIDELOVA at 34
Chyba: Posloupnost je divergující!
```

```
>> w=0.1517;
>> [x,k]=RELAXACNI(A,b,x0,w,eps)
Info: Konec výpočtu, byla dosažena požadovaná přesnost.
x =
    -1.9952    1.9976   -0.0010    1.9975
k =
    2020
```

Gaussova–Seidelova metoda pro tento systém tedy diverguje. Relaxační metoda při vhodné volbě relaxačního parametru bude konvergovat k přesnému řešení.



3 Analýza metod

V této kapitole budeme užívat uvedené m-fily z předchozí kapitoly na různých systémech lineárních rovnic $\mathbf{Ax} = \mathbf{b}$ a porovnávat dosažené výsledky.

Příklad 3.1. Použijte všechny tři uvedené m-fily na tuto matici \mathbf{A} a vektor \mathbf{b} s danou počáteční aproximací \mathbf{x}^0

$$\mathbf{A} = \begin{pmatrix} 10 & -2 & 2 & 1 \\ -1 & 10 & -2 & 3 \\ -1 & -1 & 10 & -1 \\ 2 & -2 & 1 & 10 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 17 \\ -12 \\ -2 \\ 36 \end{pmatrix}, \quad \mathbf{x}^0 = \begin{pmatrix} -5 \\ 4 \\ 6 \\ 9 \end{pmatrix}.$$

Hledejte aproximaci \mathbf{x}^k přesného řešení s přesností $\epsilon = 10^{-2}$. Pro Relaxační metodu použijte volbu parametrů $\omega_1 = 1.0123$ a $\omega_2 = 1.7000$.

Řešení

Přesným řešením je $\mathbf{x}^* = (1, -2, 0, 3)^T$. Zadáme obdobně jako v předchozích příkladech vstupní parametry a zavoláme m-fily.

```
>> [x,k]=JACOBY(A,b,x0,eps)
```

```
Info: Konec výpočtu, byla dosažena požadovaná přesnost.
```

```
x =
```

```
0.9953 -1.9969 -0.0010 3.0016
```

```
k =
```

```
4
```

```
>> [x,k]=GAUSSOVA_SEIDLOVA(A,b,x0,eps)
```

```
Info: Konec výpočtu, byla dosažena požadovaná přesnost.
```

```
x =
```

```
0.9996 -1.9999 -0.0000 3.0000
```

```
k =
```

```
4
```

```

>> w=1.0123;
>> [x,k]=RELAXACNI(A,b,x0,w,eps)
Info: Konec výpočtu, byla dosažena požadovaná přesnost.
x =
    0.9982 -1.9998 -0.0000 3.0000
k =
    4

>> w=1.7000;
>> [x,k]=RELAXACNI(A,b,x0,w,eps)
Info: Zvolte jinou metodu.
??? Error using ==> RELAXACNI at 37
Chyba: Posloupnost je divergující!

```

Na tomto systému jsme si ukázali, že i když Jacobiova i Gaussova–Seidelova metoda konvergují, může Relaxační metoda v závislosti na relaxačním parametru divergovat. Zde pro $\omega = 1.0123$ Relaxační metoda konverguje, ale pro $\omega = 1.7000$ diverguje.



Příklad 3.2. Porovnejte pro daný systém, jak mění se ϵ ovlivňuje přesnost a počet provedených iterací při použití všech tří metod pro stejnou počáteční aproximaci. Relaxační parametr je $\omega = 0.4514$. Je dáno

$$\mathbf{A} = \begin{pmatrix} 795 & 59 & 27 & 27 & 645 \\ 0 & 795 & 45 & 442 & 16 \\ 0 & 0 & 795 & 54 & 64 \\ 0 & 0 & 0 & 795 & 21 \\ 70 & 0 & 0 & 0 & 795 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} -30346 \\ 44304 \\ 18518 \\ 6300 \\ 27825 \end{pmatrix}, \mathbf{x}^0 = \begin{pmatrix} 26 \\ -64 \\ 24 \\ 92 \\ -46 \end{pmatrix}.$$

Řešení

Výsledky přehledně uspořádáme do následující tabulky. Přesným řešením tohoto systému je vektor $\mathbf{x}^* = (-71, 50, 20, 7, 35)^T$.

ϵ	JACOBOVA	GAUSSOVA–SEIDELOVA	RELAXACNI
10^{-1}	k=3 $\mathbf{x}^k = -75.9487$	k=3 $\mathbf{x}^k = -76.1877$	k=7 $\mathbf{x}^k = -65.8364$
	49.8880	49.8856	44.0472
	19.5188	19.4953	19.8025
	6.8421	6.8343	8.3590
	41.6873	41.7142	39.1292
10^{-3}	k=5 $\mathbf{x}^k = -76.6410$	k=5 $\mathbf{x}^k = -76.6432$	k=17 $\mathbf{x}^k = -76.5216$
	49.9924	49.9928	49.9579
	19.4691	19.4690	19.4755
	6.8219	6.8218	6.8283
	41.7483	41.7485	41.7233
10^{-8}	k=10 $\mathbf{x}^k = -76.6457$	k=9 $\mathbf{x}^k = -76.6457$	k=42 $\mathbf{x}^k = -76.6457$
	49.9934	49.9934	49.9934
	19.4688	19.4688	19.4688
	6.8217	6.8217	6.8217
	41.7487	41.7487	41.7487

Relaxační parametr je menší než jedna, a proto Relaxační metoda konverguje pomaluji než Gaussova–Seidelova metoda. Je názorné, že čím více se bude blížit přesnost ϵ k nule, tím více proběhne iterací a aproximace \mathbf{x}^k bude tím přesnější.



Příklad 3.3. Porovnejte jak změny počátečních aproximací ovlivňují počet provedených iterací k při použití všech tří metod. Je dáno

$$\mathbf{A} = \begin{pmatrix} 49 & -4 & 4 & 22 & 2 \\ 2 & 9 & 2 & -2 & 2 \\ -1 & 1 & 18 & -11 & 1 \\ 3 & 3 & 33 & 43 & -3 \\ -2 & 2 & -34 & 4 & 50 \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} 1150 \\ -69 \\ -828 \\ 976 \\ 959 \end{pmatrix},$$

s přesností $\epsilon = 0.0010$ hledané aproximace \mathbf{x}^k a parametrem $\omega = 0.7898$.

Řešení

Výsledky jsme opět z důvodu přehlednosti uspořádali do tabulky. Přesným řešením je $\mathbf{x}^* = (8, 1, -22, 40, 15)^T$.

\mathbf{x}^0	JACOPIOVA	GAUSSOVA–SEIDELOVA	RELAXACNI
162 282 -414 330 0	k=17 $\mathbf{x}^k = 8.0672$ 3.8174 -22.0613 38.8850 1.2375	k=12 $\mathbf{x}^k = 8.0741$ 3.8074 -22.0699 38.8919 1.2352	k=8 $\mathbf{x}^k = 8.0755$ 3.8123 -22.0665 38.8893 1.2343
-77 58 52 -96 -15	k=16 $\mathbf{x}^k = 8.0679$ 3.8163 -22.0622 38.8857 1.2369	k=11 $\mathbf{x}^k = 8.0747$ 3.8067 -22.0706 38.8924 1.2354	k=8 $\mathbf{x}^k = 8.0754$ 3.8139 -22.0671 38.8892 1.2338
8523 9874 6321 4523 -9538	k=22 $\mathbf{x}^k = 8.0743$ 3.8057 -22.0706 38.8933 1.2313	k=15 $\mathbf{x}^k = 8.0666$ 3.8163 -22.0612 38.8859 1.2328	k=11 $\mathbf{x}^k = 8.0759$ 3.8137 -22.0671 38.8892 1.2338

Relaxační parametr je jen o málo menší než jedna, a proto Relaxační metoda pro tento systém konverguje rychleji než Gaussova–Seidelova metoda. Provedené výpočty prokázaly, že přiblížení se přesnému řešení a počet potřebných iterací k je závislý na počáteční aproximaci \mathbf{x}^0 .



Závěr

Primárním cílem této bakalářské práce bylo nastudovat pro mne nové Iterační metody pro řešení systémů lineárních rovnic. Práci zpracovat tak, aby mohla sloužit jako studijní materiál pro studenty se základním kurzem lineární algebry. Dalším neméně důležitým cílem bylo nastudovat možnosti řešení systémů lineárních rovnic pomocí těchto metod, předvést je na vlastních příkladech a zapracovat do této práce vlastnoručně vytvořené m-file v matematickém softwaru Matlab. Doufám, že tyto cíle byly dostatečně splněny.

V této bakalářské práci je prezentován nezbytný úvod k pochopení Iteračních metod. Dále jsou zde uvedeny tři základní iterační metody, kterými jsou Jacobiova, Gaussova–Seidelova a Relaxační metoda. Ke každé z nich jsou uvedeny nejdůležitější věty a tvrzení. Pro snadnější pochopení je každá tato část obohacena o příklady počítané ručně, o m-fily samostatně vytvořené a posléze i příklady počítané pomocí těchto m-filů. V poslední kapitole jsem se snažila dostatečně porovnat zmíněné tři metody pomocí aplikování vytvořených m-filů na různé příklady.

Přínosem při tvorbě této bakalářské práce pro mne bylo bližší seznámení s jednou z částí numerické matematiky, práce se zpracováváním odborné literatury a zdokonalení mých dovedností s typografickým programem \LaTeX , ve kterém je práce vysázena. Samozřejmě jsem musela překonávat i různé překážky, jednou z hlavních byla práce v matematickém softwaru Matlab. Po usilovné pili a snaze jsem snad ale způsob práce v tomto softwaru dostatečně pochopila.

Věřím, že tato práce poslouží nejen mě, ale i mnoha dalším čtenářům, pro získání mnoha užitečných informací.

Literatura

- [1] Boháč Z., Častová N.: Základní numerické metody, VŠB - Technická univerzita Ostrava, Ostrava, 2004.
- [2] Horová I., Zelinka J.: Numerické metody, skriptum MU, Brno, 2004.
- [3] Hort D., Rachůnek J.: Algebra 1, UP v Olomouci, Olomouc, 2005.
- [4] Krejsa, M.: Algoritmizace inženýrských výpočtů, VŠB - Technická univerzita Ostrava, Ostrava, 2011.
- [5] Kučera R.: Numerické metody, VŠB - Technická univerzita Ostrava, Ostrava, 2006.
- [6] Ralston A.: Základy numerické matematiky, Československá akademie věd, Praha, 1978.
- [7] Segethová J.: Základy numerické matematiky, Karolinum, Praha, 2002.
- [8] Šik F.: Lineární algebra zaměřená na numerickou analýzu, Masarykova univerzita, Brno, 1988.
- [9] Žáková K.: Základy práce v Matlabe, Slovenská technická univerzita, Bratislava, 2006.
- [10] Lineární algebra, dostupná na: <http://vondrak.am.vsb.cz/la-it/Books/LI-NALG.pdf>, [citováno 28. 12. 2010].