

UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Iterační metody řešení soustav lineárních rovnic
Václav Baťka

Bakalářská práce
2012

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: 2011/2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Václav Bařka**
Osobní číslo: **I08013**
Studijní program: **B2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Iterační metody řešení soustav lineárních rovnic**
Zadávatel katedra: **Katedra informačních technologií**

Z á s a d y p r o v y p r a c o v á n í :

Bakalářská práce se zabývá iteračními metodami řešení soustav lineárních rovnic. V teoretické části budou nadefinovány základní pojmy z teorie systémů rovnic, budou popsány vlastnosti řešení soustav lineárních rovnic a budou popsány základní numerické metody řešící tyto soustavy (přímé i nepřímé metody). V další kapitole teoretické části budou podrobně popsány iterační metody řešení, zejména Gauss-Seidelova metoda a Jacobiho metoda. Budou popsány klady a zápory implementace v porovnání s přímými metodami a řešeny problémy s konvergencí obou metod.

Implementační část bakalářské práce se bude soustředit na vytvoření UserFriendly software pro řešení soustav lineárních rovnic iteračními metodami. Pro vytvoření softwaru je preferován programovací jazyk Java. V následující části budou popsány algoritmy pro řešení systémů rovnic, které jsou součástí jádra Matlabu. Na závěr bude vyvinutý software testován na typových příkladech a výsledky (přesnost, časová náročnost, ...) budou porovnány s výstupy z Matlabu.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

Dušek F.: Matlab a Simulink - úvod do používání, Pardubice 2000

E. Vitásek: Numerické metody. SNTL, Praha 1987

Horová I., Zelinka J.: Numerické metody, Brno 2004

Vedoucí bakalářské práce:

Mgr. Jana Heckenbergerová, Ph.D.

Katedra matematiky a fyziky

Datum zadání bakalářské práce: **16. prosince 2011**

Termín odevzdání bakalářské práce: **11. května 2012**



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 30. března 2012

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 9. 5. 2012

Václav Bařka

Poděkování

V první řadě by chtěl autor poděkovat své rodině, která ho po dobu studia podporovala, jak finančně, tak psychicky. Dále by chtěl poděkovat vedoucí Mgr. Janě Heckenbergerové Ph.D., která autora v průběhu práce směřovala ke zdárnému konci. V neposlední řadě přátelům, kteří autora po dobu studia povzbuzovali.

Anotace

Práce se zabývá řešením soustav lineárních rovnic, jak přímými tak hlavně iteračními metodami. Srovnává navrženou aplikaci s Matlabem.

Klíčová slova

Soustava lineárních rovnic, přímé metody, iterační metody, SLR,

Title

Iterative methods for solving systems of linear equations

Annotation

The work deals with the solving systems of linear equations, both direct and iterative methods mainly. Compares the proposed application with Matlab

Keywords

systems of linear equations, direct methods, iterative methods,

Obsah

Seznam grafů	8
Seznam tabulek	9
Seznam obrázků.....	10
Úvod	11
1 Základní pojmy z teorie systémů rovnic	12
1.1 Soustava lineárních rovnic.....	12
2 Přímé metody řešení systémů lineárních rovnic	15
2.1 Gaussova eliminační metoda	15
2.2 Gaussova eliminační metoda s částečným výběrem hlavního prvku	17
2.3 Gaussova eliminační metoda s úplným výběrem hlavního prvku	19
2.4 LU rozklad	20
2.5 Choleského metoda	22
2.6 Croutova metoda	24
3 Nepřímé metody řešení systémů lineárních rovnic	27
3.1 Princip iteračních metod	27
3.2 Jacobiova iterační metoda	27
3.3 Gaussova-Seidelova iterační metoda.....	30
3.4 Ukončovací kritéria iteračních metod	32
4 Algoritmy pro výpočet iteračních metod v Matlabu	33
4.1 BICG – Metoda bi-sdružených gradientů.....	33
4.2 BICGSTAB – Stabilizovaná metoda bi-sdružených gradientů	34
4.3 CGS – Čtvercová metoda sdružených gradientů	34
4.4 GMRES – Zobecněná metoda minimálního rezidua	35
4.5 QMR – Metoda částečně minimálního rezidua	35
5 Popis praktické části bakalářské práce	36
5.1 Popis grafického rozhraní programu	36
5.2 Popis použitých algoritmů.....	37
6 Srovnání s matematickým softwarem Matlab	39
6.1 Srovnání časových nároků při výpočtu	39
6.2 Srovnání přesnosti výsledků	44
Závěr	46

Literatura	47
Příloha A – Dokumentace	49
Příloha B – Zdrojové kódy.....	50

Seznam grafů

Graf 1 Srovnání časové náročnosti Jacobiovy metody.....	41
Graf 2 Srovnání časové náročnosti Jacobiovy metody maticovým způsobem	42
Graf 3 Srovnání časové náročnosti G-S metody	43
Graf 4 Srovnání časové náročnosti G-S metody maticovým způsobem	44

Seznam tabulek

Tabulka 1– Iterační výpočet SLR pomocí Jacobiovy metody.....	30
Tabulka 2– Iterační výpočet SLR pomocí G-S metody	31
Tabulka 3 Srovnání časové náročnosti Jacobiovy metody	40
Tabulka 4 Srovnání časové náročnosti Jacobiovy metody maticovým způsobem.....	41
Tabulka 5 Srovnání časové náročnosti G-S metody.....	42
Tabulka 6 Srovnání časové náročnosti G-S metody maticovým způsobem	43
Tabulka 7 Srovnání přesnosti výsledků Jacobiovy metody.....	44
Tabulka 8 Srovnání přesnosti výsledků G-S metody	45

Seznam obrázků

Obrázek 1 Struktura praktické části BP	36
Obrázek 2 Grafické rozhraní BP	36
Obrázek 3 Kód pro výpočet Jacobiovy metody v Matlabu	39
Obrázek 4 Kód pro výpočet G-S metody v Matlabu	40

Úvod

Tato bakalářská práce si klade za cíl seznámit čtenáře s hlavními aspekty teorie systémů lineárních rovnic. V první kapitole seznamuje čtenáře se základními pojmy teorie systému lineárních rovnic. V druhé kapitole se nejdříve obecně věnuje problematice přímých iteračních metod. Poté se důkladně projdou jednotlivé přímé metody (Gaussova eliminační metoda bez výběru i s výběrem hlavního prvku, LU rozklad, Choleského metoda, Croutova metoda pro třídiagonální matice) a ukáže se jejich postup řešení na příkladech. V třetí kapitole se seznámí čtenář s iteračními metodami řešení soustav lineárních rovnic. Projdou se Jacobiova a Gauss-Seidelova metoda, ukáže se názorně průběh řešení na vzorových příkladech. Dále se čtenář seznámí s ukončovými kritérii pro iterační výpočty. Čtvrtou kapitolou si čtenář projde algoritmy pro řešení soustav lineárních rovnic iteračními metodami, které jsou součástí jádra Matlabu. Pátá kapitola seznamuje čtenáře s praktickou částí bakalářské práce, ukazuje návrh grafického rozhraní aplikace pro řešení soustav lineárních rovnic iteračními metodami a dále popisuje algoritmy, které byly napsány v programovacím jazyce Java. Šestá kapitola srovnává časovou náročnost výpočtu praktické části bakalářské práce a Matlabu. Srovnává se na posloupnosti příkazů, které se vykonávají v praktické části bakalářské práce a dále srovnává přesnost výsledků.

1 Základní pojmy z teorie systémů rovnic

Matice nad tělesem P je zobrazení $\{1, 2, \dots, n\} \times \{1, 2, \dots, n\} \rightarrow P$. Zapisuje se většinou velkými písmeny. $A = \{\dots\}$. Je to struktura tvořená n sloupci a m řádky.

Čtvercová matice je matice tvořena n sloupci a n řádky.

$$A = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

Nulová matice je taková, kterou má na všech pozicích a_{ij} hodnotu 0.

$$A = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

Jednotková matice je taková čtvercová matice, která má na hlavní diagonále prvky s hodnotou 1 a na všech zbylých pozicích nulové prvky. Hlavní diagonála obsahuje prvky a_{ij} pro $i=j$.

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Matice transponovaná k matici A se značí A^T a platí pro ni $a_{ij} = a^T_{ji}$. To znamená, že prvky které byly na i -tém řádku a j -tém sloupci budou nově na j -tém řádku a i -tém sloupci.

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix} \quad A^T = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$$

1.1 Soustava lineárních rovnic

Za soustavou lineárních rovnic považujeme libovolně velkou množinu lineárních rovnic. Obecně může být soustava m lineárních rovnic s n proměnnými zapsána následovně.

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n = b_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \dots + a_{3n}x_n = b_3$$

:

$$a_{m1}x_1 + a_{m2}x_2 + a_{m3}x_3 + \dots + a_{mn}x_n = b_m$$

Proměnné x_1 až x_n jsou neznámé a a_{ij} jsou koeficienty vystupující v soustavě rovnic. Členy b_i jsou absolutní členy soustavy nebo též vektorem pravé strany. V obecném případě mohou být koeficienty i absolutní členy komplexními čísly. Koeficienty soustavy lze zapsat také v maticovém tvaru. Tuto matici poté nazýváme jako matici soustavy. Stejně tak můžeme neznámé, ale také koeficienty pravé strany zapsat jako vektory. Vektor neznámých \vec{x} a vektor pravé strany \vec{b} .

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{pmatrix}, \quad \vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix}, \quad \vec{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_m \end{pmatrix}$$

Pokud máme soustavu lineárních rovnic vyjádřenou jako matici soustavy, vektor neznámých a vektor pravé strany lze celou soustavu lineárních rovnic zapsat ve tvaru.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_m \end{pmatrix}$$

$$A * \vec{x} = \vec{b}$$

Matice A a vektor pravé strany se někdy zapisuje do jedné matice, kterou označujeme A|b. Tvoří ji celá matice A do které za poslední sloupec přidáme řádky vektoru pravé strany b.

$$A|b = \left(\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & b_1 \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} & b_2 \\ a_{31} & a_{32} & a_{33} & \dots & a_{3n} & b_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} & b_m \end{array} \right)$$

Matice A|b se nazývá rozšířená matice soustavy. Je-li vektor pravé strany b nulový, potom se jedná o soustavu homogenní. Pokud je aspoň jeden prvek vektorů b_i , $i = 1, 2, \dots, m$ nenulový, potom se jedná o soustavu nehomogenní.

Soustavu můžeme označit řešitelnou či neřešitelnou, jestli existuje alespoň jedno, respektive neexistuje žádné řešení. Aby soustava $A * \vec{x} = \vec{b}$ měla řešení, plně postačuje rovnost hodnoty matice soustavy $h(A)$ a hodnost rozšířené matice soustavy $h(A|b)$. Řešení nehomogenních soustav má tři konce. Prvním koncem je, že daná soustava nemá řešení, dále pak soustava má právě jedno řešení. Poslední eventualitou je soustava s nekonečným počtem řešení.

2 Přímé metody řešení systémů lineárních rovnic

Přímé metody lze aplikovat na regulárních čtvercových maticích soustav řádu n ($A \in M_n$), kde b je vektor pravé strany. Systém má jediné řešení.

$$x^* = A^{-1} * b$$

Jsou to metody, které po aplikaci konečným počtu kroků dají přesný výsledek x^* , ovšem za předpokladu že všechny aritmetické operace proběhly matematicky přesně.

2.1 Gaussova eliminační metoda

Jedná se o jednu z nejznámějších přímých metod řešení systémů rovnic. “Hlavní myšlenka této metody spočívá v převedení daného systému $A x = b$ vhodnými ekvivalentními úpravami na systém $R x = c$ s horní trojúhelníkovou maticí R , tj. problém se převede na redukovaný problém.”¹ Gaussovou eliminační metodou lze také řešit výpočet inverzní matice, nebo výpočet determinantu matice.

Matice $A = (a_{ij})$ tvaru $m \times n$ je v řádkově odstupňovaném tvaru, jestliže splňuje dvě podmínky. Obsahuje-li i -tý řádek A_{i*} samé nuly, pak všechny pod i -tým řádkem jsou také nulové. Je-li v i -tém řádku A_{i*} první nenulový prvek v j -tém sloupci, tedy na místě a_{ij} , pak všechny prvky ležící v prvních j sloupcích a současně pod i -tým řádkem se rovnají nule. První nenulový prvek v každém řádku nazýváme hlavním prvkem.

V prvním kroku algoritmu vynásobíme první řádek číslem $-k_1$, které se součtem druhého řádku odstraní hodnotu a_{21} ze soustavy. V druhém kroku taktéž vynásobíme první řádek číslem $-k_2$ které tentokrát eliminuje hodnotu a_{31} . Ve třetím kroku vynásobíme třetí řádek $-k_3$ a tím odstraníme poslední prvek a_{32} k dokončení algoritmu.

$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{21} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{array} \right] \begin{array}{l} \downarrow (-k_1) \\ \downarrow (-k_2) \end{array} \sim \left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & \tilde{a}_{21} & \tilde{a}_{23} & \tilde{b}_2 \\ 0 & \tilde{a}_{32} & \tilde{a}_{33} & \tilde{b}_3 \end{array} \right] \downarrow (-k_3) \sim$$
$$\left[\begin{array}{ccc|c} a_{11} & a_{12} & a_{13} & b_1 \\ 0 & \tilde{a}_{21} & \tilde{a}_{23} & \tilde{b}_2 \\ 0 & 0 & \tilde{\tilde{a}}_{33} & \tilde{\tilde{b}}_3 \end{array} \right]$$

¹ (Horová, a další, 2008)

Následující krok rozhoduje o řešitelnosti soustavy. Soustavy řešené Gaussovou eliminační metodou jsou neřešitelné, pokud $h(A|b) > h(A)$, vždy když $\tilde{a}_{33} = 0$, $\tilde{b}_3 \neq 0$. Jsou řešitelné případe rovnosti hodnotí matic $h(a|b)=h(a)$.

V kroku tři musíme zjistit počet neznámých, které si můžeme zvolit a počet neznámých vyjádřených parametrem.

$$k = n - h(A)$$

K je číslo volnosti, které nám ukazuje kolik neznámých si můžeme zvolit, nebo které musíme nahradit parametrem. N je celkový počet neznámých obsažených v soustavě.

V případě kdy dostaneme $k=0$, tak výsledek soustavy je jednoznačný - například $x_1 = 1$, $x_2 = 2$, $x_3 = 3$. Při výsledku, kdy se k rovná jedné, výsledek soustavy bude v podobném tvaru $x_1 = 1 + p$, $x_2 = 2 - p$, $x_3 = p$. Kde p si zvolíme a soustava má nekonečně mnoho řešení.

V posledním kroku algoritmu z redukované matice $A|b$ se vyjádříme neznámé.

$$\begin{array}{r}
 a_{11} x_1 + a_{12} x_2 + a_{13} x_3 \\
 a_{22} x_2 + a_{23} x_3 \\
 \tilde{a}_{33} x_3
 \end{array}
 \left|
 \begin{array}{l}
 \tilde{b}_3 \\
 \tilde{b}_2 \\
 b_1
 \end{array}
 \right.
 \begin{array}{l}
 x_3 = \frac{\tilde{b}_3}{\tilde{a}_{33}} \\
 x_2 = \frac{\tilde{b}_2}{\tilde{a}_{22}} - \tilde{a}_{23} x_3 \\
 x_1 = \frac{b_1}{a_{11}} - a_{12} x_2 - a_{13} x_3
 \end{array}$$

Nyní si tuto metodu názorně převedeme na příkladu.

$$\begin{array}{r}
 2x_1 + 4x_2 - x_3 = -5 \\
 x_1 + x_2 - 3x_3 = -9 \\
 4x_1 + x_2 + 2x_3 = 9
 \end{array}$$

Eliminujeme hodnoty a_{21} , a_{31} na hodnotu 0. První řádek násobíme $\frac{1}{2}$ a odečteme od druhého řádku. Poté první násobíme 2 a odečteme od třetího řádku.

$$\left(\begin{array}{ccc|c}
 2 & 4 & -1 & -5 \\
 1 & 1 & -3 & -9 \\
 4 & 1 & 2 & 9
 \end{array} \right) \begin{array}{l} \downarrow \left(\frac{1}{2}\right) \\ \downarrow (2) \end{array}$$

$$\left(\begin{array}{ccc|c} 2 & 4 & -1 & -5 \\ 0 & -1 & -\frac{5}{2} & -\frac{13}{2} \\ 0 & -7 & 4 & 19 \end{array} \right) \downarrow (-7)$$

Druhý řádek již máme ve tvaru, který potřebujeme. Nyní druhý řádek vynásobíme -7 a sečteme s posledním řádkem rozšířené matice soustavy.

$$\left(\begin{array}{ccc|c} 2 & 4 & -1 & -5 \\ 0 & -1 & -\frac{5}{2} & -\frac{13}{2} \\ 0 & 0 & \frac{43}{2} & \frac{129}{2} \end{array} \right)$$

V tuto chvíli se nám soustava natolik zjednodušila do počitatelného tvaru.

$$\begin{aligned} 2x_1 + 4x_2 - 1x_3 &= -5 \\ -x_2 - \frac{5}{2}x_3 &= -\frac{13}{2} \\ \frac{43}{2}x_3 &= \frac{129}{2} \end{aligned}$$

K řešení neznámých využijeme tzv. zpětný chod. Při této metodě se výpočet začíná od poslední neznámé.

$$x_3 = \frac{258}{86} = 3, \quad x_2 = \frac{13}{2} - 3 \frac{5}{2} = -1, \quad x_1 = \frac{7-5}{2} = 1$$

2.2 Gaussova eliminační metoda s částečným výběrem hlavního prvku

Při výpočtu Gaussovou eliminační metodou musíme předpokládat, že v průběhu algoritmu nám vznikají nenulová čísla, kterými v dalších krocích dělíme. Protože jsou tato čísla velmi malá, v absolutní hodnotě blízcí se nule, vznikají při zaokrouhlování velké nepřesnosti a tím ovlivňují celkový výsledek. Toto se dá řešit metodami s výběrem hlavního prvku.

V první řadě je to řádkový výběr hlavního prvku. V k -tém kroku prohledáváme k -tý řádek a hledáme prvek, který je v absolutní hodnotě větší, než zbylé hodnoty v řádku.

Prohledáváme sloupce, ze kterých jsme dosud nevybrali hlavní prvek. Hlavní prvek se nachází v p-tém řádku a q-tém a platí pro něj

$$|a_{kq}| = \max_j |a_{ij}^{(k-1)}|.$$

Druhou možností je sloupcový výběr hlavního prvku. Kdy v k-tém kroku vybíráme z k-tého sloupce hodnotu, který je v absolutní hodnotě maximální ze všech hodnot v daném sloupci. Výběr probíhá v řádcích, ze kterých ještě nebyl vybrán hlavní prvek. Hlavní prvek je potom v p-tém řádku a platí pro něj

$$|a_{pk}| = \max_i |a_{ik}^{(k-1)}|.$$

Následující příklad je řešen Gaussovou eliminační metodou s částečným výběrem hlavního prvku ve sloupcích.

$$\left(\begin{array}{ccc|c} 1 & 2 & 3 & 14 \\ 2 & 4 & 5 & 25 \\ \boxed{7} & 8 & 9 & 50 \end{array} \right) \sim \left(\begin{array}{ccc|c} \boxed{7} & 8 & 9 & 50 \\ 2 & 4 & 5 & 25 \\ 1 & 2 & 3 & 14 \end{array} \right) \begin{array}{l} \downarrow \left(-\frac{2}{7} \right) \\ \downarrow \left(-\frac{1}{2} \right) \end{array}$$

$$\left(\begin{array}{ccc|c} 7 & 8 & 9 & 50 \\ 0 & \boxed{\frac{12}{7}} & \frac{17}{7} & \frac{75}{7} \\ 0 & \frac{6}{7} & 3 & \frac{48}{7} \end{array} \right) \downarrow \left(-\frac{1}{2} \right)$$

$$\left(\begin{array}{ccc|c} 7 & 8 & 9 & 50 \\ 0 & \frac{12}{7} & \frac{17}{7} & \frac{75}{7} \\ 0 & 0 & \frac{1}{2} & \frac{3}{2} \end{array} \right)$$

$$x_1 = 1, \quad x_2 = 2, \quad x_3 = 3$$

2.3 Gaussova eliminační metoda s úplným výběrem hlavního prvku

Základní myšlenkou metody je prohledávat submatici, která se zmenšuje každým již hotovým řádkem trojúhelníkové matice. Jistou nevýhodou eliminace s úplným výběrem hlavního prvku je potřeba měnit pořadí jak řádků, tak sloupců, tím je tato metoda méně efektivnější než eliminace s částečným výběrem hlavního prvku.

Hlavní prvek v k-tém kroku eliminace, $k=1, \dots, n-1$, vybíráme ze sloupců a řádků, ze kterých jsme dosud nevybírali. Hlavní prvek se nachází v p-tém řádku a q-tém sloupci a platí pro něj

$$|a_{pq}| = \max_{ij} |a_{ij}^{(k-1)}|.$$

Nejprve prohledáváme celou matici a hledáme prvek, který je v absolutní hodnotě maximální z celé soustavy.

$$\left(\begin{array}{ccc|c} 1 & 2 & 3 & 4 \\ 2 & 4 & 5 & 8 \\ 7 & 8 & 9 & 22 \end{array} \right) \sim \left(\begin{array}{ccc|c} 9 & 8 & 7 & 22 \\ 5 & 4 & 2 & 8 \\ 3 & 2 & 2 & 4 \end{array} \right) \downarrow \left(-\frac{5}{9} \right) \\ \downarrow \left(-\frac{3}{9} \right)$$

$$\left(\begin{array}{ccc|c} 9 & 8 & 7 & 4 \\ 0 & -\frac{4}{9} & -\frac{17}{9} & -\frac{38}{9} \\ 0 & -\frac{2}{3} & -\frac{4}{3} & -\frac{10}{3} \end{array} \right)$$

Nyní hledáme prvek od indexu a_{22} v našem případě je to $-\frac{17}{9}$. Tentokrát ale již řádky posunovat nemusíme, vystačíme si pouze výměnou sloupců.

$$\left(\begin{array}{ccc|c} 9 & 7 & 8 & 4 \\ 0 & -\frac{17}{9} & -\frac{4}{9} & -\frac{38}{9} \\ 0 & -\frac{4}{3} & -\frac{2}{3} & -\frac{10}{3} \end{array} \right) \downarrow \left(-\frac{12}{17} \right)$$

$$\left(\begin{array}{ccc|c} 9 & 7 & 8 & 4 \\ 0 & -\frac{17}{9} & -\frac{4}{9} & -\frac{38}{9} \\ 0 & 0 & -\frac{18}{51} & -\frac{18}{51} \end{array} \right) \Rightarrow \vec{x} = (0 \ 2 \ 1)$$

V průběhu eliminace jsme zaměňovali sloupce, proto teď na konci algoritmu musíme opět provést zpětně výměny sloupců vektoru výsledků \vec{x} .

$$x_1 = 2, \quad x_2 = 1, \quad x_3 = 0$$

2.4 LU rozklad

Alan Turing, autor LU rozkladu matice $A \in R^{n \times n}$ postupoval tak, že postupně počítal m -tý řádek matice U a m -tý sloupec matice L , $m = 1, \dots, n$. Příslušné vzorce odvozoval pomocí vzorce pro násobení matic.

$$A = L \cdot U$$

$$a_{ij} = \sum_{k=1}^n l_{ik} u_{kj}$$

„Je-li matice řádkově, nebo sloupcově diagonálně dominantní tj.

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

nebo

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

pak LU rozklad existuje. Speciálně, je-li matice sloupcově diagonálně dominantní, je $|l_{ij}| \leq 1 \forall i, j = 1, \dots, n$.²

² (Felcman, 2012)

Pro regulární matice, ve kterých lze měnit pořadí řádků, tento rozklad lze udělat. Matice U je horní trojúhelníková matice, kterou dostaneme výše zmíněnou Gaussovou eliminací. Matice L je dolní trojúhelníková a na hlavní diagonále jedničky.

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

Vše si názorně předvedeme na příkladu.

$$\begin{aligned} 2x_1 - x_2 + 7x_3 &= 11 \\ 8x_1 - 2x_2 + 5x_3 &= 25 \\ 4x_1 + 3x_2 - 2x_3 &= 16 \end{aligned}$$

$$\begin{pmatrix} 2 & -1 & 7 \\ 8 & -2 & 5 \\ 4 & 3 & -2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} \times \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}$$

Nyní si naplníme hodnoty matic L a U.

$$a_{11} = 2 = 1 * u_{11} \Rightarrow u_{11} = 2$$

$$a_{21} = 8 = l_{21} * u_{11} \Rightarrow l_{21} = 4$$

$$a_{31} = 4 = l_{31} * u_{11} \Rightarrow l_{31} = 2$$

$$a_{12} = -1 = 1 * u_{12} \Rightarrow u_{12} = -1$$

$$a_{22} = -2 = l_{21} * u_{12} + 1 * u_{22} \Rightarrow u_{22} = 2$$

$$a_{32} = 3 = l_{31} * u_{12} + l_{32} * u_{22} \Rightarrow l_{32} = 2,5$$

$$a_{13} = 7 = 1 * u_{13} \Rightarrow u_{13} = 7$$

$$a_{23} = 5 = l_{21} * u_{13} + 1 * u_{23} \Rightarrow u_{23} = -23$$

$$a_{33} = -2 = l_{31} * u_{13} + l_{32} * u_{23} + 1 * u_{33} \Rightarrow u_{33} = 41,5$$

$$\begin{pmatrix} 2 & -1 & 7 \\ 8 & -2 & 5 \\ 4 & 3 & -2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 2 & 2,5 & 1 \end{pmatrix} \times \begin{pmatrix} 2 & -1 & 7 \\ 0 & 2 & -23 \\ 0 & 0 & 41,5 \end{pmatrix}$$

V tuto chvíli máme naplněné matice L a U a můžeme pokračovat dále ve výpočtu. V první řadě musíme vypočítat soustavu $L\vec{y} = \vec{b}$, dále z této soustavy dostaneme vektor \vec{y} , který dosadíme do další soustavy $U\vec{x} = \vec{y}$ a tím dostaneme vektor výsledku \vec{x} .

$$\begin{pmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 2 & 2,5 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 11 \\ 25 \\ 16 \end{pmatrix}$$

$$y_1 = 11$$

$$y_2 = 25 - 4 * y_1 = -19$$

$$y_3 = 16 - 2 * y_1 - 2,5 * y_2 = 41,5$$

V tuto chvíli přistupujeme k poslednímu kroku.

$$U\vec{x} = \vec{y}$$

$$\begin{pmatrix} 2 & -1 & 7 \\ 0 & 2 & -23 \\ 0 & 0 & 41,5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 11 \\ -19 \\ 41,5 \end{pmatrix}$$

$$x_3 = 1$$

$$x_2 = 23 * x_3 - 19 = 2$$

$$x_1 = 11 + x_2 - 7 * x_3 = 3$$

2.5 Choleského metoda

Autor metody André-Louis Cholesky ji založil na předpokladu pozitivně definitivní čtvercové matici A a na součinu horní a dolní trojúhelníkové matice a druhá matice je z první transponovaná. Matice L se nazývá Choleského trojúhelník matice A. Využití této metoda má při výpočtu inverzní matice podle vzorce $A^{-1} = (L^{-1})^T \cdot L^{-1}$, ale hlavně se využívá při řešení soustav lineárních rovnic.

$$A = L \cdot L^T$$

Metoda opět směřuje k řešení dvou systémů trojúhelníkových matic $L^T \vec{y} = \vec{b}$, $L\vec{x} = \vec{y}$. Výpočet probíhá po sloupcích, vzorce pro výpočet jednotlivých prvků vypadají následujícím způsobem.

$$l_{11} = \sqrt{a_{11}}$$

$$l_{1j} = \frac{a_{1j}}{l_{11}}, \quad j = 2, \dots, n$$

$$l_{ii} = \sqrt{a_{ii} - \sum_{t=1}^{i-1} l_{ti}^2} \quad i = 2, \dots, n$$

$$l_{ij} = \frac{1}{l_{ii}} a_{ij} - \sum_{t=1}^{i-1} l_{ti} l_{tj} \quad \text{platí pro } j > i$$

$$l_{ij} = 0 \quad \text{platí pro } j > i$$

Průběh výpočtu si ukážeme na následujícím případě, kdy máme zadanou soustavu rovnic.

$$\begin{aligned} x_1 + 2x_2 - x_3 &= 1 \\ 2x_1 + 2x_2 + 4x_3 &= 3 \\ -x_1 + 4x_2 + 8x_3 &= 6 \end{aligned}$$

$$l_{11} = 1, \quad l_{12} = 2, \quad l_{13} = -1$$

$$l_{22} = \sqrt{a_{22} - l_{12}^2} = i\sqrt{2} \quad l_{23} = \frac{1}{l_{22}}(a_{23} - l_{13}l_{12}) = -i3\sqrt{2}$$

$$l_{33} = \sqrt{a_{33} - (l_{13}^2 + l_{23}^2)} = 5$$

$$T = \begin{pmatrix} 1 & 2 & -1 \\ 0 & i\sqrt{2} & -i3\sqrt{2} \\ 0 & 0 & 5 \end{pmatrix}$$

Teď když už máme připravenou matici L tak smíme přistoupit k dalšímu výpočtu $L^T \vec{y} = \vec{b}$.

$$\begin{pmatrix} 1 & 0 & 0 \\ 2 & i\sqrt{2} & 0 \\ -1 & -i3\sqrt{2} & 5 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 1 \\ 3 \\ 6 \end{pmatrix}$$

Nyní získáme hodnotu vektoru $\vec{y}(1, -\frac{i\sqrt{2}}{2}, 2)$ a pokračujeme v řešení posledním krokem $L\vec{x} = \vec{y}$ a tím získáme hodnotu vektoru neznámých \vec{x} .

$$\begin{pmatrix} 1 & 2 & -1 \\ 0 & i\sqrt{2} & -i3\sqrt{2} \\ 0 & 0 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 1 \\ -\frac{i\sqrt{2}}{2} \\ 2 \end{pmatrix}$$

$$x_1 = 0, \quad x_2 = \frac{7}{10}, \quad x_3 = \frac{2}{5}$$

2.6 Croutova metoda

Prescott Durand Crout navrhl metodu, která dokáže třídiagonální matice převést na součin dolní a horní trojúhelníkové matice.

$$A = \begin{pmatrix} a_{11} & a_{12} & 0 & \dots & 0 \\ a_{21} & a_{22} & a_{23} & & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & a_{n-1,n} \\ 0 & \dots & 0 & a_{n,n-1} & a_{n,n} \end{pmatrix}$$

Potřebujeme matici A nyní rozložit na matice L a U

$$L = \begin{pmatrix} l_{11} & 0 & 0 & \dots & 0 \\ l_{21} & l_{22} & & & 0 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \dots & 0 & l_{n,n-1} & l_{n,n} \end{pmatrix} \quad U = \begin{pmatrix} 1 & u_{12} & 0 & \dots & 0 \\ a_{21} & 1 & u_{23} & & 0 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & & \ddots & 1 & u_{n-1,n} \\ 0 & \dots & \dots & 0 & 1 \end{pmatrix}$$

Je tedy potřeba vyjádřit koeficienty matice L a také matice U.

$$a_{11} = l_{11}$$

$$a_{i,i-1} = l_{i,i-1}, \quad \text{plati pro } i = 2, 3, \dots, n$$

$$a_{i,i} = l_{i,i-1}u_{i-1,i} + l_{i,i}, \quad \text{plati pro } i = 2, 3, \dots, n$$

$$a_{i,i+1} = l_{i,i}u_{i,i} + 1, \quad \text{plati pro } i = 1, 2, \dots, n-1$$

Z těchto uvedených vzorců si již snadno dopočteme všechny potřebné koeficienty do obou matic. Poté co vyjádříme matici A jako $A = LU$, poté systém $Ax = b$ převedeme na již známý systém $L\vec{y} = \vec{b}$, $L\vec{x} = \vec{y}$. Demonstrujme tuto metodu na následujícím příkladě.

$$\begin{aligned} 2x_1 - x_2 &= 4 \\ x_1 + 4x_2 + x_3 &= 5 \\ x_2 + 3x_3 - 2x_4 &= -1 \\ 2x_3 - 3x_4 &= \frac{7}{5} \end{aligned}$$

$$L = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 1 & \frac{9}{2} & 0 & 0 \\ 0 & 1 & \frac{25}{9} & 0 \\ 0 & 0 & 2 & -\frac{39}{25} \end{pmatrix} \quad U = \begin{pmatrix} 1 & -\frac{1}{2} & 0 & 0 \\ 0 & 1 & \frac{2}{9} & 0 \\ 0 & 0 & 1 & -\frac{18}{25} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Nyní vypočítáme vektor \vec{y} ze kterého následně zjistíme řešení této soustavy tedy vektor \vec{x} .

$$\begin{pmatrix} 2 & 0 & 0 & 0 \\ 1 & \frac{9}{2} & 0 & 0 \\ 0 & 1 & \frac{25}{9} & 0 \\ 0 & 0 & 2 & -\frac{39}{25} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix} = \begin{pmatrix} 4 \\ 5 \\ -1 \\ \frac{7}{5} \end{pmatrix}$$

$$\vec{y} \left(2, \frac{2}{3}, -\frac{3}{5}, -\frac{5}{3} \right)$$

$$\begin{pmatrix} 1 & -\frac{1}{2} & 0 & 0 \\ 0 & 1 & \frac{2}{9} & 0 \\ 0 & 0 & 1 & -\frac{18}{25} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 2 \\ 2 \\ \frac{3}{3} \\ -\frac{3}{5} \\ \frac{5}{5} \\ -\frac{5}{3} \end{pmatrix}$$

$$\vec{x}\left(\frac{38}{15}, \frac{16}{15}, -\frac{9}{5}, -\frac{5}{3}\right)$$

3 Nepřímé metody řešení systémů lineárních rovnic

Tyto metody nazýváme také metodami iteračními. Při aplikaci těchto metod nedostaneme přesná, ale aproximovaná řešení. Využívají se v případě, když máme matice velkých řádů, při kterých by výpočet přímými metodami byl velmi zdlouhavý a paměťově náročný.

3.1 Princip iteračních metod

Základním rysem iteračních metod je vyjádřit soustavu lineárních rovnic ve tvaru.

$$x = Tx + g, \quad T \in M_n$$

„ x^* je řešení systému $Ax = b$ právě tehdy, když x^* je řešením systému $x = Tx + g$, $x^* = (E - T)^{-1}g$ za předpokladu, že $E - T$ je regulární. Necht' $x^0 \in \mathbb{R}^n$ je libovolná počáteční aproximace. Posloupnost $\{x^k\}_{k=0}^\infty$ určená rekurentně vztahem

$$x^{k+1} = Tx^k + g, \quad k = 0, 1, \dots, n$$

se nazývá iterační posloupnost a matice T se nazývá iterační matice.“³

$$x^1 = Tx^0 + g,$$

$$x^2 = Tx^1 + g = T(Tx^0 + g) + g = T^2x^0 + (T + E)g,$$

$$x^{k+1} = T^{k+1}x^0 + (T^k + T^{k-1} + \dots + E)g$$

3.2 Jacobiova iterační metoda

Tvůrcem metody byl pruský matematik Carl Gustav Jacob Jacobi. Matici soustavy můžeme rozepsat do následného tvaru

$$A = D - L - U,$$

kde D je matice diagonální, L je dolní trojúhelníková matice, U horní trojúhelníková matice.

$$D = \begin{pmatrix} a_{11} & & 0 \\ & \ddots & \\ 0 & & a_{nn} \end{pmatrix},$$
$$L = \begin{pmatrix} 0 & & & 0 \\ a_{11} & \ddots & & \\ \vdots & \ddots & \ddots & \\ a_{n1} & \dots & a_{n,n-1} & 0 \end{pmatrix}, \quad U = \begin{pmatrix} 0 & a_{12} & \dots & 0 \\ & \ddots & \ddots & \vdots \\ & & \ddots & a_{n-1,n} \\ 0 & & & 0 \end{pmatrix}$$

³ (Horová, a další, 2008)

$$Ax = b \Rightarrow (D - L - U)x = b$$

Je potřeba tuto rovnici transformovat.

$$Dx = (L + U)x + b$$

Vycházíme z předpokladu, že jednotlivé prvky $a_{ii} \neq 0$, pro $i = 1, \dots, n$, je matice D regulární a lze tudíž vypočítat

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

a z tohoto vztahu získáme maticový tvar Jacobiovy iterační metody.

$$x^{k+1} = T_j x^k + D^{-1}b, \quad T_j = D^{-1}(L + U)$$

Lze také si také touto metodou zapsat vektor \vec{x} následovně.

$$x_i^{k+1} = - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{a_{ij}}{a_{ii}} x_j^k + \frac{b_i}{a_{ii}}, \quad \begin{matrix} i = 1, \dots, n \\ k \geq 0 \end{matrix}$$

„Pro některé speciální matice A je zaručena konvergence Jacobiovy iterační metody.

a) Silné řádkové sumační kritérium:

Nechť matice A je ryze řádkově diagonálně dominantní, tj.

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

Pak Jacobiova iterační metoda konverguje pro každou počáteční aproximaci $x^0 \in R^n$.

b) Silné sloupcové sumační kritérium:

Nechť matice A je ryze sloupcově diagonálně dominantní, tj.

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

Pak Jacobiova iterační metoda konverguje pro každou počáteční aproximaci $x^0 \in R^n$.⁴

⁴ (Horová, a další, 2008)

Příklad:

$$\begin{aligned} 11x_1 + 2x_2 + x_3 &= 15 \\ x_1 + 10x_2 + 2x_3 &= 16 \\ 2x_1 + 3x_2 - 8x_3 &= 1 \end{aligned}$$

Nyní z rovnice převedeme na iterační tvar, tak že z každé rovnice vyjádřím jednu neznámou.

$$x_1 = \frac{1}{11}(15 - 2x_2 - x_3)$$

$$x_2 = \frac{1}{10}(16 - x_1 - 2x_3)$$

$$x_3 = \frac{1}{8}(-1 + 2x_1 + 2x_2)$$

$$\begin{pmatrix} x_1^{k+1} \\ x_2^{k+1} \\ x_3^{k+1} \end{pmatrix} = \underbrace{\begin{pmatrix} 0 & -\frac{2}{11} & -\frac{1}{11} \\ -\frac{1}{10} & 0 & -\frac{1}{5} \\ \frac{1}{4} & \frac{3}{8} & 0 \end{pmatrix}}_{T_J} \begin{pmatrix} x_1^k \\ x_2^k \\ x_3^k \end{pmatrix} + \underbrace{\begin{pmatrix} \frac{15}{11} \\ \frac{16}{10} \\ -\frac{1}{8} \end{pmatrix}}_{D^{-1}b}$$

$$x_1^{k+1} = \frac{1}{11}(15 - 2x_2^k - x_3^k)$$

$$x_2^{k+1} = \frac{1}{10}(16 - x_1^k - 2x_3^k)$$

$$x_3^{k+1} = \frac{1}{8}(-1 + 2x_1^k + 2x_2^k)$$

Pokud si zvolíme počáteční stav $\vec{x}(0,0,0)$ a hodnotu konečného kritéria 10^4 , potom výsledek po průběhu deseti iterací je $\vec{x}(1.0564, 1.3642, 0.6507)$.

	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$ x^{(k)} - x^{(k-1)} $
0	0	0	0	
1	1,3636	1,6	-0,125	1,60000
2	1,0841	1,4886	0,8159	0,94090
3	1,0188	1,3284	0,7043	0,16020
4	1,0581	1,3573	0,6279	0,07640
5	1,0598	1,3686	0,6485	0,02060
6	1,0558	1,3643	0,6532	0,00470
7	1,0562	1,3638	0,6506	0,00260
8	1,0565	1,3643	0,6505	0,00050
9	1,0565	1,3643	0,6507	0,00020
10	1,0564	1,3642	0,6507	0,00010

Tabulka 1– Iterační výpočet SLR pomocí Jacobiovy metody

3.3 Gaussova-Seidelova iterační metoda

Metoda je velmi podobná předešlé Jacobiově metodě, s tím rozdílem, že pro x_i^{k+1} , $i > 1$, počítá již s vyčíslenými koeficienty $x_i^{k+1}, \dots, x_{i-1}^{k+1}$.

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

Maticový zápis se dá zapsat takto.

$$A x = b \Rightarrow (D - L - U)x = b$$

$$(D - L)x = U x + b$$

Za předpokladu, že $a_{ii} \neq 0, i = 1, \dots, n$, je matice $D - L$ regulární.

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

Potom je Gaussova-Seidelova iterační metoda tvaru.

$$x^{k+1} = T_G x^k + g, \quad T_G = (D - L)^{-1} U, \quad g = (D - L)^{-1} b$$

„Posloupnost $\{x^k\}_{k=0}^{\infty}$ generovaná Gaussovou-Seidelovou iterační metodou konverguje pro každou počáteční aproximaci $x^0 \in R^n$ právě tehdy, když $\rho(T_G) < 1$.“⁵

Metodu provedeme na stejném příkladě jako Jacobiovou, opět s počátečními podmínkami $\vec{x}(0,0,0)$ a hodnotu konečného kritéria 10^4 .

$$\begin{aligned} 11x_1 + 2x_2 + x_3 &= 15 \\ x_1 + 10x_2 + 2x_3 &= 16 \\ 2x_1 + 3x_2 - 8x_3 &= 1 \end{aligned}$$

$$x_1^{k+1} = \frac{1}{11}(15 - 2x_2^k - x_3^k)$$

$$x_2^{k+1} = \frac{1}{10}(16 - x_1^{k+1} - 2x_3^k)$$

$$x_3^{k+1} = \frac{1}{8}(-1 + 2x_1^{k+1} + 2x_2^{k+1})$$

	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$	$ x^{(k)} - x^{(k-1)} $
0	0	0	0	
1	1,3636	1,4636	0,7648	1,46360
2	1,028	1,3442	0,6361	0,33560
3	1,0614	1,3666	0,6528	0,03340
4	1,0558	1,2639	0,6504	0,10270
5	1,0565	1,3643	0,6507	0,10040
6	1,0564	1,3642	0,6507	0,00011
7	1,0564	1,3642	0,6507	0,00001

Tabulka 2– Iterační výpočet SLR pomocí G-S metody

Při porovnání obou tabulek výsledků dojdeme k závěru, že výsledek z desáté iterace Jacobiovou a výsledek sedmé iterace G-S metodou se rovnají. Stačilo nám k tomu o tři iterace méně, díky použití již vypočítaných hodnot.

$$\vec{x}(1.0564, 1.3642, 0.6507).$$

⁵ (Horová, a další, 2008)

3.4 Ukončovací kritéria iteračních metod

Ukončovacím kritériem je myšlen předpis, kterým získáme reálné číslo. Toto číslo následně porovnáme s hodnotou tolerance odchylky dvou posledních iterací. Pokud je toto číslo větší než tolerance, pokračuje výpočet další iterací. V opačném případě je iterační proces ukončen a výsledek systému rovnic je poslední iterace.

„Nechť C^n resp. R^n je vektorový prostor všech uspořádaných n -tic komplexních resp. reálných čísel. Prvky tohoto prostoru budeme zapisovat ve sloupcových vektorech.

Vektorová norma C^n je funkce $\|\cdot\|$ (z C^n do R^n) s následujícími vlastnostmi:

- 1) $\|x\| > 0, \forall x \in C^n$
- 2) $\|x\| = 0 \Leftrightarrow x = o, o = (0, \dots, 0)^T$
- 3) $\|\alpha x\| = |\alpha| \|x\|, \forall \alpha \in C, \forall x \in C^n$
- 4) $\|x + y\| \leq \|x\| + \|y\|, \forall x, y \in C^n$ ⁶

Norma vektoru udává velikost vektoru \vec{x} , kterou získáme $\|\vec{x}\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$.

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

Tímto předpisem se označuje absolutní chyba, která bere rozdíl vektorů posledních dvou iterací, ze kterých zjistíme normu vektoru a tu porovnáme s ε , což je hodnota tolerance.

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

Relativní chyba se počítá velmi podobně jako absolutní chyba. Rozdíl je v tom, že ještě normu vektoru $\|x^{k+1} - x^k\|$ vydělíme normou vektoru $\|x^k\|$.

$$\|r^{k+1}\| = Ax^{k+1} - b < \varepsilon$$

Residuum se získá z $Ax^{k+1} - b$, které musí pro ukončení iteračního výpočtu menší než tolerance.

⁶ (Horová, a další, 2008)

4 Algoritmy pro výpočet iteračních metod v Matlabu

Pro řešení iteračních metod v Matlabu je zapotřebí mít čtvercovou matici. Metody jsou navrženy pro řešení systému $Ax = b$, nebo pro minimalizaci výrazu $\|b - Ax\|$. Algoritmy, které Matlab používá k řešení systému lineárních rovnic iterační metodou, jsou:

BICG – Biconjugate gradient method (metoda bi-sdružených gradientů)

BICGSTAB – Biconjugate gradient stabilized method (stabilizovaná metoda bi-sdružených gradientů).

CGS – Conjugate gradients square method (čtvercová metoda sdružených gradientů)

GMRES – Generalized minimum residual method (zobecněná metoda minimálního rezidua)

QMR – Quasi-minimal residual method (metoda částečně minimálního rezidua).

Všechny metody využívají předpoklady M. Systém lineárních rovnic $Ax = b$ je pak zaměněn za ekvivalentní systém $M^{-1}Ax = M^{-1}b$. Předpoklad M je zvolen pro zrychlení konvergence iteračních metod. Většina iteračních metod si při řešení iteračních rovnic používá funkci \ tzv. backslash.

$A \setminus B$ je matice podílu A a B, který je zhruba stejný jako $\text{inv}(A)*B$, ale je počítán jiným postupem. Jestliže A je matice typu $n \times n$ a B je sloupcový vektor o n složkách, nebo matice složená z několika takových sloupců, pak $X = A \setminus B$ je řešením rovnice $A*X=B$ Gaussovou eliminací. Varovná zpráva se zobrazí, je-li A špatného typu, nebo je-li A skoro singulární. Jestliže A je matice typu $m \times n$ a B je sloupcový vektor o m složkách, nebo matice složená z takových sloupců, pak $X = A \setminus B$ je řešením systému $A*X=B$ ve smyslu metody nejmenších čtverců.

4.1 BICG – Metoda bi-sdružených gradientů

$X = \text{BICG}(A, B)$

Pokouší se řešit systém lineárních rovnic $A*X = B$ pro X. Matice koeficientů A musí být čtvercová typu $n \times n$ a sloupcový vektor musí být délky n. BICG začíná iterační proces v počátečním odhadu, který je standardně nulový vektor délky n. Iterace se opakují, dokud metoda nekonverguje, nebo dokud není spočítán maximální počet iterací. Konvergence je dosaženo, jestliže relativní reziduum $\text{NORM}(B-A*X)/\text{NORM}(B)$ je menší nebo rovno toleranci metody. Tolerance této metody je $1e-6$. Standardní počet iterací je od n do 20. Ne

$\text{BICG}(A, B, \text{TOL})$ určuje toleranci metody.

$\text{BICG}(A, B, \text{TOL}, \text{MAXIT})$ určuje maximální počet iterací.

BICG(A, B, TOL, MAXIT, M) a BICG(A, B, TOL, MAXIT, M1, M2) používají předpoklad M nebo $M = M1 * M2$ a efektivně řeší systém rovnic $\text{inv}(M)*A*X = \text{inv}(M)*B$ pro X .

BICG(A, B, TOL, MAXIT, M1, M2, X0) specifikuje počáteční odhad. Jestliže $X0$ je zadán jako prázdná matice, pak je standardně použit nulový vektor.

$X = \text{BICG}(A, B, TOL, MAXIT, M1, M2, X0)$ vrací řešení X . Jestliže BICG konverguje, pak se zobrazí informační zpráva. Jestliže u BICG selhala konvergence po maximálním množství iterací nebo se BICG zastavil z jakéhokoliv důvodu, pak se zobrazí varovná zpráva informující o relativním reziduu a o čísle iterace, při kterém se metoda zastavila či selhala.

4.2 BICGSTAB – Stabilizovaná metoda bi-sdružených gradientů

Funkce BICGSTAB používá stejnou syntaxi jako funkce BICG.

$X = \text{BICGSTAB}(A, B)$

$\text{BICGSTAB}(A, B, TOL)$

$\text{BICGSTAB}(A, B, TOL, MAXIT)$

$\text{BICGSTAB}(A, B, TOL, MAXIT, M)$

$\text{BICGSTAB}(A, B, TOL, MAXIT, M1, M2)$

$\text{BICGSTAB}(A, B, TOL, MAXIT, M1, M2, X0)$

$X = \text{BICGSTAB}(A, B, TOL, MAXIT, M1, M2, X0)$

4.3 CGS – Čtvercová metoda sdružených gradientů

Využívá stejnou syntaxi jako BICG.

$X = \text{CGS}(A, B)$

$\text{CGS}(A, B, TOL)$

$\text{CGS}(A, B, TOL, MAXIT)$

$\text{CGS}(A, B, TOL, MAXIT, M)$

$\text{CGS}(A, B, TOL, MAXIT, M1, M2)$

$\text{CGS}(A, B, TOL, MAXIT, M1, M2, X0)$

$X = \text{CGS}(A, B, \text{TOL}, \text{MAXIT}, M1, M2, X0)$

4.4 GMRES – Zobecněná metoda minimálního rezidua

$X = \text{GMRES}(A, B)$ a $X = \text{GMRES}(A, B, \text{RESTART})$ řeší systém lineárních rovnic $A * X = B$ pro X . A musí být čtvercová matice typu $n \times n$ a sloupcový vektor B musí být délky n . Na rozdíl od BICG, se GMRES sám restartuje při dosažení iterace RESTART, kterou použije jako nový počáteční odhad. Standardní hodnota restart je n . Konvergence je dosaženo, jestliže relativní reziduum $\text{NORM}(B - A * X) / \text{NORM}(B)$ je menší nebo rovno toleranci metody. Tolerance této metody je $1e-6$. Standardní maximální počet iterací je menší z čísel $n / \text{RESTART}$ a 10.

$\text{GMRES}(A, B, \text{RESTART}, \text{TOL})$ určuje toleranci metody.

$\text{GMRES}(A, B, \text{RESTART}, \text{TOL}, \text{MAXIT})$ určuje maximální počet iterací.

$\text{GMRES}(A, B, \text{RESTART}, \text{TOL}, \text{MAXIT}, M)$ a $\text{GMRES}(A, B, \text{RESTART}, \text{TOL}, \text{MAXIT}, M1, M2)$ využívá předpoklad M nebo $M = M1 * M2$ a efektivně řeší systém rovnic $\text{inv}(M) * A * X = \text{inv}(M) * b$ pro X .

$\text{GMRES}(A, B, \text{RESTART}, \text{TOL}, \text{MAXIT}, M1, M2, X0)$ specifikuje počáteční odhad.

$X = \text{GMRES}(A, B, \text{RESTART}, \text{TOL}, \text{MAXIT}, M1, M2, X0)$ vrací řešení X . Jestliže GMRES konverguje, pak se zobrazí informační zpráva. Jestliže u GMRES selhala konvergence po maximálním množství iterací nebo se GMRES zastavil z jakéhokoliv důvodu, pak se zobrazí varovná zpráva informující o relativním reziduu a o čísle iterace, při kterém se metoda zastavila či selhala.

4.5 QMR – Metoda částečně minimálního rezidua

Syntaxe příkazů je stejná jako u funkce BICG s výjimkou $\text{QMR}(A, B, \text{TOL}, \text{MATIX}, M1)$ a $\text{QMR}(A, B, \text{TOL}, \text{MAXIT}, M1, M2)$. U těchto příkazů je použito levého předpokladu $M1$ a pravého předpokladu $M2$, řešení soustavy $A * X = B$ se tedy zefektivní po převodu na ekvivalentní soustavu $\text{inv}(M1) * A * \text{inv}(M2) * Y = \text{inv}(M1) * B$, kde $X = \text{inv}(M1) * Y$.

V dolní části aplikace je editor pane, který se plní výsledky iteračních cyklů a label ve kterém se zobrazuje čas výpočtu.

5.2 Popis použitých algoritmů

Metoda matPlusMat si alokuje dvourozměrné pole double, které naplní součtem prvků na odpovídajících si pozicích vstupních matic. Nakonec funkce vrací pole, které je výsledkem součtu matic.

Metoda matXmat vrací dvourozměrné pole, které si alokovalo v těle funkce a naplnilo hodnotami odpovídajícími součinu matic.

Metoda matXvekt vrací jednorozměrné pole odpovídající součinu matice a vektoru.

Metoda vektMinusVekt si alokuje jednorozměrné pole double. Naplní ho rozdíly prvků na odpovídajících si pozicích vektorů, vstupujících do metody v argumentu. Nakonec pole vrátí výsledný vektor.

Metoda inverse zapíše do reference, která vstupuje do funkce jako druhý argument inverzní matici k matici, která je prvním argumentem funkce.

Metody kriterium1, kriterium2 a kriterium3 vypočítávají hodnotu podle zvoleného kritéria. Kriterium1 vrací reálné číslo double, které vypočte podle předpisu pro absolutní chybu. Kriterium2 vrací číslo podle předpisu relativní chyby a kriterium3 podle předpisu residua.

Metoda Jacobiho provádí iterační výpočet na základě vstupních argumentů. Provádí jej na základě následující rovnice.

$$x_i^{k+1} = - \sum_{\substack{j=1 \\ j \neq i}}^n \frac{a_{ij}}{a_{ii}} x_j^k + \frac{b_i}{a_{ii}}, \quad \begin{array}{l} i = 1, \dots, n \\ k \geq 0 \end{array}$$

Po provedení sérii výpočtu se funkcí pro výpočet kritéria přiřadí hodnota tolerance aktuální iterace a porovná se s tolerancí, kterou zvolil uživatel. Po ukončení iteračního výpočtu se vrací matice všech iteračních hodnot neznámých x.

Metoda JacobiMaticove provádí iterační výpočet stejně jako metoda Jacobiho, jen za použití rozdílné rovnice $x^{k+1} = T_j x^k + D^{-1}b$, $T_j = D^{-1}(L + U)$. Vrací stejně jako metoda předešlá matici všech iteračních hodnot neznámých x

Metoda Gauss_Seidelova provádí iterační výpočet podle následující rovnice.

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

Stejně jako předešlé metody pro výpočet iteračních metod, také tato metoda každé iteraci zjišťuje hodnotu tolerance v jednotlivých iteračních cyklech. Po konci výpočtu vrátí matici iteračních hodnot všech výsledků x .

Metoda Gauss_SeidelMaticove analogicky k Jacobiovým se chová stejně jako metoda Gauss_Seidelova, jen s rozdílem použití jiné rovnice pro výpočet.

$$x^{k+1} = T_G x^k + g, \quad T_G = (D - L)^{-1}U, \quad g = (D - L)^{-1}b$$

6 Srovnání s matematickým softwarem Matlab

Matlab je matematický software vyvíjen společností MathWorks. Název Matlab vznikl zkrácením slov matrix laboratoř, což lze volně přeložit jako maticová laboratoř. Tento překlad plně vystihuje strukturu, která je při výpočtech založena na maticích. „Matlab umožňuje počítání s maticemi, vykreslování 2D a 3D grafů funkcí, implementaci algoritmů, počítačovou simulaci, analýzu a prezentaci dat i vytváření aplikací včetně grafického rozhraní“⁷

6.1 Srovnání časových nároků při výpočtu

Srovnání autorem vytvořenou aplikací a matematickým softwarem Matlab se provede na deseti pokusech výpočtu stejné soustavy lineární rovnice.

$$\begin{aligned}11x_1 + 2x_2 + x_3 &= 15 \\x_1 + 10x_2 + 2x_3 &= 16 \\2x_1 + 3x_2 - 8x_3 &= 1\end{aligned}$$

```
>> tic;
A = [11 2 1;1 10 2;2 3 -8];
B = [15;16;1];
D = diag(diag(A));
L = tril(A,-1);
P = triu(A,1);
UJ = -inv(D) * (L+P);
VJ = inv(D) * B;
X = [0;0;0];
for n=1:10
X=UJ * X + VJ;
end;
toc
Elapsed time is 0.011666 seconds.
```

Obrázek 3 Kód pro výpočet Jacobiovy metody v Matlabu

⁷ <http://cs.wikipedia.org/wiki/MATLAB>


```

>> tic;
A = [11 2 1;1 10 2;2 3 -8];
B = [15;16;1];
D = diag(diag(A));
L = tril(A,-1);
P = triu(A,1);
UG = -inv(D+L) * P;
VG = inv(D+L) * B;
X = [0;0;0];
for n=1:7
X=UG * X + VG;
end;
toc
Elapsed time is 0.014773 seconds.

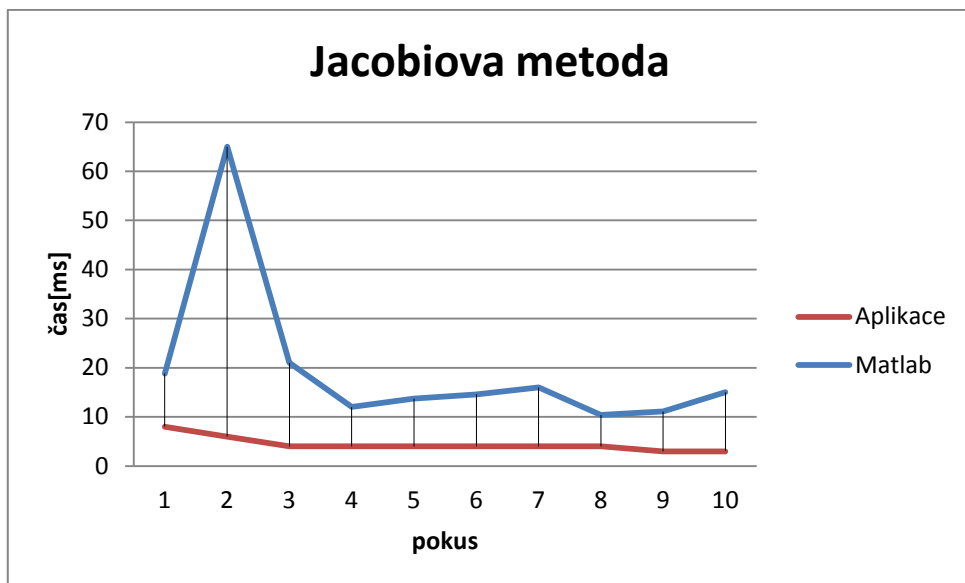
```

Obrázek 4 Kód pro výpočet G-S metody v Matlabu

Tyto kódy byly zvoleny, aby se časová náročnost měřila na přesně stejných úkonech, které se vykonají ve zdrojovém kódu praktické části bakalářské práce.

	Aplikace(ms)	Matlab(ms)
1	8	18,83
2	6	65
3	4	21
4	4	12
5	4	13,7
6	4	14,6
7	4	16
8	4	10,4
9	3	11,1
10	3	15
průměr	4,4	19,763

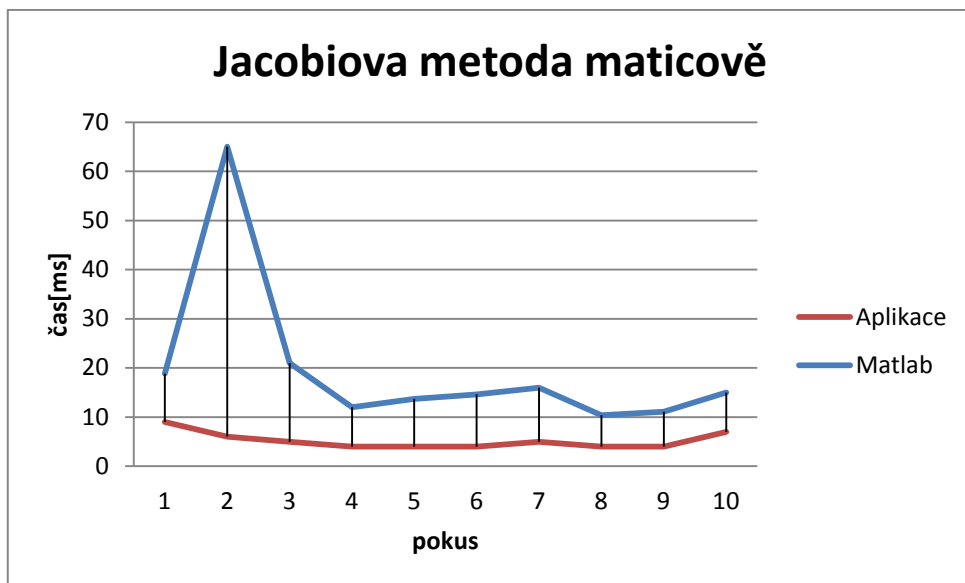
Tabulka 3 Srovnání časové náročnosti Jacobiovy metody



Graf 1 Srovnání časové náročnosti Jacobiovy metody

	Aplikace(ms)	Matlab(ms)
1	9	18,83
2	6	65
3	5	21
4	4	12
5	4	13,7
6	4	14,6
7	5	16
8	4	10,4
9	4	11,1
10	7	15
průměr	5,2	19,763

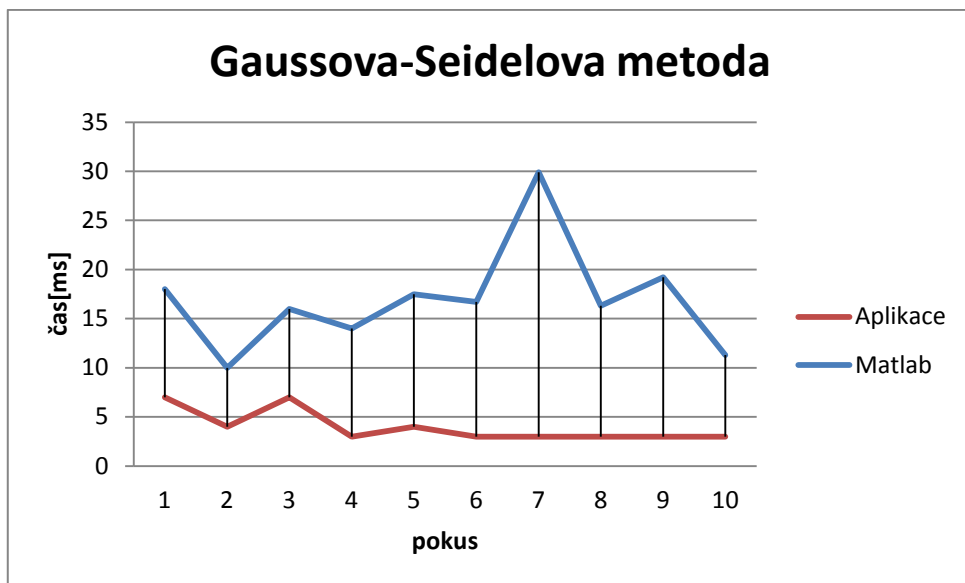
Tabulka 4 Srovnání časové náročnosti Jacobiovy metody maticovým způsobem



Graf 2 Srovnání časové náročnosti Jacobiovy metody maticovým způsobem

	Aplikace(ms)	Matlab(ms)
1	7	18
2	4	10
3	7	16
4	3	14
5	4	17,5
6	3	16,7
7	3	29,9
8	3	16,3
9	3	19,2
10	3	11,3
průměr	4	16,89

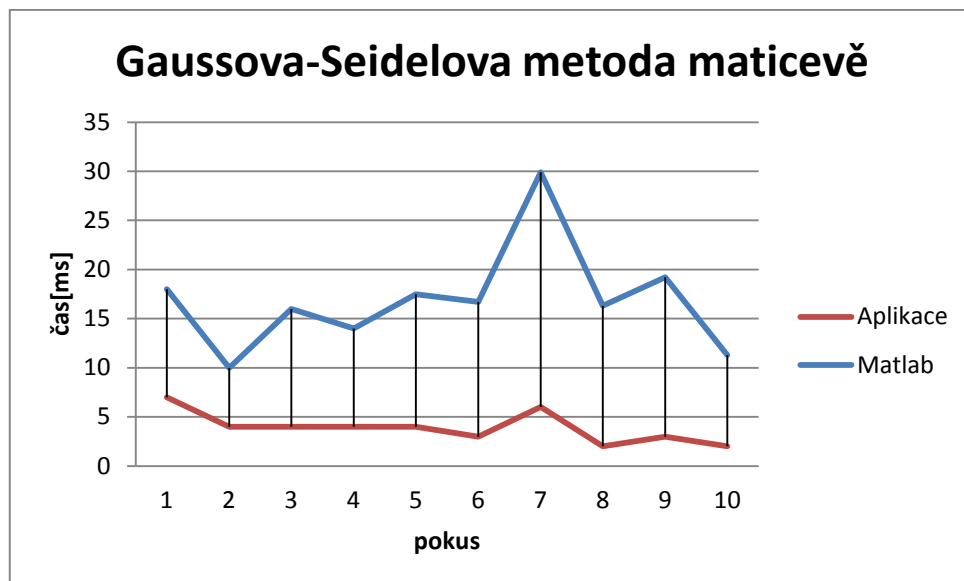
Tabulka 5 Srovnání časové náročnosti G-S metody



Graf 3 Srovnání časové náročnosti G-S metody

	Aplikace(ms)	Matlab(ms)
1	7	18
2	4	10
3	4	16
4	4	14
5	4	17,5
6	3	16,7
7	6	29,9
8	2	16,3
9	3	19,2
10	2	11,3
průměr	3,9	16,89

Tabulka 6 Srovnání časové náročnosti G-S metody maticovým způsobem



Graf 4 Srovnání časové náročnosti G-S metody maticovým způsobem

6.2 Srovnání přesnosti výsledků

Přesnost výsledků se provede na totožné matici jako u srovnávání časových nároků na výpočet.

$$\begin{aligned} 11 x_1 + 2 x_2 + x_3 &= 15 \\ x_1 + 10 x_2 + 2 x_3 &= 16 \\ 2 x_1 + 3 x_2 - 8 x_3 &= 1 \end{aligned}$$

	Aplikace	Matlab
x_1	1,0564	1,0564
x_2	1,3642	1,3642
x_3	0,6507	0,6507

Tabulka 7 Srovnání přesnosti výsledků Jacobiovy metody

	Aplikace	Matlab
x_1	1,0564	1,0564
x_2	1,3642	1,3642
x_3	0,6507	0,6507

Tabulka 8 Srovnání přesnosti výsledků G-S metody

Z předchozích tabulek 7 a 8 je zřejmé, že přesnost výsledků praktické části bakalářské práce zaokrouhlených na 4 desetinná místa, je shodná u Jacobiovy metody po 10 iteracích a G-S metoda po 7 iteracích s výsledky získané z Matlabu.

Závěr

Teoretická část bakalářské práce má za úkol seznámit čtenáře s problematikou řešení systémů lineárních rovnic, jak přímými metodami, tak i iteračními metodami. Při psaní byla snaha popisovat metody matematickými a přesnými definicemi, ale také je popisovat jednoduchým a snadno pochopitelným způsobem. K podtržení snadnosti pochopení se názorně každá metoda ukázala na typových příkladech s jasnými instrukcemi dalších úkonů. Dále zde srovnáváme výslednou aplikaci, navrženou a realizovanou v praktické části bakalářské práce, se softwarem pro vědecké výpočty Matlabem. Příjemným zjištěním bylo, když časové nároky na dobu výpočtu a všech úkonů, které se dějí při výpočtu v praktické části bakalářské práce, jsou rychlejší než ty samé úkony ve zmiňovaném Matlabu. Při srovnávání přesnosti výsledků došlo ke konstatování, že přesnost jednotlivých kořenů soustavy lineárních rovnic je naprosto stejná jako v Matlabu.

V praktické části byly následně implementovány algoritmy, které byly popsány v teoretické části. Pro aplikaci byl zvolen programovací jazyk Java, který je považován za jeden z nejpopulárnějších programovacích jazyků na světě pro svou přenositelnost mezi různými platformami.

Literatura

- 2011.** Centrum aplikované matematiky. *Západočeská univerzita v Plzni*. [Online] 2011. [Citace: 6. 5 2012.] http://www.cam.zcu.cz/~danek/Students/2006_LS/soubory/prednasky/NM_prednaska_03.pdf.
- 2011.** Centrum aplikované matematiky. *Západočeská univerzita v Plzni*. [Online] 2011. [Citace: 26. 4 2012.] http://www.cam.zcu.cz/~danek/Students/2006_LS/soubory/prednasky/NM_prednaska_03.pdf.
- Dušek, František. 2000.** *MATLAB a SIMULINK úvod do používání*. Pardubice : Univerzita Pardubice, 2000. str. 146. 80-7194-273-1.
- 2005.** Fakulta aplikovaných věd. *Západočeská univerzita v Plzni*. [Online] 2005. [Citace: 16. 4 2012.] <http://dimatia.fav.zcu.cz/2005/vyuka/zm1/soustavy.pdf>.
- Felcman, Jiří. 2012.** 4.4 LU rozklad v obecném případě for nm. *Scribd*. [Online] 2012. [Citace: 16. 4 2012.] <http://www.scribd.com/doc/53207894/21/LU-rozklad-v-obecnem-p%C5%99ipad%C4%9B>.
- 2011.** Gaussova eliminační metoda - Gaussova eliminace. *Aristoteles.Cz - matematika, chemie a fyzika online*. [Online] 21. 11 2011. [Citace: 2. 5 2012.] http://www.aristoteles.cz/matematika/linearni_algebra/soustavy/gaussova-eliminacni-metoda.php.
- Horová, Ivana a Zelenka, Jiří. 2008.** *Numerické metody*. Brno : Masarykova univerzita, 2008. str. 299. 978-80-210-3317-7.
- Kolářek, Jan. 2012.** Jan Kolářek. *Masarykova univerzita*. [Online] 2012. [Citace: 15. 4 2012.] http://www.math.muni.cz/~kolacek/vyuka/nummet/prime_metody.pdf.
- 2006.** Matematická sekce MFF UK. *Univerzita Karlova - UK*. [Online] 3. 10 2006. [Citace: 19. 4 2012.] <http://www.karlin.mff.cuni.cz/~tuma/2002/NLinalg2.pdf>.
- 2001.** MATLAB - Wikipedie. *Wikipedie, otevřená encyklopedie*. [Online] 2001. [Citace: 1. 5 2012.] <http://cs.wikipedia.org/wiki/MATLAB>.
- Rod, Michal. 2008.** Michal Rod. *ČVUT v Praze*. [Online] 2008. [Citace: 25. 4 2012.] <http://rod.aspone.cz/files/LU.pdf>.
- 2001.** Soustava lineárních rovnic - Wikipedie. *Wikipedie, otevřená encyklopedie*. [Online] 2001. [Citace: 29. 4 2012.] http://cs.wikipedia.org/wiki/Soustava_line%C3%A1rn%C3%ADch_rovnic.

The MathWorks, Inc. 2012. Mathematics :: Function Reference (MATLAB®).
MathWorks - MATLAB and Simulink for Technical Computing. [Online] 2012. [Citace: 6.
5 2012.] <http://www.mathworks.com/help/techdoc/ref/f16-5872.html#f16-6565>.

Vitásek, Emil. 1987. *Numerické metody*. Praha : SNTL-Nakladatelsví technické literatury,
1987. str. 516. 04-009-87.

Příloha A – Dokumentace

Dokumentace k Aplikaci je přiložena na CD.

Příloha B – Zdrojové kódy

Zdrojové kódy Aplikaci jsou přiloženy na CD.