



**MATEMATICKO-FYZIKÁLNÍ
FAKULTA**
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Phuong Thao Hoang

Neúplná Choleského faktorizace

Katedra numerické matematiky

Vedoucí bakalářské práce: Prof. Ing. Miroslav Tůma, CSc.

Studijní program: Matematika

Studijní obor: Obecná matematika

Praha 2018

Prohlašuji, že jsem tuto bakalářskou práci vypracoval(a) samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V dne

Podpis autora

Chtěla bych poděkovat svému vedoucímu bakalářské práce Prof. Ing. Miroslavovi Tůmovi, CSc. za odborné vedení, za pomoc a rady při zpracování této práce.

Název práce: Neúplná Choleského faktorizace

Autor: Phuong Thao Hoang

Katedra: Katedra numerické matematiky

Vedoucí bakalářské práce: Prof. Ing. Miroslav Tůma, CSc., Katedra numerické matematiky

Abstrakt: Práce se zabývá neúplnou Choleského faktorizací a jejími variantami, které mají velký význam pro předpodmiňování úloh se symetrickou a pozitivně definitní maticí. Zde se soustředíme především na řešení těchto velmi rozsáhlých soustav s řídkými maticemi, které vznikají v mnoha technických a přírodovědných oborech, pomocí předpodmíněných sdružených gradientů. Kromě dalších postupů můžeme na soustavu aplikovat Choleského faktorizaci přibližně, neúplně. V této práci studujeme existenci této faktorizace a chování a potenciál různých variant základního algoritmu.

Klíčová slova: řešení soustav lineárních rovnic, neúplné faktorizace, předpodmiňování, Choleského rozklad, metoda sdružených gradientů

Title: Incomplete Cholesky factorization

Author: Phuong Thao Hoang

Department: Department of Numerical Mathematics

Supervisor: Prof. Ing. Miroslav Tůma, CSc., Department of Numerical Mathematics

Abstract: The thesis is about the incomplete Cholesky factorization and its variants, which are important for preconditioning a system with symmetric and positive definite matrix. Our main focus is on solving these systems, which arise in many technical applications and natural sciences, using preconditioned Conjugate Gradients. Besides many other ways we can apply Cholesky factorization approximately, incompletely. In this thesis we study existence of the incomplete Cholesky factorization and we evaluate behaviour and potential of different variants of the generic algorithm.

Keywords: solving systems of linear equations, incomplete decompositions, preconditioning, Cholesky factorization, Conjugate Gradient Method

Obsah

Úvod	2
1 Choleského faktorizace	3
1.1 Husté matice	3
1.1.1 Způsoby faktorizace	3
1.1.2 Řešení soustavy	6
1.2 Řídké matice	7
2 Přímé metody vs iterační metody	9
3 Metoda sdružených gradientů	10
3.1 Odvození algoritmu	10
3.2 Konvergence CG	12
4 Předpodmíněné iterační metody	13
5 Neúplná Choleského faktorizace	15
5.1 Existence	15
5.2 Základní algoritmy	19
5.2.1 Klasická neúplná Choleského faktorizace	19
5.2.2 Modifikovaná neúplná Choleského faktorizace	20
5.3 Varianty	22
5.3.1 Podle struktury	22
5.3.2 Podle velikosti prvků	23
5.3.3 Současné praktické strategie	23
5.4 Nová varianta	25
5.4.1 Popis nové varianty	25
5.4.2 Výsledky zkoumání	27
Závěr	30
Literatura	32
Seznam obrázků	34
Seznam tabulek	35

Úvod

Iterační metody nejsou samy o sobě příliš spolehlivé ve smyslu konvergence, proto se dříve pro řešení velkých soustav lineárních algebraických rovnic často nevyužívali. K významné změně názoru na jejich praktickou použitelnost došlo po objevení předpodmínění a v souvislosti s rozvojem výpočetní techniky. Jedna z nejpoužívanějších metod předpodmínění pro úlohy s hermitovskými a pozitivně definitními maticemi je náplní této bakalářské práce. Neúplná Choleského faktorizace se používá zejména k předpodmínění metody sdružených gradientů, což je iterační metoda, která řeší takovýto typ úloh. V práci se seznámíme s uvedenými pojmy a blíže nahlédneme do problematiky neúplné Choleského faktorizace a jejích variant. Omezíme se přitom na reálné symetrické a pozitivně definitní matice.

Práce je strukturována následujícím způsobem. První kapitola je věnována standardní Choleského faktorizaci, způsobům jejího výpočtu a aplikací na soustavu lineárních algebraických rovnic s hustou a řídkou maticí. U řídkých matic jsou vysvětleny problémy, které mohou v průběhu výpočtu nastat, zejména zaplnění faktoru L . V druhé kapitole srovnáváme přímé a iterační metody, vystihujeme jejich klady a zápory. V další kapitole pokračujeme představením jedné z nejpoužívanějších iteračních metod - metody sdružených gradientů. Po odvození algoritmu sdružených gradientů stručně zmíníme známá fakta o jeho konvergenci a na konec uvedeme téma předpodmínění, které je náplní čtvrté kapitoly. Zde si vysvětlíme hlavní ideu předpodmínění, jeho potenciál pro iterační metody a způsoby jeho aplikace. Jako příklad je pak uvedena neúplná LU faktorizace (ILU) a její speciální případ, neúplná Choleského faktorizace (IC). Pátá kapitola tvoří jádro práce. Nejprve se zabýváme samotnou existencí IC, která vyplyne z důkazu věty o existenci ILU, k níž bude za potřebí zavést několik definic a pomocných vět. Dozvíme se tak, že existenci IC lze zatím dokázat jen pro H -matice a že u obecnějších matic může dojít k tzv. zlomu. Uvedeme pár návrhů, jak tomuto problému zamezit. Dále pokračujeme představením dvou základních algoritmů IC, ze kterých jsou posléze odvozeny jejich různé varianty. Ty si rozdělíme v další sekci do třech tříd podle jejich charakteristik a uvedeme si v každé třídě alespoň jednu nejpoužívanější variantu, kde zároveň ohodnotíme její potenciál. Na závěr navrhneme novou variantu IC, jež se snaží o nový pohled na problém hledání optimální matice předpodmínění pro CG. Popíšeme si její navržení a použijeme ji pro pár vybraných řídkých matic. Výsledky zkoumání poté vyhodnotíme a stručně shrneme v závěru práce.

Jelikož je práce zaměřena především na představení neúplné Choleského faktorizace a jejích variant, zvědavý čtenář je odkázán k bližšímu seznámení s některými problémy, které jsou v práci jen stručně zmíněny, k další literatuře.

1. Choleského faktorizace

1.1 Husté matice

Choleského faktorizace je varianta Gaussovy eliminace určená speciálně pro symetrické a pozitivně definitní matice (SPD), která danou matici rozloží na součin dvou matic, kde jedna je dolní trojúhelníková s kladnými prvky na diagonále a druhá je k ní transponovaná matice, tj.

$$A = LL^T. \quad (1.1)$$

Věta 1. *Nechť $A \in \mathbb{R}^{n \times n}$ je symetrická a pozitivně definitní matice. Pak existuje právě jedna dolní trojúhelníková matice L s kladnými prvky na diagonále taková, že $A = LL^T$.*

Přestože matice L je jednoznačně určena dle Věty 1, existuje více variant algoritmů, kterými můžeme L vypočítat. Tyto varianty se liší především způsobem výpočtu a vzájemným pořadím, v jakém prvky matice L získáváme.

1.1.1 Způsoby faktorizace

První variantu, kterou uvedeme, je varianta sloupcová. Zde v každém kroku počítáme právě jeden sloupec matice L , a k tomu využíváme původního sloupce matice A a kombinace již vypočítaných sloupců matice L . Schéma algoritmu vypadá takto (používáme standardní algoritmické značení programu MATLAB)

1.

$$w_{j:n} = A_{j:n,j} - \sum_{k=1}^{j-1} L_{j,k} L_{j:n,k}$$

2.

$$L_{j,j} = \sqrt{w_j}$$

3.

$$L_{j+1:n,j} = \frac{1}{\sqrt{w_j}} w_{j+1:n}.$$

Druhou variantou je výpočet po řádcích. Uvažujme blokové rozdělení matice A tvaru

$$A = \begin{pmatrix} M & u \\ u^T & s \end{pmatrix}, \quad (1.2)$$

kde $M \in \mathbb{R}^{(n-1) \times (n-1)}$, $u \in \mathbb{R}^{n-1}$ a $s \in \mathbb{R}$ je kladné číslo. Nejprve si ukážeme, že platí následující lemma.

Lemma 2. *Nechť $A \in \mathbb{R}^{n \times n}$ je symetrická a pozitivně definitní matice a buď (1.2) její blokové rozdělení. Pak submatice M je také SPD.*

Důkaz. Buď $x \in \mathbb{R}^{n-1}$ libovolný vektor. Protože A je SPD, tak platí

$$\begin{aligned}
0 &< \begin{pmatrix} x^T & -\frac{x^T u}{s} \end{pmatrix} \begin{pmatrix} M & u \\ u^T & s \end{pmatrix} \begin{pmatrix} x \\ -\frac{x^T u}{s} \end{pmatrix} \\
&= \begin{pmatrix} x^T & -\frac{x^T u}{s} \end{pmatrix} \begin{pmatrix} Mx - \frac{ux^T u}{s} \\ u^T x - x^T u \end{pmatrix} \\
&= \begin{pmatrix} x^T & -\frac{x^T u}{s} \end{pmatrix} \begin{pmatrix} Mx - \frac{uu^T x}{s} \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} x^T Mx - x^T \frac{uu^T}{s} x \end{pmatrix} \\
&= x^T \left(M - \frac{uu^T}{s} \right) x.
\end{aligned}$$

My však chceme $x^T Mx > 0$. To jde ale snadno odvodit, neb $\frac{uu^T}{s}$ je zřejmě nezáporné reálné číslo, tedy

$$0 < x^T \left(M - \frac{uu^T}{s} \right) x = x^T Mx - x^T \frac{uu^T}{s} x = x^T Mx - \frac{uu^T}{s} x^T x \leq x^T Mx$$

□

Dle Věty 1 existuje dolní trojúhelníková matice L_M taková, že

$$M = L_M L_M^T.$$

Odsud

$$A = \begin{pmatrix} L_M & 0 \\ w^T & t \end{pmatrix} \begin{pmatrix} L_M^T & w \\ 0 & t \end{pmatrix}, \quad (1.3)$$

kde $w = L_M^{-1}u$ a $t = (s - w^T w)^{1/2}$. Označme $L = \begin{pmatrix} L_M & 0 \\ w^T & t \end{pmatrix}$, pak $A = LL^T$.

Je jasné, že matici L_M lze získat stejným způsobem jako jsme získali L a takto můžeme pokračovat rekurzivně dál. V tomto schématu tedy sestavujeme matici L po řádcích a to bez toho, že bychom využívali a jakkoli měnili strukturu dolní trojúhelníkové části matice pod modifikovaným řádkem.

Třetí varianta, nazývaná submaticová varianta, je nejbližší představě o klasické Gaussově eliminaci.

Dá se ukázat, že matici L lze dostat jako součin n dolních trojúhelníkových matic s kladnými prvky na diagonále [8], tj.

$$L = L_1 L_2 \dots L_n, \quad (1.4)$$

kde tyto matice pro každé $i = 1, \dots, n$ mají tvar

$$L_i = \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & \sqrt{d_i} & \\ & & & \frac{v_i}{\sqrt{d_i}} & I_{n-i} \end{pmatrix}.$$

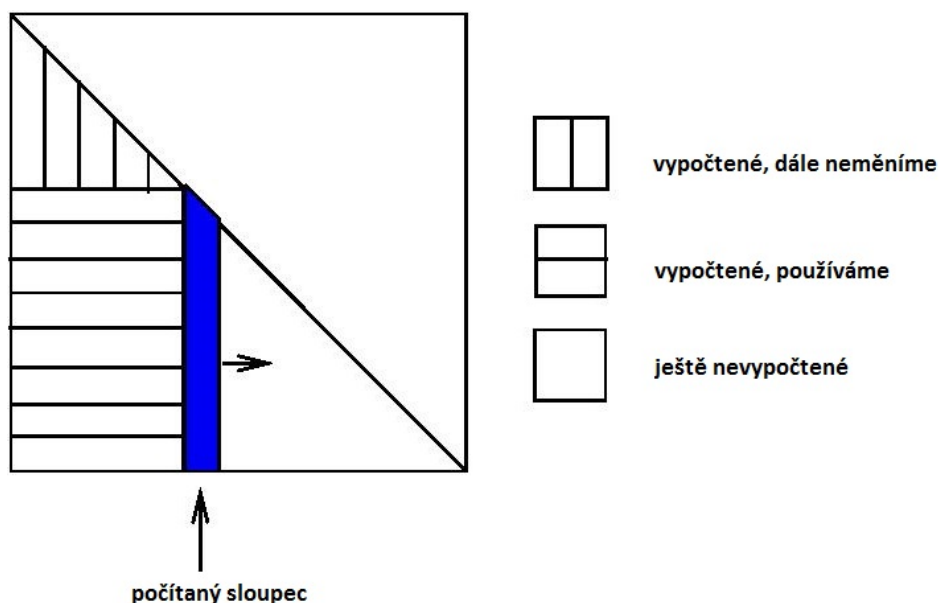
Přitom $d_i \in \mathbb{R}^+$, $v_i \in \mathbb{R}^{n-i}$ a $I_{n-i} \in \mathbb{R}^{(n-i) \times (n-i)}$ je jednotková matice. Odsud dále dostáváme (viz [6])

$$L = L_1 + L_2 + \dots + L_n - (n-1)I_n. \quad (1.5)$$

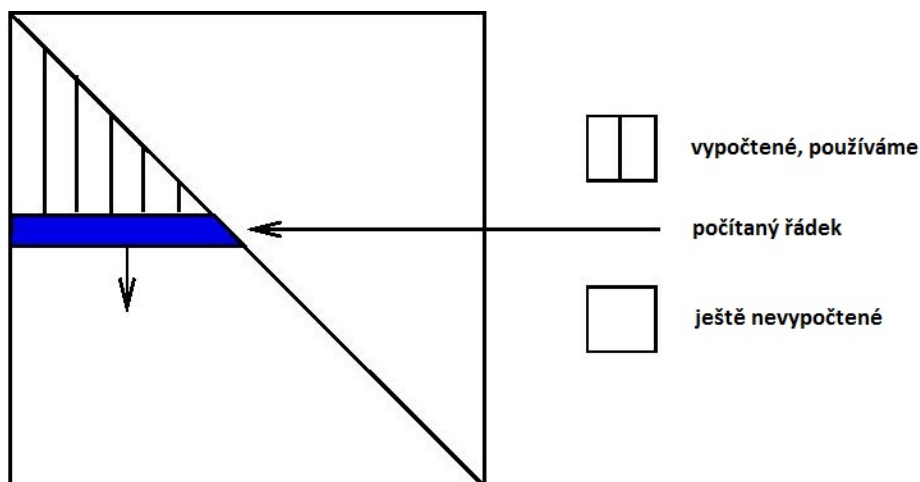
Tedy i -tý sloupec matice L je stejný jako i -tý sloupec matice L_i .

V každém kroku jsme vytvořili Schurův doplněk, ve kterém jsme prováděli další změny, aniž bychom zasahovali do zbylé části matice. Ve sloupcovém algoritmu jsme naopak využívali již vypočítaných sloupců a v části dolního trojúhelníka napravo od sloupce, který v daném kroce modifikujeme, jsme neprováděli žádné změny.

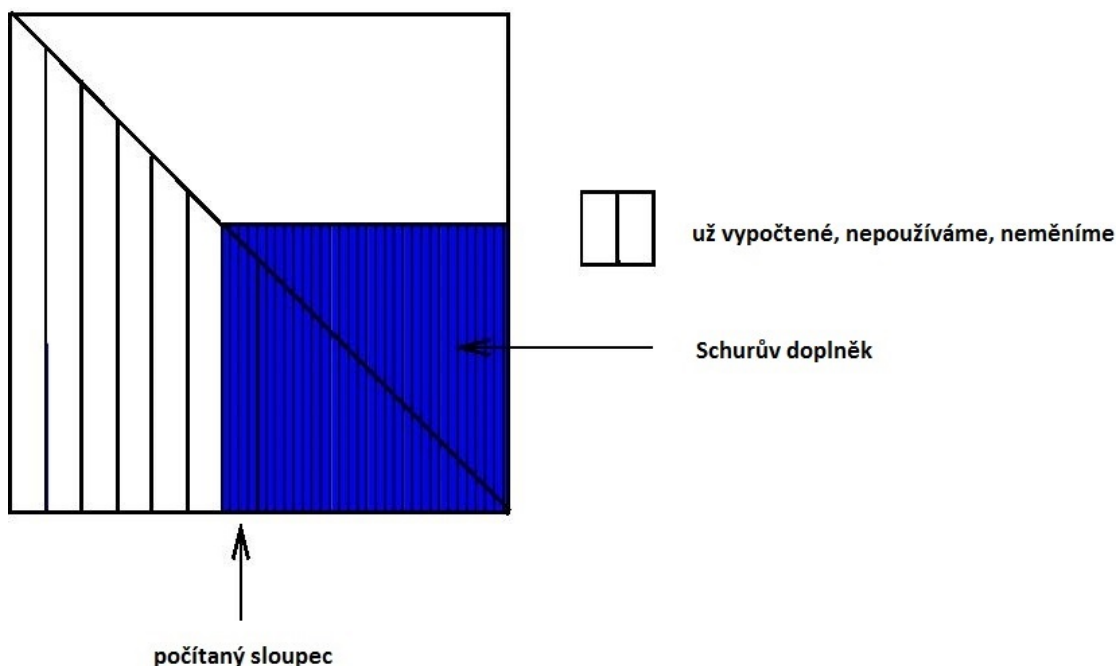
Pro názornost si uvedeme schématické znázornění zmíněných tří variant Choleského rozkladu.



Obrázek 1.1: sloupcová varianta



Obrázek 1.2: řádková varianta



Obrázek 1.3: submaticová varianta

Uvedené tři varianty nám dávají různé náhledy na Choleského faktorizaci matice A , což lze využít při zkoumání rychlosti výpočtu faktoru L . Nedá se jednoznačně říct, že je jeden algoritmus lepší než druhý. Při porovnání záleží na řídkosti matice, struktuře jejích nenulových prvků nebo na konkrétní implementaci a počítačové architektuře.

1.1.2 Řešení soustavy

Mějme lineární soustavu rovnic

$$Ax = b, \quad (1.6)$$

kde $A \in \mathbb{R}^{n \times n}$ je SPD matice, $b \in \mathbb{R}^n$ a $x \in \mathbb{R}^n$ je hledaný vektor řešení. Po aplikaci Choleského faktorizace na matici A dostáváme

$$LL^T x = b, \quad (1.7)$$

což lze napsat jako

$$Ly = b, \quad (1.8)$$

kde $y = L^T x$. Řešíme nejprve soustavu (1.8), a až poté soustavu

$$L^T x = y. \quad (1.9)$$

1.2 Řídké matice

Řídká matice je obvykle definována jako matice, která obsahuje velmi malé množství nenulových prvků. Korektnější je však tato formulace.

Definice 1 ([15]). *Řídká matice je taková matice, jejíž rozložení nulových a nenulových prvků nám umožňuje využít speciálních technik při maticových operacích.*

U hustých matic je obecné schéma hledání vektoru řešení pomocí soustav (1.8) a (1.9) podle předešlého algoritmu vyhovující. Při práci na paralelních počítačových architekturách se ale proces řešení také zkomplikuje. U řídkých matic nastává v průběhu Choleského faktorizace problém se zaplňováním matice A , což vede ke ztrátě řídkosti matice L . Jinak řečeno má matice L nenulové prvky i na místech, kde matice A ne (bráno v dolní trojúhelníkové části). Rozdíl mezi počtem operací potřebných k výpočtu dolní trojúhelníkové matice L u řídkých a hustých matic je razantní, co se týče paměti a ceny výpočtu. Tento problém se dá řešit pomocí tzv. přeuspořádání, který nám umožní pracovat s novou ekvivalentní soustavou

$$\hat{A}\hat{x} = \hat{b}, \quad (1.10)$$

jež vznikla vhodným přeuspořádáním rovnic a pořadím neznámých v soustavě (1.6). Vhodným přeuspořádáním se myslí to, že matice \hat{A} je nejen SPD, ale pustíme-li na ní Choleského faktorizaci, pak se oproti původní matici zaplňuje mnohem méně. Soustavu (1.10) pak řešíme stejně jako bylo uvedeno u hustých matic. Problém vhodného výběru typu přeuspořádání matice A (transformace pomocí nějaké permutační matice) je velmi obsáhlé a diskutované téma, které je nad rámec této bakalářské práce. Pro ilustraci si ale uvedeme alespoň jeden příklad z [8].

Nechť

$$A = \begin{pmatrix} 4 & 1 & 2 & \frac{1}{2} & 2 \\ 1 & \frac{1}{2} & 0 & 0 & 0 \\ 2 & 0 & 3 & 0 & 0 \\ \frac{1}{2} & 0 & 0 & \frac{5}{8} & 0 \\ 2 & 0 & 0 & 0 & 16 \end{pmatrix}, b = \begin{pmatrix} 7 \\ 3 \\ 7 \\ -4 \\ -4 \end{pmatrix}$$

a mějme soustavu $Ax = b$. Matice L z Choleského rozkladu má tvar

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

Zde vidíme, že došlo k zaplnění dolní trojúhelníkové části matice A na šesti místech: $a_{32}, a_{42}, a_{43}, a_{52}, a_{53}, a_{54}$.

Bud'

$$P = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Přenásobením matice A zleva maticí P a zprava maticí P^T dostaneme

$$\hat{A} = PAP^T = \begin{pmatrix} 16 & 0 & 0 & 0 & 2 \\ 0 & \frac{5}{8} & 0 & 0 & \frac{1}{2} \\ 0 & 0 & 3 & 0 & 2 \\ 0 & 0 & 0 & \frac{1}{2} & 1 \\ 2 & \frac{1}{2} & 2 & 1 & 4 \end{pmatrix}.$$

Spustíme-li na tuto matici Choleského faktorizaci, dostaneme $\hat{A} = \hat{L}\hat{L}^T$, kde

$$\hat{L} = \begin{pmatrix} 4 & & & & \\ 0 & 0,791 & & & \\ 0 & 0 & 1,73 & & \\ 0 & 0 & 0 & 0,707 & \\ 0,5 & 0,632 & 1,15 & 1,41 & 1,29 \end{pmatrix}.$$

Všimněme si, že nedošlo k žádnému zaplnění dolní trojúhelníkové části matice \hat{A} . Dále P je unitární matice, tedy platí $P^T = P^{-1}$. Nyní můžeme řešit novou soustavu $\hat{A}\hat{x} = \hat{b}$, přičemž $\hat{x} = Px$ a $\hat{b} = Pb$. Lze snadno nahlédnout, že tato soustava je ekvivalentní původní soustavě. Správný výběr permutační matice P tedy může drasticky zlepšit efektivitu výpočtu řešení x .

2. Přímé metody vs iterační metody

Pro řešení soustav typu

$$Ax = b, \quad A \in \mathbb{R}^{n \times n}, \quad b \in \mathbb{R}^n, \quad x \in \mathbb{R}^n \quad (2.1)$$

je možné použít dva základní typy metod, a to přímé a iterační. Typickým příkladem přímé metody je Gaussova eliminace. Její speciální případ, kterým je Choleského faktorizace, jsme diskutovali výše. Příkladem iterační metody pro řešení soustavy se symetrickou a pozitivně definitní maticí je metoda sdružených gradientů, kterou si odvodíme v další kapitole. Tyto metody jsou velmi odlišné a opět nelze říct, která z nich je lepší, neboť záleží na mnoha faktorech.

Iterační metody jsou založeny na tom, že se zvolí nějaká vhodná počáteční aproximace přesného řešení x , označme x_0 , a určí se posloupnost aproximací $\{x_n\}_{n \in \mathbb{N}}$ tak, že

$$\lim_{n \rightarrow \infty} x_n = x. \quad (2.2)$$

V metodě sdružených gradientů je například $x_0 = 0$ a rekurentní předpis k výpočtu k -té aproximace je dán vzorcem $x_k := x_{k-1} + \gamma_{k-1} p_{k-1}$ (viz třetí kapitola).

Problémem iteračních metod je to, že jsou obecně nespolehlivé, neboť nevíme, jestli vůbec dokonvergujeme k přesnému řešení. V teorii dosáhneme přesného řešení rovnice (2.1) použitím iterační metody po nekonečně mnoha krocích. V praxi pak máme zastavovací kritéria, podle kterých určujeme, kdy pro nás má výpočet ještě smysl [8]. U přímých metod je naopak zaručeno jejich zastavení po konečně mnoha krocích.

Dalším rozdílem je potřebný prostor v paměti počítače. Zatímco u přímých metod provádíme maticové operace, k čemuž je potřeba si vždy uchovávat v paměti celou matici, iterační metody využívají často pouze operace typu matice krát vektor, což je značně levnější a efektivnější na provedení, neboť u matic se speciální strukturou stačí znát způsob, jak vypočítat matice krát vektor. Příkladem nám buď cirkulantní matice (čtvercová matice, jejíž $(j+1)$ -tý sloupec vznikl z j -tého sloupce cyklickou záměnou, tj. $a_1 \rightarrow a_n, a_2 \rightarrow a_1$, atd. pro každé $j = 1, \dots, n-1$)

$$A = \begin{pmatrix} a_1 & a_n & \dots & a_2 \\ a_2 & a_1 & \dots & a_3 \\ \vdots & \vdots & \ddots & \vdots \\ a_n & a_{n-1} & \dots & a_1 \end{pmatrix},$$

kde si stačí uložit pouze první sloupec matice, tj. vektor $a = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$.

Komplikovanější je srovnání podle rychlosti výpočtu. Přímé metody jsou velmi závislé na struktuře matice, to jsme mohli vidět v předešlé kapitole u řídkých matic, kde docházelo k jejich zaplnění. U iteračních metod je zase počet iterací závislý na povaze matice A a na zastavovacím kritériu.

3. Metoda sdružených gradientů

V této kapitole se budeme věnovat nejpoužívanější iterační metodě pro řešení soustavy lineárních rovnic $Ax = b$ s reálnou symetrickou a pozitivně definitní maticí - metodě sdružených gradientů (CG), jež je metodou Krylovových podprostorů.

Princip CG spočívá v tom, že se v každém kroku algoritmu chceme co nejvíce přiblížit k přesnému řešení x vzhledem k energetické normě A a tato aproximace má v k -tém kroku ležet ve varietě $x_0 + K_k(A, r_0)$, tj.

$$\min \|x - x_k\|_A, \quad x_k \in x_0 + K_k(A, r_0),$$

kde x_0 je počáteční aproximace a r_0 je počáteční reziduum.

3.1 Odvození algoritmu

Řešení soustavy lineárních rovnic

$$Ax = b \tag{3.1}$$

lze ztotožnit s úlohou minimalizace funkcionálu energie

$$F(x) = \frac{1}{2}x^T Ax - x^T b. \tag{3.2}$$

Stačí tedy položit $\text{grad}(F(x)) = Ax - b =: 0$.

Definice 2. *Buď A symetrická a pozitivně definitní matice. Skalární součin na A definujeme předpisem $(x, y)_A = x^T Ay$ a energetickou normu indukovanou $(x, y)_A$ jako $\|x\|_A = \sqrt{(x, x)_A}$.*

Dosažením bodu x_k do F dostaneme $F(x_k) = \frac{1}{2}\|x - x_k\|_A^2 - \frac{1}{2}\|x\|_A^2$. Odsud vidíme, že je nutné minimalizovat $x - x_k$ v A -normě. Konstruujeme tedy postupně posloupnost aproximací $\{x_k\}_{k \in \mathbb{N}}$ tak, že

$$x_k = x_{k-1} + \gamma_{k-1}p_{k-1}, \tag{3.3}$$

kde p_{k-1} je směr minimalizace (zatím předpokládáme, že nám ho někdo dal) a γ_{k-1} je tzv. krok minimalizace zvolený jako

$$\gamma_{k-1} = \frac{p_{k-1}^T r_{k-1}}{p_{k-1}^T A p_{k-1}}. \tag{3.4}$$

Přenosobením rovnice (3.3) maticí A zleva a odečtením od vektoru b dostaneme rekurentní vztah pro reziduum

$$r_k = r_{k-1} - \gamma_{k-1}A p_{k-1}. \tag{3.5}$$

Lze jednoduše ukázat, že platí $r_k \perp p_{k-1}$ pro každé $k \in \mathbb{N}$:

$$p_{k-1}^T r_k = p_{k-1}^T r_{k-1} - \gamma_{k-1} p_{k-1}^T A p_{k-1} = p_{k-1}^T r_{k-1} - \frac{p_{k-1}^T r_{k-1}}{p_{k-1}^T A p_{k-1}} p_{k-1}^T A p_{k-1} = 0.$$

Nyní už máme aparát na sestavení směrů minimalizace. Volme $p_0 = r_0$ (tj. budeme opravovat počáteční aproximaci ve směru rezidua) a pro $k \in \mathbb{N}$ položíme

$$p_k = r_k + \delta_k p_{k-1}, \quad (3.6)$$

kde δ_k je číslo definované tak, aby $p_k \perp_A p_{k-1}$.

Tedy

$$0 = p_{k-1}^T A p_k = p_{k-1}^T (A r_k + \delta_k A p_{k-1}),$$

což dává

$$\delta_k = -\frac{p_{k-1}^T A r_k}{p_{k-1}^T A p_{k-1}}. \quad (3.7)$$

Lemma 3. *Pro výše definované skaláry platí*

$$\gamma_{k-1} = \frac{r_{k-1}^T r_{k-1}}{p_{k-1}^T A p_{k-1}}, \quad \delta_k = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}.$$

Důkaz. Dosazením vztahů z (3.6), (3.5) do (3.4), (3.7) a následnými úpravami dostaneme požadované rovnosti. \square

Algorithm 1 Metoda sdružených gradientů

```

1: input  $A, b, x_0$ 
2:  $r_0 := b - Ax_0$ 
3:  $p_0 := r_0$ 
4: for  $k = 1, 2, \dots$  do
5:    $\gamma_{k-1} := \frac{r_{k-1}^T r_{k-1}}{p_{k-1}^T A p_{k-1}}$ 
6:    $x_k := x_{k-1} + \gamma_{k-1} p_{k-1}$ 
7:    $r_k := r_{k-1} - \gamma_{k-1} A p_{k-1}$ 
8:    $\delta_k = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}$ 
9:    $p_k := r_k + \delta_k p_{k-1}$ 
10: end for

```

Z lokální A -ortogonalitý směrůvých vektorů, kterou jsme dostali speciální volbou δ_k , lze matematickou indukcí dokázat i jejich globální A -ortogonalitu, a navíc globální ortogonalitu reziduí vzhledem k eukleidovskému skalárnímu součinu, tj.

$$p_i \perp_A p_j \text{ a } r_i \perp r_j, \forall i \neq j.$$

Jelikož pracujeme v konečněrozměrném prostoru ($A \in \mathbb{R}^{n \times n}$), tak můžeme Algoritmem 1 zvolit maximálně n směrůvých vektorů. Z tohoto faktu plyne, že metodou sdružených gradientů dokonvergujeme k přesnému řešení (v matematicky přesné aritmetice [12]) maximálně po n krocích. Dá se totiž ukázat, že platí

Lemma 4. *Následující tvrzení jsou ekvivalentní*

- i) *Tříčlenná rekurence (3.3) se zastaví, tj. nastane $x_k = x_{k-1}$;*
- ii) *$x_k = x$, kde x je přesné řešení soustavy (3.1);*
- iii) *$p_{k-1} = 0$ nebo $\gamma_{k-1} = 0$.*

Důkaz. Předpokládejme, že platí $x_k = x_{k-1}$. Pak z rovnice (3.3) vyplývá, že toto nastane právě tehdy, když $p_{k-1} = 0$ nebo $\gamma_{k-1} = 0 \iff p_{k-1}^T r_{k-1} = 0 \iff r_{k-1} = 0$. Tedy dostáváme, že reziduum je nulové, tedy jsme dokonvergovali k přesnému řešení x . \square

3.2 Konvergence CG

Odvozením výše uvedených vztahů získáváme rovnosti

$$\text{span}\{p_0, p_1, \dots, p_{k-1}\} = \text{span}\{r_0, r_1, \dots, r_{k-1}\} = K_k(A, r_0).$$

Zároveň si x_k můžeme vyjádřit jako

$$x_k = x_0 + \sum_{j=1}^k \gamma_{j-1} p_{j-1},$$

tedy x_k leží ve varietě $x_0 + K_k(A, r_0)$. Odsud

$$\|x - x_k\|_A = \min_{y \in x_0 + K_k(A, r_0)} \|x - y\|_A.$$

Nejpoužívanějším odhadem pro konvergenci CG je odhad, který je založen na odhadu pro Čebyševovu semiiterační metodu [12]

$$\frac{\|e^{(k)}\|_A}{\|e^{(0)}\|_A} \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k, \quad (3.8)$$

kde $e^{(k)} = x - x_k$ je chyba k -té aproximace a $\kappa(A)$ je číslo podmíněnosti matice A . Tento odhad nám dává rychlou konvergenci CG pro $\kappa(A) \sim 1$, naopak pro velké $\kappa(A)$ není odhad v praxi příliš zajímavý. Konvergence CG závisí především na rozložení všech vlastních čísel matice A a na zaokrouhlovacích chybách.

Zlepšení konvergence můžeme dosáhnout pomocí předpodmínění matice A nějakou jí blízkou, symetrickou a pozitivně definitní, hezkou maticí v tom smyslu, že její konstrukce a aplikace budou levné. Zároveň se vyžaduje, aby předpodmíněný systém byl snadno řešitelný. V 70. letech 20. století představili Meijerink a Van der Vorst neúplnou Choleského faktorizaci pro sdružené gradienty (ICCG), jež právě řeší tento problém nalezení matice splňující výše uvedené požadavky.

4. Předpodmíněné iterační metody

V matematickém modelování fyzikálních jevů se často setkáme s parciálními diferenciálními rovnicemi, k jejichž vyřešení je potřeba numerických schémat, které vedou k soustavám lineárních rovnic s velkou řídkou maticí. K řešení těchto soustav se využívají především iterační metody, neboť (jak jsme již viděli v kapitolách 1-2) u přímých metod dochází k zaplňování matice, a tedy k podstatnému zvýšení ceny výpočtu. Problémem iteračních metod je však jejich nespolehlivost. V 70. letech 20. století se zjistilo, že lze dosáhnout zrychlení konvergence iteračních metod využitím tzv. předpodmínění, více o historii předpodmínění v [1].

Hlavní ideou předpodmínění je modifikace dané úlohy

$$Ax = b \quad (4.1)$$

na „hezčí“ úlohu

$$\hat{A}\hat{x} = \hat{b}, \quad (4.2)$$

kteřá je lépe řešitelná a výpočetně méně nákladná nežli (4.1). Chceme tedy matici A aproximovat nějakou maticí M , která je jí v nějakém smyslu blízka (tj. $M \approx A$, $M^{-1}A \approx I$), lze snadno zkonstruovat a výpočet s ní je levný. Tato matice je vždy regulární a umožňuje nám transformaci (4.1) na (4.2) a nazývá se matice předpodmínění.

Matici předpodmínění M lze aplikovat třemi způsoby:

- zleva

$$M^{-1}Ax = M^{-1}b \quad (4.3)$$

- zprava

$$AM^{-1}y = b, \quad x = M^{-1}y \quad (4.4)$$

- oboustranně

$$M_1^{-1}AM_2^{-1}y = M_1^{-1}b, \quad x = M_2^{-1}y, \quad M = M_1M_2. \quad (4.5)$$

Je důležité, kterou z těchto možností zvolíme, neboť se dle typu matic A a M může konvergence iterační metody chovat různě v (4.3), (4.4), (4.5) jako je tomu například u matic daleko od normálních aplikací GMRES [1].

Z odhadu (3.8) z kapitoly 3 víme, že u SPD matic záleží rychlost konvergence CG na rozložení vlastních čísel matice A . Jsou-li vlastní čísla rovnoměrně rozložená, pak CG konverguje obecně špatně.

Uvažujme nyní soustavu (4.1) s velkou řídkou maticí A . Jeden z nejpoužívanějších postupů k nalezení matice předpodmínění je neúplná LU faktorizace (ILU) matice A , která vznikla modifikací Gaussovy eliminace. Jelikož chceme, aby matice L a U byly také řídké a operace s nimi byly levné, tak definujeme množinu S jako množinu obsahující všechny pozice prvků matice A , kterým dovolíme se v průběhu Gaussovy eliminace měnit. Obvykle sem patří i diagonální

prvky. Množinu S volíme podle nějakých kritérií zohledňujících strukturu matice A nebo velikost jejích prvků (viz následující kapitola). Speciálním případem ILU pro symetrickou a pozitivně definitní matici A je neúplná Choleského faktorizace (IC), kde matice předpodmínění je tvaru

$$M = LL^T \approx A.$$

5. Neúplná Choleského faktorizace

5.1 Existence

Neúplná Choleského faktorizace (IC) se záhy po svém uvedení stala nejslibnějším typem předpodmínění pro metodu sdružených gradientů. Bylo přirozené klást si otázku, kdy a za jakých okolností existuje neúplná Choleského faktorizace. Touto problematikou se zabývali v 70. letech 20. století Meijerink a Van der Vorst, kteří v [14] dokázali existenci neúplné LU faktorizace (ILU) s libovolným zaplněním pro M -matice. Odtud následně vyplývá existence IC pro symetrické M -matice. Není potřeba zvlášť vyzdvihovat pozitivní definitnost, neboť každá symetrická M -matice je pozitivně definitní dle [2, 10]. Na základě těchto a dalších výsledků Meijerinka a Van der Vorsta se začalo počítání pomocí předpodmíněných sdružených gradientů (PCG) intenzivně rozvíjet. Ukažme si nyní několik teoretických tvrzení.

Definice 3. Řekneme, že matice $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ je M -matice, jestliže platí

$$\begin{aligned} a_{ij} &\leq 0 \quad \forall i \neq j, \\ a_{ii} &> 0 \quad \forall i \in \{1, \dots, n\}, \end{aligned}$$

A je regulární a $A^{-1} \geq 0$.

Poznámka: Takovýto typ matic získáváme často diskretizací eliptických a parabolických parciálních diferenciálních rovnic.

Definice 4. Řekneme, že štěpení $A = B - C$ je regulární, pokud $B^{-1} \geq 0$ a $C \geq 0$.

K důkazu věty o existenci neúplné LU faktorizace budeme potřebovat následující dvě věty, které nebudeme dokazovat. Důkazy lze najít v [14, 7, 16].

První věta nám říká, že aplikací jednoho kroku Gaussovy eliminace na M -matici dostaneme opět M -matici.

Věta 5 (Ky Fan [7, strana 44]). *Bud' $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ M -matice. Označme $A^{(1)}$ matici získanou z A eliminací prvního sloupce pomocí prvního řádku. Pak $A^{(1)} = (a_{ij}^{(1)})$ je také M -matice.*

Následující větu použijeme k tomu, abychom mohli vynechat některé mimodiagonální prvky během konstrukce neúplné LU faktorizace matice A ve Větě 7.

Věta 6 ([14, strana 150], [16]). *Nechť $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ je M -matice a $B = (b_{ij}) \in \mathbb{R}^{n \times n}$ je matice splňující*

$$\begin{aligned} a_{ij} \leq b_{ij} \leq 0 \quad \forall i \neq j, \\ 0 < a_{ii} \leq b_{ii} \quad \forall i \in \{1, \dots, n\}. \end{aligned}$$

Pak je B také M -matice.

Nyní už máme nástroje k dokázání věty o existenci ILU. Před tím ale ještě definujeme množinu S takovýmto způsobem

$$S \subset n \times n \quad \wedge \quad \{(i,j) \in n \times n; \quad i = j\} \subset S. \quad (5.1)$$

Věta 7. *Nechť $A = (a_{ij}) \in \mathbb{R}^{n \times n}$ je M -matice. Pro každou množinu S definovanou výše pak existují dolní trojúhelníková matice $L = (l_{ij})$ s jedničkami na diagonále, horní trojúhelníková matice $U = (u_{ij})$ a matice $R = (r_{ij})$ splňující*

$$\begin{aligned} l_{ij} &= 0 \quad \text{pro } (i,j) \notin S, \\ u_{ij} &= 0 \quad \text{pro } (i,j) \notin S, \\ r_{ij} &= 0 \quad \text{pro } (i,j) \in S, \\ A &= LU - R \quad \text{je regulární štěpení.} \end{aligned}$$

Matice L a U jsou navíc určeny jednoznačně.

Důkaz. Na začátek shrňme v několika větách, jak budeme v důkazu postupovat. Matice L , U a R zkonstruujeme v $(n - 1)$ krocích. V k -tém kroku odečteme od matice $A^{(k-1)}$ její prvky na pozicích (k,j) , (i,k) , pokud neleží v množině S , a dostaneme tak matici $A^{(k)}$. Odečtené prvky budou na stejných pozicích s opačným znaménkem v matici $R^{(k)}$ a ostatní hodnoty této matice nastavíme na nulu. Poté spustíme jeden krok Gaussovy eliminace na matici $A^{(k)}$. Po $(n - 1)$ krocích bychom tak měli dostat horní trojúhelníkovou matici $A^{(n-1)}$, což bude naše matice U , dolní trojúhelníkovou matici $L = \left(L^{(n-1)}L^{(n-2)} \dots L^{(1)}\right)^{-1}$ a nezápornou matici $R := R^{(1)} + R^{(2)} + \dots + R^{(n-1)}$ složenou z příspěvků k celkové chybě tak, že $A = LU - R$.

Nejdříve si zavedeme značení

$$A^{(k)} = \left(a_{ij}^{(k)}\right), \quad \hat{A}^{(k)} = \left(\hat{a}_{ij}^{(k)}\right), \quad L^{(k)} = \left(l_{ij}^{(k)}\right), \quad R^{(k)} = \left(r_{ij}^{(k)}\right).$$

Definujeme pro $\forall k = 1, \dots, n - 1$

$$\begin{aligned} A^{(0)} &:= A, \\ \hat{A}^{(k)} &:= A^{(k-1)} + R^{(k)}, \\ A^{(k)} &:= L^{(k)} \hat{A}^{(k)}, \end{aligned}$$

kde matice $R^{(k)}$ vypadá takto

$$\begin{aligned} r_{k,j}^{(k)} &= -a_{k,j}^{(k-1)}, \quad \text{pokud } (k,j) \notin S, \\ r_{i,k}^{(k)} &= -a_{i,k}^{(k-1)}, \quad \text{pokud } (i,k) \notin S, \\ r_{i,j}^{(k)} &= 0, \quad \text{jinak.} \end{aligned}$$

Matice $L^{(k)}$ je matice z Gaussovy eliminace aplikovaná na $\hat{A}^{(k)}$, která vynuluje prvky v k -tém sloupci matice $\hat{A}^{(k)}$ pod prvkem $\hat{a}_{kk}^{(k)}$ pomocí k -tého řádku. Tedy

matice $L^{(k)}$ je tvaru

$$L^{(k)} = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -\frac{\hat{a}_{k+1,k}^{(k)}}{\hat{a}_{k,k}^{(k)}} & & & \\ & & \vdots & & \ddots & \\ & & -\frac{\hat{a}_{n,k}^{(k)}}{\hat{a}_{k,k}^{(k)}} & & & 1 \end{pmatrix} = I + l^{(k)} e_k^T, \quad l^{(k)} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -\frac{\hat{a}_{k+1,k}^{(k)}}{\hat{a}_{k,k}^{(k)}} \\ \vdots \\ -\frac{\hat{a}_{n,k}^{(k)}}{\hat{a}_{k,k}^{(k)}} \end{pmatrix}.$$

$A^{(0)} = A$ byla dle předpokladů M -matice, tedy všechny její mimodiagonální prvky jsou nekladné. Z tohoto faktu a z definice $R^{(1)}$ plyne, že $R^{(1)} \geq 0$, tj. všechny prvky matice $R^{(1)}$ jsou nezáporné.

Použitím Věty 6 na matice $A^{(0)}$ a $\hat{A}^{(1)} = A^{(0)} + R^{(1)}$ dostaneme, že $\hat{A}^{(1)}$ je M -matice, neboť

$$\begin{aligned} a_{ij}^{(0)} &\leq \hat{a}_{ij}^{(1)} \leq 0 \quad \forall i \neq j, \\ \hat{a}_{ii}^{(1)} &= a_{ii}^{(0)} > 0 \quad \forall i. \end{aligned}$$

Tedy musí platit

$$L^{(1)} = \begin{pmatrix} 1 & & & & & \\ -\frac{\hat{a}_{2,1}^{(1)}}{\hat{a}_{1,1}^{(1)}} & 1 & & & & \\ -\frac{\hat{a}_{3,1}^{(1)}}{\hat{a}_{1,1}^{(1)}} & & 1 & & & \\ \vdots & & & \ddots & & \\ -\frac{\hat{a}_{n,1}^{(1)}}{\hat{a}_{1,1}^{(1)}} & & & & & 1 \end{pmatrix} \geq 0.$$

Dále aplikací Věty 5 na $A^{(1)} = L^{(1)} \hat{A}^{(1)}$ dostáváme, že $A^{(1)}$ je také M -matice. Stejným postupem ukážeme, že

$$\begin{aligned} A^{(k)} &\text{ je } M\text{-matice,} \\ \hat{A}^{(k)} &\text{ je } M\text{-matice,} \\ L^{(k)} &\geq 0, \\ R^{(k)} &\geq 0 \quad \forall k = 1, \dots, n-1. \end{aligned}$$

Je jednoduché odvodit, že platí

$$L^{(k)} R^{(m)} = R^{(m)} \quad \text{pro } k < m. \quad (5.2)$$

Odtud a z definic uvedených na začátku důkazu plyne

$$\begin{aligned}
A^{(n-1)} &= L^{(n-1)} \hat{A}^{(n-1)} \\
&= L^{(n-1)} A^{(n-2)} + L^{(n-1)} R^{(n-1)} \\
&= L^{(n-1)} L^{(n-2)} \hat{A}^{(n-2)} + L^{(n-1)} R^{(n-1)} \\
&= L^{(n-1)} L^{(n-2)} A^{(n-3)} + L^{(n-1)} L^{(n-2)} R^{(n-2)} + L^{(n-1)} R^{(n-1)} \\
&\quad \vdots \\
&= L^{(n-1)} L^{(n-2)} \dots L^{(1)} A^{(0)} + L^{(n-1)} \dots L^{(1)} R^{(1)} + L^{(n-1)} \dots L^{(2)} R^{(2)} \\
&\quad + \dots + L^{(n-1)} R^{(n-1)} \\
&\stackrel{(5.2)}{=} L^{(n-1)} L^{(n-2)} \dots L^{(1)} \left(A + R^{(1)} + R^{(2)} + \dots + R^{(n-1)} \right).
\end{aligned}$$

Označíme-li nyní $U := A^{(n-1)}$, $L := \left(L^{(n-1)} L^{(n-2)} \dots L^{(1)} \right)^{-1}$ a $R := R^{(1)} + R^{(2)} + \dots + R^{(n-1)}$, pak $A = LU - R$. Máme tedy dokázanou existenci matic L , U a R .

Navíc platí

$$(LU)^{-1} = U^{-1} L^{-1} = \left(A^{(n-1)} \right)^{-1} L^{-1} \geq 0, \quad R \geq 0,$$

a tedy štěpení $A = LU - R$ je regulární.

Jednoznačnost matic L a U vyplývá z jejich konstrukce a z toho, že L má na diagonále samé jedničky. □

Z této věty už přímo plyne její speciální verze pro symetrické M -matice, která nám dává existenci neúplné Choleského faktorizace.

Věta 8. *Buď A symetrická M -matice. Pak pro každou množinu S splňující (5.1) existuje jednoznačně určená dolní trojúhelníková matice L s kladnými prvky na diagonále a symetrická nezáporná matice R tak, že*

$$\begin{aligned}
l_{ij} &= 0 \quad \text{pro } (i,j) \notin S, \\
r_{ij} &= 0 \quad \text{pro } (i,j) \in S
\end{aligned}$$

a štěpení $A = LL^T - R$ je regulární.

Důkaz. Stejně jako v důkazu Věty 7 dostaneme $A = \hat{L}U - R$, kde $U = D\hat{L}^T$ a D je diagonální matice s kladnými prvky na diagonále. Tedy $L := \hat{L}\hat{D}$, kde $\hat{D} = \text{diag}(\sqrt{d_{11}}, \dots, \sqrt{d_{nn}})$. □

Definice 5 ([9]). *Matice $A \in \mathbb{R}^{n \times n}$ je ostře diagonálně dominantní, jestliže platí*

$$\sum_{\substack{j=1 \\ j \neq i}}^n |a_{ij}| < |a_{ii}| \quad \text{pro } \forall i \in \{1, \dots, n\}.$$

Definice 6 ([11]). *Matici $A \in \mathbb{R}^{n \times n}$ nazveme H -maticí, pokud je A zobecněná ostře diagonálně dominantní matice, tj. existuje vektor $x \in \mathbb{R}^n$, $x > 0$ takový, že $\hat{A} = A \text{diag}(x_1, \dots, x_n)$ je ostře diagonálně dominantní matice.*

Nedlouho poté Varga, Saff a Mehrman rozšířili ve své práci důkaz na existenci ILU (resp. IC) pro obecnější typ matic, a to H -matice. Naneštěstí pak ale Kershaw ukázal, že pro SPD matice IC obecně nemusí existovat, neboť může nastat tzv. zlom (ang. pivot breakdown). Tato situace vzniká tehdy, když se nám ve faktoru L objeví na diagonále nula nebo záporné číslo. To vede k závažnému problému, kdy naše matice předpodmínění už není pozitivně definitní, a nemůžeme ji tak použít k předpodmínění metody sdružených gradientů. Začalo se tedy zkoumat, jak neúplnou Choleského faktorizaci zlepšit, aby nedošlo ke zlomu.

S návrhem přišel nejdříve Kershaw, který záporné a malé pivoty nahradil v každém kroku nějakým kladným číslem p_i . Ukázalo se, že pro malý počet takovýchto pivotů, nám dává tato metoda pořád uspokojivou matici předpodmínění. Problém ale nastane při výběru čísla p_i [1].

O něco úspěšnější postup navrhl Manteuffel v [13]. Dokázal, že lze vždy najít takové číslo $\alpha > 0$, aby proběhla neúplná Choleského faktorizace matice $\hat{A} = A + \alpha \text{diag}(A)$ bez zlomu. Jelikož se ale číslo α získá zkoušením, může se nám výpočet značně prodražit.

Nakonec ještě uvedeme návrh prezentovaný Ajizem a Jenningsem, který spočívá v tom, že vynulujeme-li během IC prvek a_{ij} , pak musíme následně jeho absolutní hodnotu přičíst k diagonálním prvkům a_{ii} a a_{jj} . Výsledek IC, tj. dolní trojúhelníková matice L , je pak přesně matice získaná Choleského faktorizací matice $\hat{A} = A + C$, kde C je pozitivně definitní matice nazývaná matice vyrušení (ang. cancellation matrix). Velkým kladem tohoto postupu je, že dostaneme $\rho(I - \hat{A}^{-1}A) < 1$. Získaná matice předpodmínění \hat{A} však může být daleko od matice A .

5.2 Základní algoritmy

Uvedeme si nyní dva základní algoritmy neúplné Choleského faktorizace, její klasickou a modifikovanou verzi. Z těchto dvou algoritmů jsou pak volbou kritérií na množinu S z (5.1) odvozeny různé varianty IC, o kterých budeme mluvit v sekci 5.3.

5.2.1 Klasická neúplná Choleského faktorizace

O klasické neúplné Choleského faktorizaci (dále jen IC) jsme již krátce mluvili v kapitole o předpodmíněných iteračních metodách. IC je takovou modifikací standardní Choleského faktorizace, že zaplňuje pouze předem určená místa výsledek matice L (resp. L^T), neboť chceme, aby i tento faktor L udržoval řídkou strukturu původní matice A . Množinu obsahující tato místa jsme označili S , viz (5.1). Požadujeme tedy, aby

$$l_{ij} = 0 \quad \Leftrightarrow \quad j > i \quad \vee \quad (i,j) \notin S. \quad (5.3)$$

Algorithm 2 IC

```
1: input  $A = (a_{ij}) \in \mathbb{R}^{n \times n}, SPD$ 
2: for  $k = 1 : n$  do
3:    $a_{k,k} := \sqrt{a_{k,k}}$ 
4:   for  $i = k + 1 : n$  do
5:     if  $(i,k) \in S$  then
6:        $a_{i,k} := \frac{a_{i,k}}{a_{k,k}}$ 
7:     end if
8:     for  $j = k + 1 : n$  do
9:       if  $(i,j) \in S$  and  $(j,k) \in S$  then
10:         $a_{i,j} := a_{i,j} - a_{i,k}a_{j,k}$ 
11:       end if
12:     end for
13:   end for
14: end for
```

Pokud množina S obsahuje všechny možné pozice matice A , tj. $S = \{(i,j) \in n \times n\}$, pak dostaneme standardní úplnou Choleského faktorizaci. Zvolíme-li $S = \{(i,j) \in n \times n; a_{ij} \neq 0\}$, pak má matice L nuly na stejných místech jako dolní trojúhelníková část matice A . Pro takto zvolenou množinu S říkáme, že máme neúplnou Choleského faktorizaci s nulovým zaplněním a značíme ji zkratkou IC(0).

Lze ukázat, že Algoritmem 2 dostaneme pro matici $M = LL^T$ rovnost

$$m_{i,j} = a_{i,j}, \quad \text{pokud } (i,j) \in S. \quad (5.4)$$

Tato podmínka nám v některých případech může zaručit dobrou matici předpokládání. Postupovat jde ale i obráceně. Chceme-li splnit (5.3) a (5.4), dostaneme přesně Algoritmus 2.

5.2.2 Modifikovaná neúplná Choleského faktorizace

Místo podmínky (5.4) můžeme od matice M požadovat, aby platilo

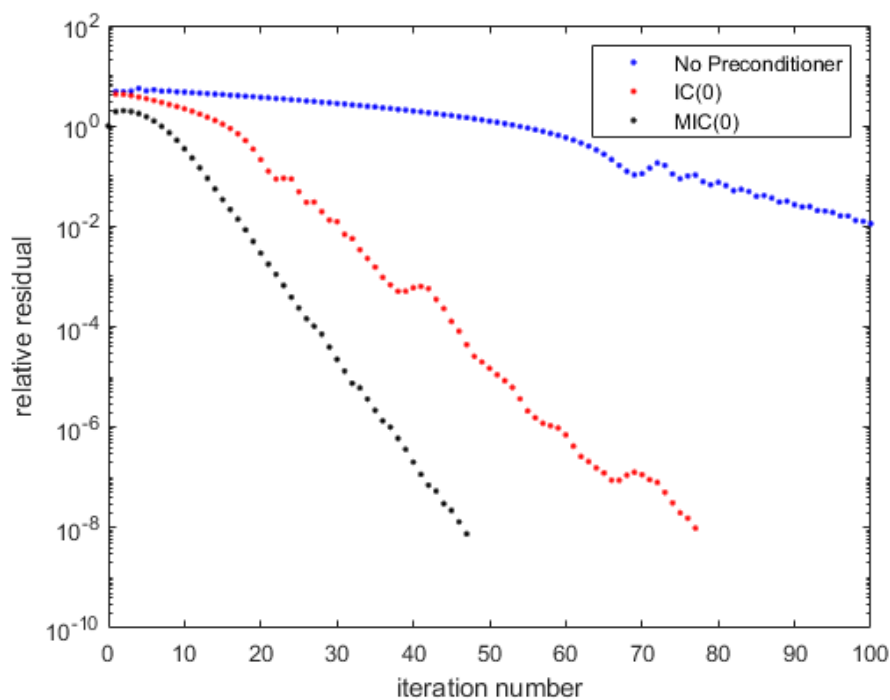
$$\sum_{j=1}^n m_{i,j} = \sum_{j=1}^n a_{i,j} \quad \forall i \quad \text{a} \quad m_{i,j} = a_{i,j}, \quad \text{pokud } i \neq j \quad \wedge \quad (i,j) \in S \quad (5.5)$$

pro nějakou volbu S . V tom tedy spočívá hlavní rozdíl mezi modifikovanou neúplnou Choleského faktorizací (dále jen MIC) a IC.

Předpokládejme nyní splnění podmínky (5.5). To jest, chceme, aby součet každého řádku matice M byl stejný jako součet daného řádku matice A a také aby mimodiagonální prvky matice M , jejichž pozice leží v množině S , měly stejnou hodnotu jako jim příslušné prvky matice A . Pak musíme v každém kroku algoritmu, kdy nulujeme nějaký prvek, přičíst jeho změnu k diagonálnímu prvku na téže řádce. Lépe je to vidět z Algoritmu 3.

Algorithm 3 MIC

```
1: input  $A = (a_{ij}) \in \mathbb{R}^{n \times n}, SPD$ 
2: for  $k = 1 : n$  do
3:    $a_{k,k} := \sqrt{a_{k,k}}$ 
4:   for  $i = k + 1 : n$  do
5:     if  $(i,k) \in S$  then
6:        $a_{i,k} := \frac{a_{i,k}}{a_{k,k}}$ 
7:     end if
8:     for  $j = k + 1 : n$  do
9:       if  $(j,k) \in S$  then
10:        if  $(i,j) \in S$  then
11:           $a_{i,j} := a_{i,j} - a_{i,k}a_{j,k}$ 
12:        else
13:           $a_{i,i} := a_{i,i} - a_{i,k}a_{j,k}$ 
14:        end if
15:      end if
16:    end for
17:  end for
18: end for
```



Obrázek 5.1: Srovnání CG s PCG, kde jsme matici předpodmínění získali pomocí IC(0) a MIC(0)

5.3 Varianty

Existuje mnoho variant neúplné Choleského faktorizace, které se liší volbou množiny S . V této sekci si představíme ty nejpůvodnější z nich a nakonec zkusíme otestovat nově navrženou variantu.

Při představování klasické verze IC jsme si již uváděli jeden příklad, jak množinu S volit. Požadovali jsme, aby matice předpodmínění $M = LL^T$ měla stejnou strukturu (tj. měla stejný počet nenulových prvků na stejných pozicích) jako matice A . Toho jsme dosáhli volbou $S = \{(i,j) \in n \times n; a_{ij} \neq 0\}$. Tuto variantu jsme poté označili IC(0) a nazvali jí IC s nulovým zaplněním.

Varianty neúplné Choleského faktorizace se dají rozdělit do tří skupin podle způsobu volby S :

- podle struktury původní matice A
- podle velikosti prvků matice A
- kombinace předchazejících dvou pohledů, tzv. současné praktické strategie.

Je jasné, že IC(0) bude spadat pod první skupinu, neboť nám stačí znát strukturu matice A . Hezké na této variantě je to, že funguje velmi jednoduše a rychle, avšak je efektivní jen pro nějaké druhy problémů (např. pro diskretizace eliptických PDR nízkých řádů vedoucí k SLR s M -maticemi nebo diagonálně dominantními maticemi), neboť nám dává velmi hrubý odhad matice A . Chceme tedy nějak vylepšit tuto variantu tím, že dovolíme více nenulových prvků v matici M . To je spojené s pojmem level-of-fill, tj. stupněm zaplnění.

5.3.1 Podle struktury

Do této skupiny se řadí varianty, jež jsou založené pouze na znalosti struktury matice A . Každému prvku matice A je na začátku přiřazeno číslo

$$lev_{ij} = \begin{cases} 0, & \text{pokud } a_{ij} \neq 0 \\ \infty, & \text{jinak,} \end{cases} \quad (5.6)$$

které postupně při každé změně hodnoty tohoto prvku, ke které dochází během IC (cyklus j v Algorimu 2), aktualizujeme vzorcem

$$lev_{ij} = \min\{lev_{ij}, lev_{ik} + lev_{kj} + 1\}. \quad (5.7)$$

Předpokládejme nyní, že jsme si na začátku zvolili číslo $l \in \mathbb{N} \cup \{0\}$, kterému budeme říkat stupeň zaplnění. S jsme pak definovali jako množinu $\{(i,j) \in n \times n; lev_{ij} \leq l\}$. V Algoritu 2 tedy navíc pod osmým řádkem přibude příkaz: spočítej lev_{ij} . Tuto variantu označíme IC(l). Nyní už tedy konečně víme, co doopravdy značilo číslo 0 v IC(0).

Kromě velmi složitých problémů je rozumné zvolit číslo l malé, neboť chceme co nejvíce zachovat efektivnost výpočtu faktoru L a operací s ním spojených. V řadě případů je už algoritmus IC(1) považován za přijatelné zlepšení IC(0).

Tento přístup funguje dobře pro diagonálně dominantní matice. Naopak pro matice daleko od diagonálně dominantních může varianta selhat, neboť nemáme

žádné informace o velikosti prvků a může se tak stát, že si v paměti ukládáme velké množství malých prvků v absolutní hodnotě, které jsou nezajímavé pro kvalitu matice předpodmínění a bylo by lepší je vynulovat. Následující přístup je založen na přijímání nových prvků zaplnění a nulování jiných prvků na základě jejich velikosti.

5.3.2 Podle velikosti prvků

Klíčovým slovem je zde tzv. dropping tolerance (dále jen tolerance), což je námi zvolená kladná konstanta τ , jež bude určovat, kdy novou hodnotu prvku a_{ij} vznikající během procesu IC, počítáme. Srovnání velikosti prvku s číslem τ můžeme dělat dvěma základními způsoby.

První způsob je tzv. absolutní nulování (ang. absolute dropping strategy), kde dovolíme zaplnění na místě (i,j) pouze v případě, že absolutní hodnota nově vypočítaného prvku a_{ij} je větší než tolerance τ . Výhodou této strategie je možnost kontroly růstového faktoru, který rozhoduje o zpětné stabilitě daného výpočtu. Z praktického pohledu ale matice musí být dobře škálovaná. V jiných případech je lepší použít druhý způsob, tzv. relativní nulování (ang. relative dropping strategy).

U relativního nulování opět bereme absolutní hodnotu nově vypočítaného prvku, který leží na i -tém řádku a v j -tém sloupci a srovnáváme ho nyní s číslem $\tau\|\mathbf{a}_i\|_2$, kde a_i je i -tý řádek matice A . Je-li $|a_{ij}| > \tau\|\mathbf{a}_i\|_2$, pak aktualizujeme hodnotu tohoto prvku, jinak necháme prvku jeho starou hodnotu. Výhodou této strategie je udržení řídkosti faktoru L , zatímco však jeho růstový faktor nemá velký vliv na proces nulování.

Problémem tohoto postupu je určení dobrého čísla τ . Jelikož je jeho výběr dosti náhodný, nemusí být tato metoda efektivní volbou. Ukázalo se, že optimální tolerance τ se nejčastěji pohybuje v rozmezí $10^{-4} - 10^{-2}$. Není tomu tak ale vždy, největší roli zde stále hraje zadaná úloha, jako ostatně ve všech variantách. Dalším velkým problémem je to, že nevíme předem, kolik bude potřeba paměti na uložení faktoru L .

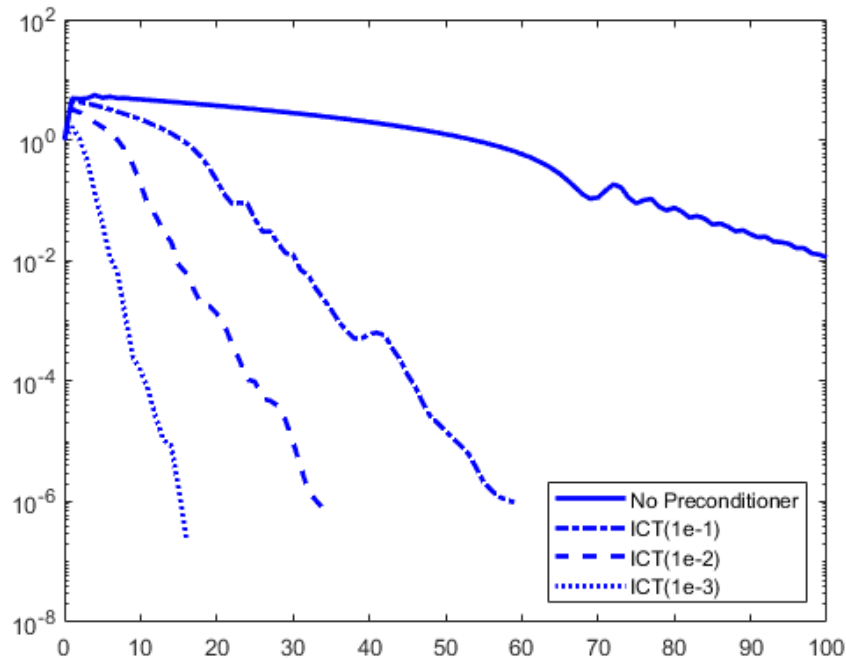
5.3.3 Současné praktické strategie

Zajímavějším přístupem jsou varianty vzniklé zkombinováním předešlých dvou pohledů, jak volit množinu S . Do této skupiny patří doposud nejméně úspěšnější varianty IC v současnosti, neboť se snaží vybalancovat klady a zápory těchto dvou pohledů.

Saad navrhl strategii zvanou duální prahová strategie (ang. dual threshold strategy) - $ICT(\tau,p)$, kde volíme na začátku procesu IC toleranci τ a nezáporné číslo p , které nám dovoluje mít v každém řádku faktoru L nejvíce p prvků. Myšlenka této metody je následující: Během eliminace každého řádku matice A vynuluj všechny prvky, jejichž absolutní hodnota je větší než $\tau\|\mathbf{a}_i\|_2$ a je-li počet zbývajících nenulových prvků v řádku větší než p , pak vyber z těchto prvků p největších v absolutní hodnotě a ostatní nastav na nuly. Pozor, diagonální prvky nikdy nenulujeme. V literatuře se často setkáme se značením $ICT(\tau)$, což označuje relativní nulování popsané výše.

Selže-li $ICT(\tau,p)$ pro dané parametry, pak lze výsledek zlepšit zvýšením čísla

p nebo snížením tolerance τ . Pro obtížné problémy lze vidět, že ke konvergování PCG (předpodmíněné metody sdružených gradientů) předpodmíněné maticí $M = LL^T$ získanou ICT(τ, p) dojde mnohem dříve než u PCG předpodmíněné maticí M z IC(0).



Obrázek 5.2: Konvergence PCG s ICT s různými hodnotami τ

Jiná varianta vzniklá z ICT(τ, p), která také pracuje se zadanými čísly τ a p , je založena na tom, že v každém řádku IC necháme aktualizovat všechny prvky, jejichž pozice se shodují s pozicemi nenulových prvků původní matice A a navíc dovolíme zaplnění řádku maximálně dalšími p nenulovými prvky, které byly stejně jako v ICT(τ, p) zvoleny podle své velikosti.

Jak už jsme konstatovali dříve, značným problémem těchto strategií je volba tolerance τ a to, že nevíme, jak velký bude faktor L . To vedlo Jonese a Plassmanna k navržení varianty, kde není potřeba čísla τ a je nám předem známo, kolik budeme potřebovat paměti k výpočtu L .

Nechť číslo n_k představuje počet nenulových prvků dolní trojúhelníkové části matice A v k -tém řádku. Pak stejně jako u IC(0) dovolíme, aby faktor L měl na každém řádku stejný počet nenul jako v A , avšak budeme ignorovat distribuci těchto nenulových prvků matice A a vybereme v každém kroku eliminace řádku n_k největších prvků v absolutní hodnotě. Bude potřeba tedy přesně tolik paměti jako u IC(0), a navíc se nám ve srovnání s IC(0) může zlepšit konvergence předpodmíněné úlohy, neboť už nebereme v potaz jen strukturu původní matice, ale také velikost jejích prvků.

Stejně jako jsme došli k vylepšení Saadovy duální prahové strategie, tak jde i zde poupravit metodu tak, že dovolíme v k -tém řádku $n_k + p$ nenulových prvků, kde p je nezáporné číslo. Pro $p > 0$ pak bude potřeba více místa na uložení matice L než u IC(0), avšak stále dokážeme předvídat kolik.

5.4 Nová varianta

Jak jsme již viděli v teoretické části práce, nejlepší výhled do budoucnosti mají právě varianty, jež zohledňují jak počet, tak i případně distribuci nenulových prvků v původní matici A a také velikost těchto prvků. Standardní postupy jsou často založeny na kombinaci tolerance τ a počtu prvků ve sloupci/řádku, který je určen uniformně vzhledem k počtu nenulových prvků v jednotlivých sloupcích/řádcích dolní trojúhelníkové části matice A (viz Jonesova a Plassmannova metoda). To dalo podnět k návrhu naší nové varianty, kde jsme se snažili o flexibilnější postup.

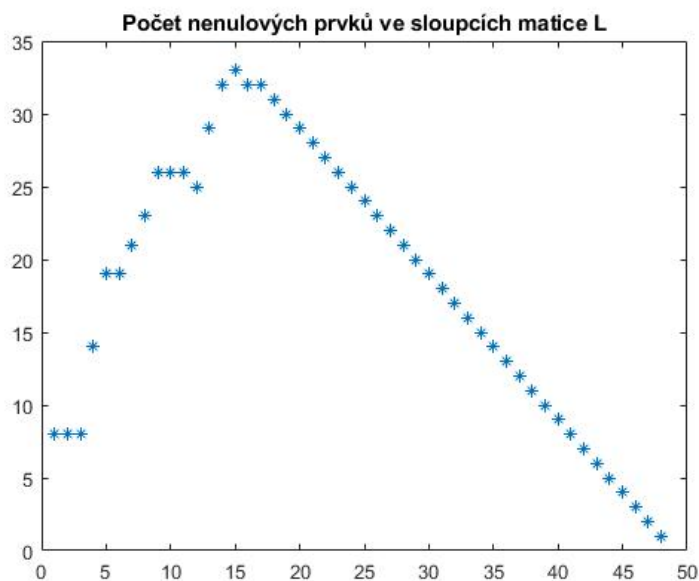
5.4.1 Popis nové varianty

Na začátku si zvolíme nějakou kladnou konstantu p a drop toleranci τ . V novém faktoru W budeme mít předem určený počet nenulových prvků (v praxi se bude hodnota lišit jen o pár čísel v případech, kde nám někde plánované zaplnění nevznikne)

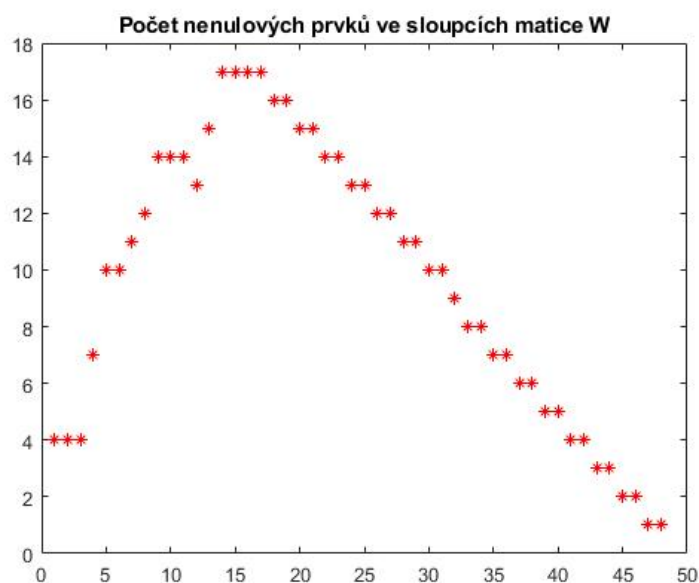
$$nnz(W) = nnz(B) + np, \quad (5.8)$$

kde B je dolní trojúhelníková část matice A (typu $n \times n$), $nnz(B)$ je počet jejích nenulových prvků a $nnz(W)$ počet nenulových prvků matice W . Novou variantu jsme navrhli pro sloupcový i řádkový přístup. V dalším odstavci budeme popisovat sloupcový přístup, řádkový by se dělal analogicky.

První rozdíl nového postupu (oproti Jonesově a Plassmannově metodě) už spočívá v tom, že nyní budeme srovnávat počty nenulových prvků ve sloupcích nového faktoru W přímo se strukturou faktoru L , jenž vzniká úplnou Choleského faktorizací. Dále chceme nenulové prvky v matici W rozdělit nikoli uniformně, ale proporcionálně podle počtů prvků ve sloupcích přesného faktoru L . Tyto počty je možné získat pro SPD matice velmi levně (viz [4, 5]). Problémem jejich výpočtu, za kterým stojí celá teorie řídkých rozkladů, se zde ale nebudeme zabývat. V závěru tedy chceme dostat matici W takovou, že graf funkce, která přiřazuje jednotlivým sloupcům matice W jejich počet nenul, se tvarově podobá grafu téže funkce aplikované na matici L , jen je na ose y jinak škálován.



Obrázek 5.3: Rozložení prvků faktoru L vzniklé z matice bcsstk01



Obrázek 5.4: Rozložení prvků nového faktoru W vzniklé z matice bcsstk01

Nulování prvků probíhá klasicky srovnáním s číslem τ a používáme k tomu absolutní nulování, tj.

$$a_{ij} = \begin{cases} 0 & \text{pokud } |a_{ij}| \leq \tau \\ a_{ij} & \text{jinak.} \end{cases}$$

Funkce VyberRadek na konci každého k -tého cyklu algoritmu vezme spočítaný k -tý řádek matice W , vybere z něho maximálně r_k největších prvků v absolutní hodnotě a ostatní nastaví na nuly. Zpátky tedy dostaneme nový k -tý řádek matice W , který v dalších krocích už nevystavujeme změnám. Samotný algoritmus rozkladu použitý na experimentech je založen na submaticovém způsobu faktorizace - viz 1.kapitola (Způsoby Choleského faktorizace).

Algorithm 4 Nová varianta IC

```
1: input  $A = (a_{ij}) \in \mathbb{R}^{n \times n} - SPD, p > 0, \tau$ 
2:  $nenulL = nnz(L)$ 
3:  $nenulB = nnz(B)$ 
4:  $N = nenulB + np$ 
5: for  $i = 1 : n$  do
6:    $l_i = nnz(L(:, i))$ 
7:    $r_i = \frac{l_i N}{nenulL}$ 
8: end for
9:  $W = A$ 
10: for  $k = 1 : n$  do
11:    $w_{k,k} = \sqrt{w_{k,k}}$ 
12:   for  $m = k + 1 : n$  do
13:      $w_{k,m} = \frac{w_{k,m}}{w_{k,k}}$ 
14:     if  $|w_{k,m}| \leq \tau$  then
15:        $w_{k,m} = 0$ 
16:     end if
17:   end for
18:   for  $i = k + 1 : n$  do
19:      $e = w_{i,k}/w_{k,k}$ 
20:     for  $j = k : n$  do
21:        $w_{i,j} = w_{i,j} - ew_{k,j}$ 
22:       if  $|w_{i,j}| \leq \tau$  then
23:          $w_{i,j} = 0$ 
24:       end if
25:     end for
26:   end for
27:    $W(k, :) = VyberRadek(W(k, :), r_k, n)$ 
28: end for
29:  $W = W^T$ 
```

5.4.2 Výsledky zkoumání

Testy jsme prováděli na SPD maticích z kolekce řídkých matic z [3], a to konkrétně s maticemi bcsstk01, bcsstk03, bcsstk04, bcsstk05, bcsstk22, lund_a a mesh1e1, které všechny pocházejí ze strukturálních problémů. Zkoumali jsme čtyři indikátory:

- počet nenulových prvků faktoru L vzniklého danou variantou IC
- přesnost faktoru L , tj. $\|A - LL^T\|_F$
- stabilitu L , tj. $\|I - L^{-1}AL^{-T}\|_F$
- konvergenci PCG s maticí předpodmínění $M = LL^T$.

Nová sloupcová varianta, kterou jsme nazvali NCIC, byla srovnána v těchto čtyřech ohledech se standardní Choleského faktorizací, IC(0), IC(1), ICT(τ) a její řádkovou verzí (NRIC).

Doposud nejlepší výsledky prokázala NCIC s volbou $p = 7$, $\tau = 10^{-2}$ u matice bcsstk22, kde překonala ostatní varianty neúplné Choleského faktorizace jak v přesnosti a stabilitě faktoru L , tak v rychlosti konvergence PCG. Přitom podíl počtu nenul v L a B se pohyboval stále pod 2. Výsledky NRIC dopadly pro $p = 7$ úplně stejně jako u NCIC.

Tabulka 5.1: Testování metod IC na matici bcsstk22

Metoda	nnz(L)	přesnost	stabilita
Choleského faktorizace	1365	$1.9801 * 10^{-9}$	$8.3286 * 10^{-14}$
IC(0)	417	$6.5102 * 10^5$	5.4803
IC(1)	506	$2.7931 * 10^5$	2.7839
ICT(10^{-5})	784	120.3503	0.0050
ICT(10^{-6})	882	11.2618	$5.7054 * 10^{-4}$
NCIC(7, 10^{-2})	788	38.9138	0.0020
NCIC(7, 10^{-3})	875	3.1718	$2.4383 * 10^{-4}$

Hodnoty v Tabulce 5.1 ukazují, že při standardní Choleského faktorizaci dochází k obrovskému zaplnění faktoru L a u IC(0) a IC(1) je zase problémem nepřesnost matice předpodmínění. Je tedy potřeba najít lepší metody. ICT a NCIC jsou v tomto konkrétním případě konkurenti. Zkusme se na ně podívat podrobněji. U ICT uvažujme dvě tolerance: 10^{-5} a 10^{-6} . U NCIC vezměme číslo zaplnění $p = 7$ v kombinaci se dvěma tolerancemi 10^{-2} , 10^{-3} . Jak je vidět z Tabulky 5.1, srovnatelné jsou

(a) $\text{ICT}(10^{-5}) \times \text{NCIC}(7, 10^{-2})$

(b) $\text{ICT}(10^{-6}) \times \text{NCIC}(7, 10^{-3})$.

Zatímco je stabilita obou metod v (a) i (b) podobná, přesnost NCIC je v (a) i (b) výrazně lepší nežli u ICT.

Tabulka 5.2: Konvergence PCG na SLR s maticí bcsstk22 a jednotkovou pravou stranou

Typ předpodmínění	počet iterací	hustota
IC(0)	35	1
IC(1)	23	1.2134
ICT(10^{-5})	3	1.8801
NCIC(7, 10^{-2})	3	1.8897
ICT(10^{-6})	2	2.1151
NCIC(7, 10^{-2})	2	2.0983

Hustotou se myslí číslo $\frac{\text{nnz}(L)}{\text{nnz}(B)}$.

Příslušné varianty ICT a NCIC konvergují v našem případě podobně rychle. Předpokládáme ale, že přesnost, která je výrazně lepší u NCIC, se u vyšších dimenzí matic projeví. Navrhli jsme tedy novou metodu, která (podle našich omezených experimentů) se při předpodmínění chová podobně jako ICT, ale rozkládá matici bližší původní matici.

U ostatních matic byly NCIC a NRIC horší než ICT a jen o trošku lepší než IC(0) a IC(1). V případech, kdy jsme už dostali celkem dobrou aproximaci Choleského rozkladu matice A , byl faktor L už příliš zaplněný. Ukázalo se, že nejlepší volba p jsou čísla 5,6,7 a τ je nejlepší volit v rozmezí $10^{-2} - 10^{-5}$.

Celkově hodnotíme výsledky experimentů, kde jsme ukázali, že nový přístup může být výhodný, jako povzbuzující z následujících důvodů.

Zřejmá tendence používání neúplného rozkladu v praxi totiž jednoznačně vede směrem k adaptivním metodám. Jsme-li tak schopni v průběhu výpočtu řešení poznat, že předpokládání nemusí být optimální, můžeme rozklad modifikovat, například změnou tolerance. Druhý důvod, proč je výhodné hledat nové postupy, je dán tím, že předpokládání mohou mít podobné vlastnosti pro celou třídu úloh z nějaké aplikace. Proto se vyplatí zkoušet vhodnost různých postupů v konkrétní aplikaci. Třetí důvod spočívá v tom, že praktické řešení nelineárních problémů vede často k celé sekvenci lineárních algebraických soustav. Nalezení vhodné varianty pro několik prvních soustav můžeme poté s výhodou uplatnit pro řešení celého nelineárního problému.

Závěr

V této bakalářské práci jsme se snažili představit neúplnou Choleského faktorizaci jako metodu využívanou k předpodmínění úloh s SPD maticemi. Tato metoda, jako ostatně mnoho dalších, ještě není dostatečně probádaná a lze zde stále využít autorské kreativity.

V práci jsme ze začátku shrnuli základní poznatky o standardní Choleského faktorizaci, kde jsme nejvíce času věnovali třem způsobům faktorizace a problému zaplnění faktoru L u řídkých matic. Jelikož se neúplné faktorizace hlavně používají ve spojení s iteračními metodami jako způsob předpodmínění původní zadané úlohy, zabývali jsme se v následujících třech kapitolách pojmy iterační metoda, CG a předpodmínění. Iterační metody jsme především srovnávali s přímými metodami. Zjistili jsme, že hlavní výhodou iteračních metod jsou levnější operace v průběhu výpočtu řešení, neboť nejsložitější operace v algoritmu je operace matice krát vektor. Naopak problémem byla nejistota jejich konvergence v rozumném čase. Jako příklad iterační metody, jež se hojně používá při řešení úloh s SPD maticemi, jsme uvedli metodu sdružených gradientů. Nejprve jsme odvodili její algoritmus, který spočíval v minimalizaci chyby aproximace přesného řešení v energetické normě A v každém iteračním kroku. Poté jsme krátce diskutovali o konvergenci CG. V další kapitole jsme představili pojem předpodmiňování. Řekli jsme si, že takto můžeme úlohu převést na jinou, která je mnohem lépe řešitelná a přitom je blízká naší původní úloze. Zároveň předpodmíněním iteračních metod zlepšíme rychlost jejich konvergence. Předpodmiňovat se dalo zleva, zprava a oboustranně. Jako příklad jsme si uvedli ILU a její speciální případ, IC. U SPD matic je vhodné použít matici předpodmínění M ve tvaru IC, kde přímo vidíme pozitivní definitnost matice $L^{-1}AL^{-T}$. Konečně jsme se tak dostali k hlavní části této práce, jež je právě neúplná Choleského faktorizace.

Nejprve jsme se zabývali existencí tohoto neúplného rozkladu. Zjistilo se, že existence je zajištěna pro symetrické M -matice, později dokonce pro SPD H -matice, a že obecně může IC selhat kvůli zlomu. Poté jsme si uvedli několik návrhů, jak této situaci předejít. Dále jsme představili dva základní algoritmy IC, jež jsou hlavním pilířem různých variant IC. Tyto varianty jsme rozdělili do tří skupin podle konkrétní volby množiny S . S každou skupinou jsme se nejprve krátce seznámili, a pak uvedli několik nejpoužívanějších příkladů.

U metod založených na struktuře matice A jsme si řekli, že pro malé stupně zaplnění je výpočet levný, avšak do potíží se dostáváme u matic daleko od diagonálně dominantních, protože zanedbáváme velikost nulovaných prvků. To vedlo k zavedení druhé skupiny, jež byla určena na základě zvolení vhodné tolerance. Uvedli jsme si dvě strategie a rozdíl mezi nimi. Nevýhodou tohoto postupu ale bylo, že předem nevíme, kolik paměti je potřeba na uložení faktoru L . Ve třetí skupině jsme si popsali některé současné praktické strategie. Nakonec jsme si sami navrhli novou metodu a vyzkoušeli si jí na několika řídkých maticích. Přitom jsme si zkusili na těchto příkladech i některé metody popsané v předchozích sekcích a výsledky experimentů jsme mezi sebou porovnali. Detailnější rozbor jsme pak udělali u matice `bcstk22`, kde naše nová varianta dosáhla skvělých výsledků a ukázala se jako dobrý způsob předpodmínění pro CG.

Díky tomuto výsledku hodnotíme novou variantu pozitivně, neboť funguje-li

tato metoda dobře na určitou matici, je možné, že je její použití výhodné pro celou třídu podobných úloh. Práce tak splnila svůj původní účel a dalo by se v ní pokračovat různými směry. Zaobírat bychom se mohli dalšími postupy předpokládající sdružené gradientů či jiných iteračních metod, nebo také hlouběji nahlédnout do samotné problematiky existence ILU a jejích řešení.

Literatura

- [1] M. Benzi. Preconditioning techniques for large linear systems: a survey. 182(2):418–477, 2002.
- [2] A. Berman and R. J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences*. Academic Press, New York, 1979.
- [3] T. A. Davis. *University of Florida Sparse Matrix Collection*. available online at <http://www.cise.ufl.edu/~davis/sparse/>, NA Digest, vol. 94, issue 42, October 1994.
- [4] Timothy A. Davis. *Direct methods for sparse linear systems*, volume 2 of *Fundamentals of Algorithms*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2006.
- [5] I. S. Duff, A. M. Erisman, and J. K. Reid. *Direct methods for sparse matrices*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York, second edition, 2017.
- [6] J. Duintjer Tebbens, I. Hnětynková, M. Plešinger, Z. Strakoš, and P. Tichý. *Analýza metod pro maticové výpočty. Základní metody*. Matfyzpress, MFF UK, 2012.
- [7] Ky Fan. Note on M -matrices. *Quart. J. Math. Oxford Ser. (2)*, 11:43–49, 1960.
- [8] A. George and J. W. H. Liu. *Computer Solution of Large Sparse Positive Definite Systems*. Prentice-Hall, Englewood Cliffs, NJ., 1981.
- [9] G. H. Golub and C. F. Van Loan. *Matrix Computations. 3rd edition*. The Johns Hopkins University Press, Baltimore and London, 1996.
- [10] D. H. Jacobson. Factorization of symmetric M -matrices. *Linear Algebra and its Applications*, 9:275–278, 1974.
- [11] Bishan Li. *Generalizations of diagonal dominance in matrix theory*. University of Regina, 1997.
- [12] J. Liesen and Z. Strakoš. *Krylov subspace methods*. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, 2013. Principles and analysis.
- [13] T. A. Manteuffel. An incomplete factorization technique for positive definite linear systems. 34:473–497, 1980.
- [14] J. A. Meijerink and H. A. van der Vorst. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M -matrix. 31(137):148–162, 1977.
- [15] Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition, 2003.

- [16] R. S. Varga. *Matrix Iterative Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey, 1962.

Seznam obrázků

1.1	sloupcová varianta	5
1.2	řádková varianta	5
1.3	submaticová varianta	6
5.1	Srovnání CG s PCG, kde jsme matici předpodmínění získali pomocí IC(0) a MIC(0)	21
5.2	Konvergence PCG s ICT s různými hodnotami τ	24
5.3	Rozložení prvků faktoru L vzniklé z matice bcsstk01	26
5.4	Rozložení prvků nového faktoru W vzniklé z matice bcsstk01	26

Seznam tabulek

5.1	Testování metod IC na matici bcsstk22	28
5.2	Konvergence PCG na SLR s maticí bcsstk22 a jednotkovou pravou stranou	28