



**MATEMATICKO-FYZIKÁLNÍ
FAKULTA**
Univerzita Karlova

BAKALÁŘSKÁ PRÁCE

Alena Jarolímová

**Aproximace pomocí matic nízkých
hodností**

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

Děkuji profesoru Tůmovi za trpělivost, ochotu a cenné rady. Také děkuji svému příteli Janu Vainerovi a své rodině za jejich podporu.

Název práce: Aproximace pomocí matic nízkých hodnotí

Autor: Alena Jarolímová

Katedra: Katedra numerické matematiky

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

Abstrakt: Práce je zaměřena na použití matic s nízkou hodnotí v numerické matematice. Nejprve uvádíme metodu sdružených gradientů a její předpokmí-nění, které pak využíváme v dalších částech. Následně popisujeme čtyři různé způsoby aproximace pomocí matic nízké hodnoti. Uvádíme zde klasickou apro-ximaci pomocí singulárního rozkladu. Dále na modelovém příkladu popisujeme hierarchické matice, které jsou úzce propojené s aplikacemi ve fyzice a technice. Následně se v kapitole o algebraických přístupech věnujeme pseudo-skeletnímu rozkladu. Uvedeme a dokážeme větu o odhadu chyby tohoto rozkladu a zmí-níme také algoritmus Maxvol, pomocí kterého je možné pseudo-skeletní rozklad spočítat pro úzké matice. Další část věnujeme pravděpodobnostním přístupům a řešiči problému nejmenších čtverců Blendenpik. Nakonec popíšeme výsledky experimentů zaměřených na předpokmí-nění pomocí algoritmu Maxvol.

Klíčová slova: hodnost matice, soustavy lineárních rovnic, maticové faktorizace, řídkost matice

Title: Low-rank matrix approximations

Author: Alena Jarolímová

Department: Department of Numerical Mathematics

Supervisor: prof. Ing. Miroslav Tůma, CSc., Department of Numerical Mathema-tics

Abstract: This thesis is focused on using low rank matrices in numerical mathe-matics. We introduce conjugate gradient method and its preconditioning which we use in other chapters. Then we describe four different approaches to appro-ximation using low rank matrices. First we discuss classical approximation using singular value decomposition. Next, using a model problem, we describe hierar-chical matrices, which are connected with applications in physics and technique. Then pseudo-skeleton decomposition is introduced. We formulate and prove a theorem about error estimate of this decomposition. We also mention algorithm Maxvol which can compute pseudo-skeletal decomposition of tall matrices. Next chapter is dedicated to probabilistic algorithms and to least-squares solver Blen-denpik. In conclusions we show results of experiments focused on preconditioning using algorithm Maxvol.

Keywords: matrix rank, systems of linear equations, matrix factorizations, matrix sparsity

Obsah

Úvod	2
1 Iterační metody	3
1.1 Sdružené gradienty	3
1.2 Předpokládání	4
2 Klasická aproximace	6
2.1 Odvození singulárního rozkladu	6
2.2 Využití singulárního rozkladu k aproximaci	7
2.3 Výpočet singulárního rozkladu	8
3 Aproximace motivovaná aplikacemi	9
3.1 Modelový příklad	9
3.1.1 Převedení na soustavu lineárních rovnic	9
3.1.2 Aproximace jádra $\log x-y $	10
3.1.3 Aproximace bloků matice	11
3.1.4 Dělení matice	11
3.1.5 Hierarchická matice	12
3.2 Operace s H-maticemi	13
3.2.1 Sčítání	14
3.2.2 Násobení a inverze	14
4 Algebraické přístupy	15
4.1 Motivace	15
4.2 Odhady pro pseudo-skeletní aproximaci	15
4.3 Algoritmus Maxvol	18
5 Pravděpodobnostní přístupy	20
5.1 Hlavní myšlenky	20
5.2 Blendepik	21
5.2.1 Problém nejmenších čtverců	21
5.2.2 Koherence	21
5.2.3 Algoritmus	22
6 Experimenty	23
6.1 Problém nejmenších čtverců	23
6.2 Soustava normálních rovnic	23
6.3 Předpokládání pomocí Maxvol	25
6.4 Realizace	25
6.4.1 Regularizace	25
6.4.2 Funkce Maxvol	25
6.5 Výsledky	26
Závěr	29
Seznam použité literatury	30

Úvod

Popisování fyzikálních jevů v přírodních vědách vede velmi často na problém řešení integrálních nebo parciálních diferenciálních rovnic. Většinou není možné najít řešení analyticky, je proto třeba používat numerické metody. Mezi některé z nich patří například metoda konečných diferencí či metoda konečných prvků. Tyto metody převádějí původní problém na řešení soustavy lineárních rovnic, kterou je třeba vyřešit pomocí počítače. Matice těchto soustav mohou dosahovat velkých rozměrů a je tedy třeba optimalizovat využití paměti a časovou náročnost výpočtu. Hlavním nástrojem pro řešení tohoto problému je ale návrh a analýza nových algoritmů.

Základní numerické metody pro obecnou matici, jako jsou Gaussova eliminace nebo QR rozklad, mají kubickou složitost a jejich použití tedy není moc praktické pro matice velkých rozměrů. Je třeba používat metody sofistikovanější. Jednou z možností, jak zmenšit časové a paměťové nároky metody, je věnovat pozornost struktuře problému. Ta se na soustavě rovnic může projevit například řídkostí matice.

Je třeba rozlišovat dva typy řídkosti. První z nich je strukturální řídkost. Matice A je strukturálně řídká, pokud je počet jejích nenulových prvků podstatně menší než její celkový počet prvků. Matice tohoto typu je výhodné reprezentovat v jiném formátu než matice obecné. Jednou z možností je například ukládat pouze hodnoty nenulových prvků a jejich souřadnice. Pokud má matice dostatečně nízký počet nenulových prvků, velmi se snižují paměťové nároky. Násobení maticí je také podstatně rychlejší, než v obecném případě. To vede k alternativní definici strukturální řídkosti. Matice A je strukturálně řídká, pokud je výhodné ji reprezentovat v řídkém formátu. Více o reprezentaci v řídkém formátu lze najít například v [12].

Druhým typem řídkosti je řídkost datová. Matice tohoto typu se vyznačují tím, že jejich bloky nebo i celou matici lze velmi dobře aproximovat pomocí matic nízké hodnoty. Tento přístup také velmi snižuje paměťové nároky. Místo celého bloku matice je třeba uložit pouze dvě matice nízké hodnoty, jejichž součin je dobře aproximuje. I v tomto případě dochází nejenom ke snížení nároků paměťových, ale i časových.

V této práci se budeme věnovat hlavně datové řídkosti a způsobům, jak matici aproximovat pomocí matic nízké hodnoty. Kapitoly 2-5 se věnují různým přístupům k této problematice. První z těchto částí věnujeme klasické aproximaci pomocí singulárního rozkladu. Pak následuje část o hierarchických maticích, které jsou vhodné například pro řešení integrálních rovnic. Ve čtvrté kapitole se budeme věnovat pseudo-skeletnímu rozkladu, který je základem pro naše experimenty. Poslední ze zde zmíněných přístupů je založen na pravděpodobnosti. To je vhodné například pro aproximaci oboru hodnot matice. Kapitulu 1 jsme věnovali zejména metodě sdružených gradientů, která je důležitou součástí našich experimentů.

1. Iterační metody

Důležitou skupinou metod používaných v numerické matematice jsou iterační metody. Jejich společným rysem je počítání přibližného řešení x_i v závislosti na předchozích iteracích, dokud není splněno jejich zastavovací kritérium. Klasické iterační metody pro řešení soustav lineárních rovnic jsou založeny na štěpení matice soustavy a patří mezi ně například Jacobiho metoda a Gauss-Seidelova metoda. Další rozsáhlou skupinou iteračních metod jsou Krylovovské metody. Do této skupiny patří například sdružené gradienty, které použijeme v experimentech.

1.1 Sdružené gradienty

Metoda sdružených gradientů se používá pro řešení soustav lineárních rovnic

$$Ax = b \tag{1.1}$$

s reálnou symetrickou pozitivně definitní maticí A a patří mezi Krylovovské iterační metody. Jednou z jejích výhod je, že nemá velké paměťové nároky. V paměti počítače je při výpočtu potřeba udržovat pouze čtyři vektory. V každé iteraci je nejnáročnější operací násobení matice A vektorem. Díky tomu nejsou sdružené gradienty ani výpočetně náročné.

Sdružené gradienty odvodíme pomocí minimalizace kvadratického funkcionálu jako v [5]. Uvažujme funkcionál

$$\mathcal{F}(x) = \frac{1}{2}x^T Ax - x^T b. \tag{1.2}$$

Z postačující podmínky pro extrém pak získáme

$$0 = \nabla \mathcal{F}(x) = Ax - b.$$

Z této úvahy plyne, že řešení soustavy (1.1) je ekvivalentní minimalizaci funkcionálu (1.2). Uvážíme-li aproximaci řešení x_k , můžeme $\mathcal{F}(x_k)$ využitím (1.1) napsat ve tvaru

$$\mathcal{F}(x_k) = \frac{1}{2}\|x - x_k\|_A^2 - \frac{1}{2}\|x\|_A^2,$$

kde $\|\cdot\|_A$ je tzv. A-norma definovaná předpisem

$$\|z\|_A^2 = z^T Az.$$

Označme rezidua $r_k = Ax_k - b$. Posloupnost řešení x_k je konstruována podle předpisu

$$x_k = x_{k-1} + \gamma_{k-1}p_{k-1}, \quad k = 1, 2, \dots,$$

kde

$$p_k = r_k + \delta_k p_{k-1}$$

je směrový vektor. Skalár γ_{k-1} je možné dopočítat z podmínky minimalizace A-normy chyby v každé iteraci. Hodnota δ_k plyne z podmínky A-ortogonalit vektorů p_k . Celý algoritmus pak vypadá následovně.

Algoritmus 1: Metoda sdružených gradientů

Input: A, b, x_0
 $r_0 := b - Ax_0$
 $p_0 := r_0$
for $k=1, 2, \dots$ **do**
 $\gamma_{k-1} := \frac{r_{k-1}^T r_{k-1}}{p_{k-1}^T A p_{k-1}}$
 $x_k := x_{k-1} + \gamma_{k-1} p_{k-1}$
 $r_k := r_{k-1} - \gamma_{k-1} A p_{k-1}$
 $\delta_k := \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}$
 $p_k := r_k + \delta_k p_{k-1}$
end

1.2 Předpokládání

Pro sdružené gradienty je možné odhadnout relativní chybu pomocí

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

kde $\kappa(A)$ je číslo podmíněnosti matice A . I když je závislost rychlosti konvergence na matici složitější a do hry vstupují i zaokrouhlovací chyby (viz [11]), protože sdružené gradienty jsou silně nelineární metodou, odhad napovídá, že malé číslo podmíněnosti matice může být užitečné. To nás vede k možnosti řešit místo původní rovnice rovnici předpokládání.

Uvažujme regulární matici M . Vynásobíme-li soustavu $Ax = b$ maticí M^{-1} zleva, získáme soustavu

$$M^{-1}Ax = M^{-1}b, \quad (1.3)$$

která má stejné řešení jako soustava původní. Pokud zvolíme M tak, aby $M \sim A$, bude platit $M^{-1}A \sim I$ a tedy i $\kappa(M^{-1}A) \sim 1$. To pro soustavu (1.3) může vést k rychlé konvergenci sdružených gradientů. Tento postup se nazývá levé předpokládání.

Další možností je pravé předpokládání, které spočívá řešení soustav

$$AM^{-1}y = b, \quad Mx = y. \quad (1.4)$$

Z podobných důvodů jako výše pak konverguje metoda sdružených gradientů pro první ze soustav v (1.4).

Je také možné použít levé i pravé předpokládání zároveň. Najdeme-li vhodné matice M_1 a M_2 , aby $M_1 M_2 \sim A$, pak platí $M_1^{-1} A M_2^{-1} \sim I$ a předpokládání soustava má tvar

$$M_1^{-1} A M_2^{-1} y = M_1^{-1} b, \quad M_2 x = y.$$

Zabývejme se nyní levým předpokládáním, které je vhodné pro metodu sdružených gradientů. Zde je třeba dodat, že pro metodu sdružených gradientů je potřeba, aby matice soustavy byla symetrická a pozitivně definitní, což v případě (1.3) nemusí vůbec platit. Formálně by bylo tedy třeba matici M na soustavu aplikovat takto

$$M^{-\frac{1}{2}} A M^{-\frac{1}{2}} y = M^{-\frac{1}{2}} b, \quad M^{\frac{1}{2}} x = y.$$

Počítat $M^{\frac{1}{2}}$ je však výpočetně náročné a rezidua, která získáme z takto předpodmíněné metody sdružených gradientů, se nebudou rovnat $b - Ax_k$. Z těchto důvodů se používá předpodmínění ve tvaru (1.3), což odpovídá změně skalárního součinu v metodě.

Zbývá vyřešit volbu matice předpodmínění. Nejrychlejší konvergence sdružených gradientů by bylo dosaženo pro $M = A$. To by ale bylo ve výsledku stejně výpočetně náročné jako řešení původní soustavy $Ax = b$. Je tedy potřeba najít matici M , jejíž výpočet nebude příliš náročný. Výpočetně nejjednodušším by bylo zvolit $M = I$. Toto předpodmínění by ale nijak nepomohlo urychlit konvergenci. Je tedy třeba volit M někde mezi těmito extrémy. Například v případě diagonálně dominantní matice A je možností volby M diagonální matice tvořená diagonálními prvky matice A . Takové předpodmínění se nazývá Jakobiho. Jedním z nejpoužívanějších oboustranných předpodmínění je neúplný Choleského rozklad. Následující algoritmus pro jeho výpočet je z [5].

Algoritmus 2: Neúplný Choleského rozklad

```

Input: A
C = tril(A)
for k=1,...,n do
    ck,k := √ck,k
    for i=k+1,...,n do
        if ci,k ≠ 0 then
            ci,k := ci,k / ck,k
        end
        for j=k+1,...,i do
            if ci,j ≠ 0 then
                ci,j := ci,j - ci,kc̄j,k
            end
        end
    end
end
end

```

V této práci budeme používat předpodmíněnou metodu sdružených gradientů a neúplný Choleského rozklad v kapitole 6 věnované experimentům.

2. Klasická aproximace

Jedním ze základních nástrojů, které je možno použít pro účely aproximace, je singulární rozklad. Ten spočívá v rozložení obecné komplexní obdélníkové matice na součin dvou unitárních matic a jedné obdélníkové diagonální matice. Ze singulárního rozkladu matice je možné vyčíst například hodnotu matice, její normu nebo bázi jádra, či oboru hodnot. Výpočet tohoto rozkladu je ale obvykle časově náročný.

2.1 Odvození singulárního rozkladu

Toto odvození singulárního rozkladu vychází z [5] a částečně také z [2], [6]. Důležitou roli při odvození spektrálního rozkladu matice A hrají spektrální rozklady matic A^*A a AA^* .

Uvažujme obecnou obdélníkovou matici $A \in \mathbb{C}^{n \times m}$. Matice A^*A a AA^* jsou pak čtvercové hermitovské pozitivně semidefinitní matice. Podle spektrální věty pro hermitovské pozitivně definitní matice z [2] je lze napsat jako součin matic UDU^* , kde U je matice unitární a D je matice $\text{diag}(\lambda_1, \dots, \lambda_k)$, $\lambda_i \geq 0$ pro každé $i \in \{1, \dots, k\}$. Zároveň také platí, že i -tý sloupec matice U je vlastní vektor příslušný vlastnímu číslu λ_i . Tento rozklad se nazývá spektrální.

Použijeme-li tedy spektrální rozklad na matici A^*A , pak s využitím definice vlastního čísla dostaneme vztah

$$A^*Av_j = \lambda_j v_j, \quad j = 1, \dots, m, \quad (2.1)$$

kde λ_j jsou vlastní čísla A^*A a v_j jsou k nim příslušné vlastní vektory. Předpokládejme, že

$$\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_r > 0, \quad \lambda_{r+1} = \dots = \lambda_m = 0. \quad (2.2)$$

Označíme-li $V = (v_1, \dots, v_m)$, pak platí

$$A^*A = VDV^*, \quad D = \text{diag}(\lambda_1, \dots, \lambda_r, 0, \dots, 0).$$

Následující lemma popisuje vztah mezi v_j a vlastními vektory matice AA^* .

Lemma 1. *Uvažujme spektrální rozklad matice A^*A se značením zavedeným v (2.1) a (2.2). Potom jsou vektory*

$$u_j \equiv Av_j / \sqrt{\lambda_j}, \quad j = 1, \dots, r,$$

ortonormální vlastní vektory matice AA^ a platí*

$$AA^*u_j = \lambda_j u_j, \quad j = 1, \dots, r.$$

Důkaz tohoto lemmatu je založen na vynásobení vztahu (2.1) maticí A zleva. Z definice vlastního čísla pak plyne, že Av_j je vlastní vektor AA^* příslušný vlastnímu číslu λ_j . Norma Av_j je rovna $\sqrt{\lambda_j}$, což důkaz dokončuje. Jeho celé znění je obsaženo v [5].

Z lemmatu 1 plyne, že

$$Av_j = \sqrt{\lambda_j}u_j, \quad j = 1, \dots, r. \quad (2.3)$$

Označíme-li $U = (u_1, \dots, u_n)$, pak platí

$$AA^* = UDU^*, \quad D = \text{diag}(\lambda_1, \dots, \lambda_r, 0, \dots, 0).$$

Definice 1. Buď $A \in \mathbb{C}^{n \times m}$. Odmocniny z nenulových vlastních čísel matice A^*A se nazývají singulární čísla matice A ,

$$\sigma_j = \sqrt{\lambda_j}, \quad j = 1, \dots, r, \quad r = \text{rank}(A).$$

Vztah (2.3) lze tedy napsat ve tvaru

$$Av_j = \sigma_j u_j, \quad j = 1, \dots, r.$$

Po zapsání do maticového tvaru dostáváme požadovaný rozklad.

Věta 2. Pro každou matici $A \in \mathbb{C}^{n \times m}$ hodnosti r existují unitární matice $U \in \mathbb{C}^{n \times n}$ a $V \in \mathbb{C}^{m \times m}$ a kladná čísla $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ tak, že platí

$$A = U\Sigma V^*,$$

kde

$$\Sigma = \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} \in \mathbb{C}^{n \times m}, \quad \Sigma_r = \text{diag}(\sigma_1, \dots, \sigma_r) \in \mathbb{C}^{r \times r}$$

Singulární rozklad je také možné zapsat pomocí dyadického rozvoje

$$A = \sum_{j=1}^r \sigma_j u_j v_j^*. \quad (2.4)$$

Dalším možným způsobem zápisu je tak zvaný ekonomický singulární rozklad

$$A = U_r \Sigma_r V_r^*, \quad \Sigma_r = \text{diag}(\sigma_1, \dots, \sigma_r), \quad U_r = (u_1, \dots, u_r), \quad V_r = (v_1, \dots, v_r).$$

Ten je výhodný, protože v aplikacích není potřeba ukládat nadbytečné části unitárních matic U a V .

2.2 Využití singulárního rozkladu k aproximaci

Budeme se nyní zabývat otázkou, jak k matici A hodnosti r najít aproximaci hodnosti nejvýše $k < r$. Nejprve je třeba definovat normu na prostoru matic.

Definice 2. Funkcionál $\|\cdot\| : \mathbb{C}^{n \times m} \rightarrow \mathbb{R}$ definovaný předpisem

$$\|A\| = \max_{\|x\|_2=1} \|Ax\|_2, \quad A \in \mathbb{C}^{n \times m}$$

nazveme spektrální normou.

Jako aproximaci matice A hledáme tedy matici B_k tak, aby platilo

$$\text{rank}(B_k) \leq k, \quad \|A - B_k\| = \min_{\substack{X \in \mathbb{C}^{n \times m} \\ \text{rank}(X) = k}} \|A - X\|. \quad (2.5)$$

Nyní použijeme dyadický rozvoj (2.4) matice A . Každý z jeho sčítanců je součinem skaláru σ_i a dvou jednotkových vektorů. V normě je tedy každý z nich roven σ_i . Matice U a V jsou unitární, platí tedy $u_i \perp u_j$ a $v_i \perp v_j, i \neq j$. Je tedy zřejmé, že počet sčítanců určuje hodnotu výsledné matice. Proto můžeme pro hledanou matici B_k použít nejvýše k členů. Naším cílem je minimalizovat chybu aproximace. Zvolíme tedy

$$B_k = \sum_{j=1}^k \sigma_j u_j v_j^*,$$

protože σ_i jsou seřazená podle velikosti. Kvalitu této aproximace potvrzuje následující věta z [5].

Věta 3. (Eckart, Young, Mirsky) *Nechť je dána matice $A \in \mathbb{C}^{n \times m}$ hodnosti r a nechť k je přirozené číslo, $k < r$. Uvažujme singulární rozklad matice A zapsaný pomocí dyadického rozvoje (2.4). Potom je matice*

$$A^{(k)} = \sum_{j=1}^k \sigma_j u_j v_j^*$$

nejlepší aproximací matice A hodnosti k ve smyslu (2.5) a platí

$$\min_{\substack{X \in \mathbb{C}^{n \times m} \\ \text{rank}(X) = k}} \|A - X\| = \|A - A^{(k)}\| = \sigma_{k+1}.$$

2.3 Výpočet singulárního rozkladu

Počítání singulárního rozkladu souvisí s hledáním vlastních čísel matic A^*A a AA^* . Vlastní čísla matice je možné spočítat jako kořeny jejího charakteristického polynomu. Pro polynomy stupně 5 a vyššího bohužel neexistuje způsob, jak obecně najít přesné kořeny. Pro matice větších rozměrů je tedy potřeba využít numerických metod pro aproximaci vlastních čísel. To samé platí i pro singulární rozklad. Na jeho výpočet se používá Golub-Kahanova iterační bidiagonalizace. Tato iterační metoda je však velmi časově náročná. V případě čtvercové matice $A \in \mathbb{C}^{n \times n}$ je pro výpočet singulárních čísel potřeba více než $\frac{8}{3}n^3$ operací a pro výpočet celého singulárního rozkladu je potřeba dokonce více než $\frac{16}{3}n^3$ operací. Aproximace pomocí singulárního rozkladu tedy není vhodná pro matice velkých rozměrů.

3. Aproximace motivovaná aplikacemi

Mnoho numerických metod pro řešení parciálních diferenciálních nebo integrálních rovnic převádí původní problém na soustavu lineárních rovnic. Jsou to například metoda konečných prvků a metoda hraničních prvků. Struktura původního problému se zde projeví na matici soustavy. Díky tomu je vhodné zavést a studovat tzv. \mathcal{H} -matice, které právě strukturu problému využívají.

3.1 Modelový příklad

Pro ilustraci postupu řešení zde uvedeme konkrétní příklad problému. Budeme postupovat obdobným způsobem jako v [4].

Uvažujme integrální rovnici

$$\int_0^1 \log|x-y|U(y)dy = F(x), \quad x \in [0,1], \quad (3.1)$$

kde $F : [0,1] \rightarrow \mathbb{R}$ je vhodná pravá strana a hledáme řešení $U : [0,1] \rightarrow \mathbb{R}$.

3.1.1 Převedení na soustavu lineárních rovnic

Pro diskretizaci rovnice použijeme Galerkinovu metodu. Ta spočívá v projekci rovnice na n -dimenzionální prostor funkcí V_n s bází $\{\varphi_0, \dots, \varphi_{n-1}\}$. Obě strany rovnosti (3.1) vynásobíme $\varphi_i(x)$ a zintegrujeme přes interval $(0,1)$ pro každou bázovou funkci $\varphi_i, i = 1, \dots, n-1$. Tím získáme rovnost

$$\int_0^1 \int_0^1 \varphi_i(x) \log|x-y|U(y)dydx = \int_0^1 \varphi_i(x)F(x)dx, \quad 0 \leq i < n, \quad (3.2)$$

Můžeme si povšimnout, že pokud najdeme U , které splňuje (3.1), pak určitě splňuje i (3.2). Obrácená implikace ale obecně neplatí. Rovnost (3.2) se proto nazývá slabá formulace rovnice (3.1).

Na V_n promítneme také U a novou hledanou funkci označíme U_n . Z vlastností báze prostoru platí, že existují koeficienty u_j , tak že $U_n = \sum_{j=0}^{n-1} u_j \varphi_j$. Po dosazení do (3.2) dostáváme

$$\sum_{j=0}^{n-1} \int_0^1 \int_0^1 \varphi_i(x) \log|x-y|\varphi_j(y)dydx = \int_0^1 \varphi_i(x)F(x)dx, \quad 0 \leq i < n,$$

což lze přepsat jako soustavu lineárních rovnic

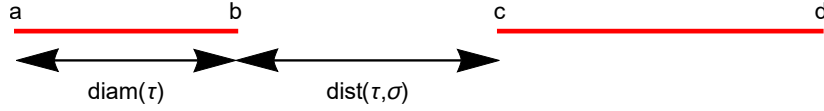
$$Gu = f, \quad G_{ij} = \int_0^1 \int_0^1 \varphi_i(x) \log|x-y|\varphi_j(y)dydx, \quad f_i = \int_0^1 \varphi_i(x)F(x)dx. \quad (3.3)$$

Nyní upřesníme volbu báze $\{\varphi_0, \dots, \varphi_{n-1}\}$. Pro tento konkrétní případ volme

$$\varphi_i(x) = \begin{cases} 1 & \text{pokud } \frac{i}{n} \leq x \leq \frac{i+1}{n} \\ 0 & \text{jinak} \end{cases}$$

Díky tomu se vztahy z (3.3) zjednoduší do podoby

$$G_{ij} = \int_{\frac{i}{n}}^{\frac{i+1}{n}} \int_{\frac{j}{n}}^{\frac{j+1}{n}} \log|x-y|dydx, \quad f_i = \int_{\frac{i}{n}}^{\frac{i+1}{n}} F(x)dx.$$



Obrázek 3.1: Podmínka přípustnosti

3.1.2 Aproximace jádra $\log|x-y|$

Matice G je v tomto případě hustá, lze ji však aproximovat maticí \hat{G} , která je datově řídká, s využitím Taylorova rozvoje funkce $\log|x-y|$. Pro $x=y$ funkce $g(x,y) := \log|x-y|$ diverguje, budeme proto uvažovat intervaly $\tau := [a, b] \subset [0, 1]$ a $\sigma := [c, d] \subset [0, 1]$ takové, že $\tau \cap \sigma = \emptyset$. Díky této podmínce máme zajištěno, že $g(x,y)$ lze parciálně derivovat podle x všude v $\tau \times \sigma$. Střed Taylorovy řady zvolme jako $x_0 := \frac{a+b}{2}$. Označme $g_k(x) := (x-x_0)^k$ a $h_k(y) := \frac{1}{k!} \partial_x^k g(x,y)$. Funkci $g(x,y)$ budeme pro $m \in \mathbb{N}$ aproximovat Taylorovým polynomem

$$\begin{aligned} \hat{g}(x,y) &= \sum_{k=0}^{m-1} \frac{1}{k!} \partial_x^k g(x,y) (x-x_0)^k \\ &= \log|x_0-y| + \sum_{k=1}^{m-1} \frac{(-1)^{k-1}}{k} \left(\frac{x-x_0}{x_0-y} \right)^k \\ &= \sum_{k=0}^{m-1} g_k(x) h_k(y). \end{aligned} \quad (3.4)$$

Odhadem chyby aproximace se zabývá následující lemma z [4].

Lemma 4. *Nechť $x \in \tau$, $y \in \sigma$ a $\hat{g}(x,y)$ je definovaná jako v (3.4). Pak platí*

$$|g(x,y) - \hat{g}(x,y)| \leq \frac{|x_0-a|}{|c-b|} \left(1 + \frac{|c-b|}{|x_0-a|} \right)^{-k} \quad (3.5)$$

Můžeme si všimnout, že pokud se bude vzdálenost mezi b a c zmenšovat, pravá strana nerovnosti (3.5) poroste do nekonečna. Z toho důvodu je vhodné podmínku $\tau \cap \sigma = \emptyset$ nahradit silnější podmínkou přípustnosti

$$\text{diam}(\tau) \leq \text{dist}(\tau, \sigma), \quad (3.6)$$

neboli v tom případě podmínkou $|b-a| \leq |c-b|$, jak je vidět z obrázku 3.1.

Pokud dáme dohromady nerovnost (3.5) z předchozího lemmatu s podmínkou přípustnosti a definicí bodu $x_0 = \frac{a+b}{2}$, dostaneme nový odhad chyby

$$|g(x,y) - \hat{g}(x,y)| \leq \frac{1}{2} \left(1 + \frac{2}{1} \right)^{-k} \leq \frac{1}{2} 3^{-k} \quad (3.7)$$

Podmínka přípustnosti tedy zajišťuje odhad pro chybu aproximace, který závisí pouze na k řádu Taylorova polynomu.

3.1.3 Aproximace bloků matice

Vraťme se nyní zpět ke vztahu (3.3), speciálně k rovnosti

$$G_{ij} = \int_0^1 \int_0^1 \varphi_i(x)g(x,y)\varphi_j(y)dydx. \quad (3.8)$$

Označme $I = \{0, \dots, n-1\}$ množinu indexů báзовých funkcí φ_i a zafixujme dvě libovolné podmnožiny t a s množiny I . Budeme zkoumat, blok matice $G|_{t \times s}$. Dále označme

$$\tau := \bigcup_{i \in t} \text{supp} \varphi_i, \quad \sigma := \bigcup_{j \in s} \text{supp} \varphi_j. \quad (3.9)$$

Pokud τ a σ splňují podmínku přípustnosti (3.6), můžeme $g(x,y)$ aproximovat pomocí $\hat{g}(x,y)$ díky nerovnosti (3.7). Po dosazení do (3.8) získáváme

$$\begin{aligned} \hat{G}_{ij} &= \int_0^1 \int_0^1 \varphi_i(x) \sum_{k=0}^{m-1} g_k(x)h_k(y)\varphi_j(y)dydx \\ &= \sum_{k=0}^{m-1} \left(\int_0^1 \varphi_i(x)g_k(x)dx \right) \left(\int_0^1 \varphi_j(y)h_k(y)dy \right) \end{aligned}$$

Nahrazení matice $G|_{t \times s}$ maticí $\hat{G}|_{t \times s}$ přináší dvě výhody. První z nich je, že integrál byl nahrazen součinem dvou výrazně jednodušších členů. Druhá výhoda spočívá v možnosti rozepsat $\hat{G}|_{t \times s}$ jako součin dvou matic hodnosti m , což můžeme napsat jako

$$\hat{G}|_{t \times s} = AB^T, \quad A \in \mathbb{R}^{t \times \{0, \dots, k-1\}}, \quad B \in \mathbb{R}^{s \times \{0, \dots, k-1\}},$$

kde A a B jsou definovány pomocí

$$A_{ik} = \int_0^1 \varphi_i(x)g_k(x)dx, \quad B_{jk} = \int_0^1 \varphi_j(x)h_k(y)dy.$$

Tento postup je motivací pro následující definici.

Definice 3. *Blok matice $t \times s \subset I \times I$ nazveme přípustný, pokud τ a σ z (3.9) splňují podmínku přípustnosti z (3.6).*

3.1.4 Dělení matice

Zbývá vyřešit pouze otázku, jak matici vhodně rozdělit do bloků. K tomuto účelu se používají \mathcal{H} -stromy.

Zavedme nejprve značení pro zjednodušení zápisu. Pro strom T označme $S(t)$ množinu všech synů $t \in T$ a $L(T)$ množinu všech listů T . Jinými slovy

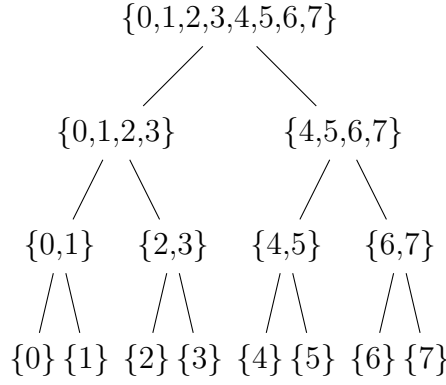
$$L(t) = \{t \in T : S(t) = \emptyset\}.$$

Následující definice \mathcal{H} -stromu je z [9].

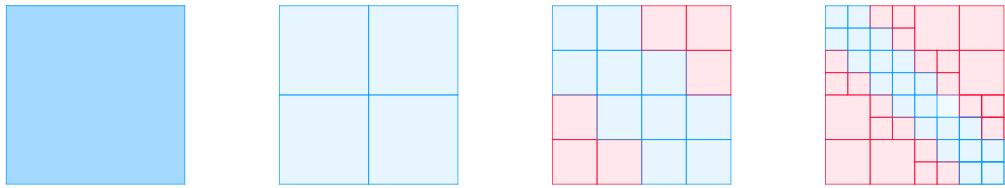
Definice 4. *Nechť J je množina indexů. Strom T nazveme \mathcal{H} -strom (založený na J), pokud platí:*

- (i) $J \in T$.
- (ii) $\#S(t) \neq 1, \quad \forall t \in T$.
- (iii) *Pokud $t \notin L(T)$, $S(t)$ obsahuje disjunktní podmnožiny J a t je sjednocením svých synů, neboli*

$$t = \dot{\bigcup}_{s \in S(t)} s.$$



Obrázek 3.2: \mathcal{H} -strom T_I pro $I = \{0,1,2,3,4,5,6,7\}$



Obrázek 3.3: Dělení $I \times I$ pomocí stromu T_I

V našem případě bude \mathcal{H} -strom obsahovat podmnožiny $I \times I$. Ty pak vytvoří dělení matice na bloky.

Předpokládejme nyní pro jednoduchost navíc, že $n = 2^p$ pro nějaké přirozené číslo p . Nejprve si definujeme pomocný \mathcal{H} -strom T_I obsahující dělení množiny I .

Množina $I_1^{(0)} = \{0, \dots, n-1\}$ je kořen T_I . Její syny jsou $I_1^{(1)} = \{0, \dots, \frac{n}{2}-1\}$ a $I_2^{(1)} = \{\frac{n}{2}, \dots, n-1\}$. Každá další množina $I_i^{(j)}$ obsahující dva nebo více prvků má právě dva syny $I_{2i-1}^{(j+1)}$ a $I_{2i}^{(j+1)}$. Každý z nich obsahuje právě polovinu prvků rodiče. Množiny obsahující pouze jeden prvek jsou listy. Strukturu stromu T_I ilustruje obrázek 3.2.

Pomocí tohoto stromu nyní zkonstruujeme \mathcal{H} -strom $T_{I \times I}$ množiny $I \times I$. Kořen stromu $T_{I \times I}$ definujeme jako $I \times I$. Pak pro každý nepřipustný list $t \times s$ stromu $T_{I \times I}$ definujeme

$$S(t \times s) := \{t' \times s' \mid t' \in S(t), s' \in S(s)\}$$

To budeme opakovat, dokud nebudou všechny bloky přípustné, nebo dokud nenarazíme na listy stromu T_I . Tyto listy nemusí být sice přípustné, zato budou velice malé. Postupné dělení množiny $I \times I$ ilustruje obrázek 3.3. Přípustné bloky jsou vybarveny červeně a nepřipustné modře.

Výsledné bloky dělení matice se vyskytují v listech stromu $T_{I \times I}$. Je tedy přirozené definovat $P := L(T)$, P nazveme dělení $I \times I$.

3.1.5 Hierarchická matice

Po rozdělení matice zbývá už jen vyhodnotit jednotlivé listy T . Zde je třeba rozlišovat listy přípustné a nepřipustné.

Nepřístupné listy

Prvky v nepřístupném bloku $t \times s \in P$ matice G nesplňují nerovnost (3.7) a nemůžeme tedy použít aproximaci pomocí $\hat{g}(x,y)$ z (3.4). Je nutné spočítat \hat{G}_{ij} numericky z definice

$$\hat{G}_{ij} = \int_{\frac{i}{n}}^{\frac{i+1}{n}} \int_{\frac{j}{n}}^{\frac{j+1}{n}} \log |x - y| dy dx, \quad i \in t, j \in s.$$

To ale není takový problém, protože nepřístupné bloky jsou z pravidla velmi malé.

Přístupné listy

U přístupných bloků $t \times s \in P$ máme k dispozici odhad chyby (3.7) pro aproximaci $g(x,y)$ pomocí $\hat{g}(x,y)$, která plyne z podmínky přístupnosti. Použitím rovností odvozených v sekci 3.1.3 dostáváme vztahy

$$\begin{aligned} \hat{G}|_{t \times s} &= AB^T, \\ A_{ik} &= \int_{\frac{i}{n}}^{\frac{i+1}{n}} (x - x_0)^k dx, \\ B_{jk} &= \begin{cases} \int_{\frac{j}{n}}^{\frac{j+1}{n}} \log |x_0 - y| dy, & k = 0 \\ (-1)^{k-1} k^{-1} \int_{\frac{j}{n}}^{\frac{j+1}{n}} (x_0 - y)^{-k} dy, & k > 0. \end{cases} \end{aligned} \quad (3.10)$$

Připomeňme ještě, že x_0 je střed intervalu $\tau = \bigcup_{i \in t} \text{supp} \varphi_i$.

Pomocí (3.10) se vypočítají matice $A \in \mathbb{R}^{t \times \{0, \dots, k-1\}}$ a $B \in \mathbb{R}^{s \times \{0, \dots, k-1\}}$. Blok $t \times s$ je pak v paměti reprezentován pomocí jejich součinu. To vyžaduje uložení $(|t| + |s|)k$ čísel místo obvyklých $|t||s|$ čísel.

Dohromady je pak matice \hat{G} tvořena bloky, které jsou tvořeny buď součinem dvou úzkých matic, nebo jsou reprezentovány jako klasická hustá matice. Tento způsob uložení matice \hat{G} výrazně urychluje operace s ní a zároveň šetří paměť. Maticím tohoto tvaru se říká \mathcal{H} -matice. Následující definice je z [9].

Definice 5. *Bud' P dělení $I \times I$ a $k \in \mathbb{N}$.*

Množinu \mathcal{H} -matic indukovanou dělením P definujeme jako

$$\mathcal{M}_{\mathcal{H},k}(I \times I, P) := \{M \in \mathbb{R}^{I \times I} : \text{rank}(M|_b) \leq k \quad \forall b \in P\}.$$

3.2 Operace s \mathcal{H} -maticemi

S \mathcal{H} -maticemi je možné provádět operace jako sčítání, násobení nebo invertování efektivněji. Touto problematikou se zabývá například [4].

Množina $\mathcal{M}_{\mathcal{H},k}(I \times I, P)$ není bohužel uzavřená na sčítání ani násobení, proto je třeba definovat seřezávací operátor \mathcal{T}_k , který zajistí, že výsledek operací bude mít správný formát. Ten využívá větu 3 z předchozí kapitoly na jednotlivé bloky dělení.

Definice 6. *Nechť $M \in \mathbb{R}^{m \times n}$, $k \in \mathbb{N}$ a P dělení množiny $I \times I$. Seřezávací operátor \mathcal{T}_k*

$$\mathcal{T}_k : \mathbb{R}^{m \times n} \rightarrow \mathcal{M}_{\mathcal{H},k}(I \times I, P), \quad M \mapsto \tilde{M}$$

definujeme blokově pro každé $t \times s \in P$ předpisem

$$\tilde{M}|_{t \times s} := \begin{cases} M|_{t \times s}^{(k)}, & \text{pokud je } t \times s \text{ přípustná} \\ M|_{t \times s}, & \text{jinak,} \end{cases}$$

kde $M|_{t \times s}^{(k)}$ je aproximace hodnoti k ve smyslu věty 3.

Bloky přípustné bloky hierarchické matice jsou reprezentovány ve tvaru AB^T . Díky tomu je možné počítat jejich singulární rozklad efektivněji ve třech krocích.

1. Spočítat ekonomické QR rozklady matic A, B : $A = Q_A R_A, B = Q_B R_B$.
2. Spočítat ekonomický rozklad matice $R_A R_B^T = U' \Sigma V'$.
3. Spočítat $U := Q_A U', V := Q_B V'$.

Tím získáme singulární rozklad $AB^T = U \Sigma V$ odpovídající seřezávacímu operátoru \mathcal{T}_k .

3.2.1 Sčítání

Sčítání hierarchických matic se provádí po složkách. Většina jejích bloků je však definována pomocí součinu matic nízké hodnoti. Je proto třeba vyřešit, jak tyto bloky sčítat.

Sčítání matic AB^T a CD^T hodnoti k je možné zapsat

$$AB^T + CD^T = \begin{pmatrix} A & C \end{pmatrix} \begin{pmatrix} B & D \end{pmatrix}^T.$$

Z toho plyne, že součet matic hodnoti k je vždy hodnoti maximálně $2k$. Protože je potřeba zachovat formát, aplikujeme na výsledný součet matic operátor \mathcal{T}_k , čímž získáme aproximaci součtu s požadovaným formátem.

3.2.2 Násobení a inverze

Násobení a invertování hierarchických matic jsou poněkud složitější operace. Obě tyto operace postupují rekurzí a sčítání diskutované výše společně s dalšími technikami.

Náročná implementace reprezentace hierarchických matic a operací s nimi je jedním z hlavních důvodů, proč jsme se je rozhodli nepoužít v experimentech. Naopak důvodem, proč jsme zde tuto metodu popsali, je, že představuje nový a rychle se rozvíjející přístup, který propojuje konkrétní aplikace s numerickou lineární algebrou.

4. Algebraické přístupy

Další z možností aproximace matice velkých rozměrů pomocí matic nízké hodnoty je použít tzv. pseudo-skeletní aproximaci. Ta je založena na vybírání určitého počtu řádků a sloupců matice, které ji dobře reprezentují. Zde budeme čerpat převážně z [8] a [13].

4.1 Motivace

Uvažme matici $A \in \mathbb{R}^{m \times n}$ hodnosti r . Pak A obsahuje regulární podmatici $\hat{A} \in \mathbb{R}^{r \times r}$. Označme $R \in \mathbb{R}^{r \times n}$ a $C \in \mathbb{R}^{m \times r}$ podmatice řádků a sloupců matice A , které obsahují podmatici \hat{A} . Pak platí

$$A = C\hat{A}^{-1}R.$$

Tento rozklad matice A se nazývá skeletní.

Předpokládejme nyní, že $\text{rank}(A) \approx r$ ve smyslu $\text{rank}(A - E) = r$, kde $\|E\| \approx 0$. V následujícím se budeme zabývat hledáním $R \in \mathbb{R}^{r \times n}$, $C \in \mathbb{R}^{m \times r}$ a $G \in \mathbb{R}^{r \times r}$, aby $A \approx B = CGR$. Taková matice B se nazývá pseudo-skeletní komponenta matice A . V tomto případě může být matice G zvolena jinak než \hat{A}^{-1} . Za předpokladů $\text{rank}(A) \approx r$ totiž \hat{A} může být velmi špatně podmíněná, nebo dokonce singulární.

4.2 Odhady pro pseudo-skeletní aproximaci

Zabývejme se nyní otázkou, jak dobré aproximace je možné dosáhnout za předpokladů $\text{rank}(A - E) \approx r$ a $\|E\|_2 < \varepsilon$. Pokud ε je dostatečně malé a \hat{A} není singulární, je možné dokázat, že

$$\|A - C\hat{A}^{-1}R\|_2 = O\left(\|A\|_2^2 \|\hat{A}^{-1}\|_2^2 \varepsilon\right).$$

Pokud bychom tedy chtěli za matici G zvolit \hat{A} , je třeba, aby $\|\hat{A}^{-1}\|$ bylo dostatečně malé. V takovém případě může správná volba řádků a sloupců výrazně vylepšit aproximaci. Ukazuje se, že $\|\hat{A}^{-1}\|$ je dostatečně malé, pokud \hat{A} je podmatice s maximálním determinantem v absolutní hodnotě. Hledáním takových podmatic se budeme zabývat v další sekci.

Věta 5. *Nechť $A, E \in \mathbb{R}^{m \times n}$, $\text{rank}(A - E) \leq r$ a $\|E\|_2 \leq \varepsilon$ pro nějaké $\varepsilon > 0$. Pak existuje r sloupců a r řádků, které určují pseudo-skeletní komponentu CGR takovou, že platí*

$$\|A - CGR\|_2 \leq \varepsilon \left(1 + \left(\sqrt{t(r, n)} + \sqrt{t(r, m)}\right)^2\right),$$

kde

$$t(r, n) = \frac{1}{\min_U \max_{P \in \mathcal{M}(U)} \sigma_{\min}(P)},$$

$U^T U = I, U \in \mathbb{R}^{n \times r}, r \leq n$. $\mathcal{M}(U)$ značí množinu všech $r \times r$ podmatic U a $\sigma_{\min}(P)$ značí nejmenší singulární číslo matice P .

Důkaz. Uvažujme singulární rozklad

$$A - E = U\Sigma V, \quad (4.1)$$

kde $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_r)$ a $\sigma_1 \leq \dots \leq \sigma_r \leq 0$. Označme $\hat{U}, \hat{V} \in \mathbb{R}^{r \times r}$ podmatice U, V s maximálním objemem. Pak platí

$$\begin{aligned} \|\hat{U}^{-1}\|_2 &\leq t(r, m), \\ \|\hat{V}^{-1}\|_2 &\leq t(r, n). \end{aligned}$$

Matice \hat{U}, \hat{V} označují výběr řádků a sloupců matice A do pseudo-skeletního rozkladu. Označme C a E_C podmatice r sloupců matic A a E . Stejným způsobem označme R a E_R podmatice r řádků. Symboly \hat{A} a \hat{E} budeme značit $r \times r$ podmatice A a E , které odpovídají stejným sloupečkům a řádkům jako v C a R . Buď $G \in \mathbb{R}^{r \times r}$ zatím libovolná matice. Pak CGR je pseudo-skeletní aproximace matice A . Tu lze napsat pomocí následujících rovností.

$$\begin{aligned} CGR &= (U\Sigma\hat{V} + E_C) G (\hat{U}\Sigma V + E_R) \\ &= U\Sigma\hat{V}G\hat{U}\Sigma V + F \\ &= U\hat{U}^{-1}\Phi G\Phi\hat{V}^{-1}V + F, \end{aligned} \quad (4.2)$$

$$\begin{aligned} F &= (U\Sigma\hat{V} + E_C) G E_R + E_C G (\hat{U}\Sigma V + E_R) - E_C G E_R \\ &= U\hat{U}^{-1} (\hat{U}\Sigma\hat{V}G) E_R + E_C (G\hat{U}\Sigma\hat{V}) \hat{V}^{-1}V + E_C G E_R \\ &= U\hat{U}^{-1} (\Phi G) E_R + E_C (G\Phi) \hat{V}^{-1}V + E_C G E_R, \end{aligned} \quad (4.3)$$

kde

$$\Phi = \hat{U}\Sigma\hat{V} = \hat{A} - \hat{E}.$$

Přepíšeme-li i (4.1) pomocí Φ , získáme

$$A - E = U\hat{U}^{-1}\Phi\hat{V}^{-1}V. \quad (4.4)$$

Uvažme nyní singulární rozklad matice Φ

$$\Phi = \tilde{U}\tilde{\Sigma}\tilde{V}, \quad \tilde{\Sigma} = \text{diag}(\tilde{\sigma}_1, \dots, \tilde{\sigma}_r)$$

a číslo $\tau > 0$ zatím libovolné. Označíme-li

$$\begin{aligned} \tilde{\Sigma}_r &= \text{diag}(\tilde{\sigma}_{\tau i}), \quad \tilde{\sigma}_{\tau i} = \begin{cases} \sigma_i & \text{pokud } \sigma_i \geq \tau, \\ 0 & \text{jinak} \end{cases} \\ \tilde{\Sigma}_r^+ &= \text{diag}(\tilde{\sigma}_{\tau i}^+), \quad \tilde{\sigma}_{\tau i}^+ = \begin{cases} \sigma_i^{-1} & \text{pokud } \sigma_i \geq \tau, \\ 0 & \text{jinak} \end{cases} \\ \Phi_r &= \tilde{U}\tilde{\Sigma}_r\tilde{V}, \\ \Phi_r^+ &= \tilde{V}^T\tilde{\Sigma}_r^+\tilde{U}^T, \end{aligned} \quad (4.5)$$

platí

$$\Phi\Phi_r^+\Phi = \Phi_r. \quad (4.6)$$

Zvolme

$$G = \Phi_\tau^+,$$

pak z (4.3) plyne

$$\begin{aligned} \|F\|_2 &\leq \|\hat{U}^{-1}\|_2 \|\Phi \Phi_\tau^+\|_2 \|E_R\|_2 + \|E_C\|_2 \|\Phi_\tau^+ \Phi\|_2 \|\hat{V}^{-1}\|_2 + \|E_C\|_2 \|\Phi_\tau^+\|_2 \|E_R\|_2 \\ &\leq \varepsilon \left(\|\hat{U}^{-1}\|_2 \|\Phi \Phi_\tau^+\|_2 + \|\Phi_\tau^+ \Phi\|_2 \|\hat{V}^{-1}\|_2 + \varepsilon \|\Phi_\tau^+\|_2 \right). \end{aligned}$$

Využitím nerovností

$$\|\Phi \Phi_\tau^+\|_2 \leq 1, \quad \|\Phi_\tau^+ \Phi\|_2 \leq 1, \quad \|\Phi_\tau^+\|_2 \leq \frac{1}{\tau},$$

které plynou ze značení (4.5), dostáváme odhad

$$\|F\|_2 \leq \varepsilon \left(\|\hat{U}^{-1}\|_2 + \|\hat{V}^{-1}\|_2 + \frac{\varepsilon}{\tau} \right).$$

Z rovností (4.2), (4.4) a (4.6) pak plyne

$$\begin{aligned} \|A - CGR\|_2 &= \|E + U\hat{U}^{-1}\Phi\hat{V}^{-1}V - U\hat{U}^{-1}\Phi\Phi_\tau^+\Phi\hat{V}^{-1}V - F\|_2 \\ &\leq \|E\|_2 + \|U\hat{U}^{-1}(\Phi - \Phi\Phi_\tau^+\Phi)\hat{V}^{-1}V\|_2 + \|F\|_2 \\ &\leq \varepsilon + \|\hat{U}^{-1}\|_2 \|\Phi - \Phi_\tau\|_2 \|\hat{V}^{-1}\|_2 + \|F\|_2 \\ &\leq \varepsilon + \tau \|\hat{U}^{-1}\|_2 \|\hat{V}^{-1}\|_2 + \frac{\varepsilon^2}{\tau} + \varepsilon \|\hat{U}^{-1}\|_2 + \varepsilon \|\hat{V}^{-1}\|_2. \end{aligned} \tag{4.7}$$

K dokončení důkazu zbývá zvolit $\tau = \varepsilon / \sqrt{\|\hat{U}^{-1}\|_2 \|\hat{V}^{-1}\|_2}$. Po dosazení do (4.7) máme

$$\begin{aligned} \|A - CGR\|_2 &\leq \varepsilon \left(1 + 2\sqrt{\|\hat{U}^{-1}\|_2 \|\hat{V}^{-1}\|_2} + \|\hat{U}^{-1}\|_2 + \|\hat{V}^{-1}\|_2 \right) \\ &\leq \varepsilon \left(1 + \left(\sqrt{\|\hat{U}^{-1}\|_2} + \sqrt{\|\hat{V}^{-1}\|_2} \right)^2 \right) \\ &\leq \varepsilon \left(1 + \left(\sqrt{t(r,n)} + \sqrt{t(r,m)} \right)^2 \right), \end{aligned}$$

což jsme chtěli dokázat. □

Věta 5 bohužel používá znalost matice chyby E , není tedy prakticky použitelná. V následující větě uvažujeme matici G , kterou je možné vybrat bez znalosti matice chyby. Zde ji uvedeme bez důkazu, který je možné najít v [8].

Věta 6. *Za předpokladů věty 5 existuje G vybrané pouze na základě \hat{A} , které splňuje*

$$\|A - CGR\|_2 \leq \varepsilon \sqrt{(1 + t^2(r,p))} \left(1 + \left(\sqrt{t(r,n)} + \sqrt{t(r,m)} \right)^2 \right),$$

kde $p = \min\{m, n\}$.

4.3 Algoritmus Maxvol

Pro nalezení dobré pseudo-skeletní aproximace je potřeba algoritmus pro hledání podmatic s maximální absolutní hodnotou determinantu. O tomto tématu pojednává článek [7].

Zde budeme uvažovat matici $A \in \mathbb{R}^{m \times r}$ a hledat mezi podmaticemi $r \times r$. Vybíráme tedy pouze řádky matice A a nikoli její sloupce.

Definice 7. Čtvercovou podmaticí A_{\max} matice A velikosti $r \times r$ nazveme matici s maximálním objemem, pokud platí

$$|\det(A_{\max})| = \max_{\hat{A}} |\det(\hat{A})|,$$

kde \hat{A} jsou $r \times r$ podmatice matice A .

Definice 8. Čtvercovou podmaticí A_D matice A velikosti $r \times r$ nazveme dominantní, pokud všechny prvky matice AA_D^{-1} mají absolutní hodnotu menší nebo rovnou 1.

Ukazuje se, že podmatice s maximálním objemem je vždy dominantní. Zároveň také pro každou dominantní podmatici platí

$$|\det(A_D)| \geq \frac{|\det(A_{\max})|}{r^{r/2}}.$$

Tento odhad je sice pesimistický, ukazuje ale, že místo podmatice A_{\max} můžeme hledat matici dominantní. K tomu slouží algoritmus Maxvol.

Algoritmus 3: Algoritmus Maxvol

Input: $A \in \mathbb{R}^{n \times r}$

Zvolit $\hat{A} \in \mathbb{R}^{r \times r}$ podmatici A , přeuspořádat řádky A tak, aby \hat{A} byla v prvních r řádcích

for $k=1,2,\dots$ **do**

$$B = A\hat{A}^{-1}$$

$$\hat{b}_{ij} = \arg \max |b_{ij}|$$

if $|\hat{b}_{ij}| > 1$ **then**

Vyměnit i -tý a j -tý řádek v A

end

if $|\hat{b}_{ij}| = 1$ **then**

return \hat{A}

end

end

Algoritmus v každé iteraci provádí některé výpočetně náročné operace, jako například hledání inverze matice či součin dvou matic. Vyšší efektivity je možné dosáhnout pomocí Sherman-Morrisonova vzorce.

Věta 7. Necht $A \in \mathbb{R}^{n \times n}$ je invertovatelná matice a $u, v \in \mathbb{R}^n$ jsou sloupcové vektory. Pak je matice $A + uv^T$ invertovatelná právě tehdy, když $1 + v^T A^{-1}u \neq 0$. Pokud je matice $A + uv^T$ invertovatelná, pak platí

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^T A^{-1}}{1 + v^T A^{-1}u}.$$

V každé iteraci algoritmu se vymění dva řádky matice A . Označíme-li symbolem a_i i -tý řádek matice A , můžeme tuto operaci napsat ve tvaru

$$A := A + e_j(a_i. - a_j.) + e_i(a_j. - a_i.) = A + (e_j - e_i)(a_i. - a_j.) = A + uv^T. \quad (4.8)$$

Stejným způsobem lze zapsat změna matice \hat{A} jako

$$\hat{A} := \hat{A} + e_j(a_i. - a_j.) = \hat{A} + \hat{u}v^T.$$

Pomocí věty 7 můžeme tedy spočítat \hat{A}^{-1} z předchozí iterace předpisem

$$\hat{A}^{-1} := \hat{A}^{-1} + \frac{\hat{A}^{-1}\hat{u}v^T\hat{A}^{-1}}{1 + v^T\hat{A}^{-1}\hat{u}} \quad (4.9)$$

Použitím vzorců (4.8), (4.9) a několika jednoduchých vztahů je pak možné vyjádřit B ve tvaru

$$B = A\hat{A}^{-1} := B - (b_{.j} - e_j + e_i)(b_{i.} - e_j^T) / b_{ij}, \quad (4.10)$$

kde symbol $b_{.j}$ značí j -tý sloupec matice B . Více podrobností k odvození vzorce (4.10) je možné nalézt v článku [7].

Dalším způsobem, jak zkrátit dobu výpočtu, je změna zastavovacího kritéria. Pokud místo $|\hat{b}_{ij}| = 1$ použijeme podmínku $|\hat{b}_{ij}| \leq 1 + \delta$, může to výrazně snížit počet iterací.

Tento algoritmus je možné využít například pro hledání prvku s maximální absolutní hodnotou v matici nízké hodnoty UV^T . To je potřeba řešit například v maximalizaci dvourozměrné separabilní funkce na pevné mřížce. My tento algoritmus budeme používat v experimentech pro počítání předpokládání.

5. Pravděpodobnostní přístupy

Velmi zajímavou a novou skupinu metod aproximací maticemi nízkých hodnotí tvoří pravděpodobnostní přístupy. Metody na nich založené jsou vhodné například k hledání oboru hodnot matice, ale mají i další široká uplatnění. Mnoha z nich se věnuje například [10]. V této kapitole se budeme zabývat hlavně myšlenkou, která tyto přístupy spojuje. Tu pak budeme ilustrovat na řešiči Blendepik pro řešení problému nejmenších čtverců z [1].

5.1 Hlavní myšlenky

Počítání aproximace nízké hodnoty matice $A \in \mathbb{R}^{m \times n}$ pomocí pravděpodobnosti je možné rozdělit na dvě fáze. V první z nich je potřeba najít podprostor nízké dimenze, který zachycuje akci matice A . To znamená, že je třeba spočítat bázi $\mathcal{R}(A)$. Hledáme tedy matici Q takovou, aby měla ortogonální sloupce a

$$A \approx QQ^*A.$$

Ve druhé fázi výpočtu je matice Q využívána ke spočtení požadovaného rozkladu. Hlavní otázkou tedy je, jak najít matici Q .

Předpokládejme nyní, že matice A je hodnoty k a my se snažíme najít bázi $\mathcal{R}(A)$. Uvažujme náhodný vektor ω a spočítejme $y = A\omega$. Vektor y pak bude náhodný vzorek z $\mathcal{R}(A)$. Budeme-li tento proces opakovat k -krát, získáme množinu $\{y^{(i)}, i = 1, 2, \dots, k\}$ vektorů z $\mathcal{R}(A)$. Pravděpodobnost, že nějaký z vektorů $\omega^{(i)}$ je lineární kombinací předchozích, je nulová. Z toho plyne, že $\{y^{(i)}, i = 1, 2, \dots, k\}$ je báze $\mathcal{R}(A)$ skoro jistě.

Obecně musíme počítat s tím, že matice A je tvaru $B + E$. Matice B je hodnoty k , která nás zajímá, a matice E je určitá chyba. Z rovnosti

$$y^{(i)} = B\omega^{(i)} + E\omega^{(i)}$$

je patrné, že člen $E\omega^{(i)}$ vychyluje vektory z podprostoru $\mathcal{R}(B)$. To by mohlo v výsledku znamenat, že báze, kterou nalezneme, nebude úplná. Proto je potřeba vzít o $p \in \mathbb{N}$ vzorků víc. Dle [10] stačí zvolit $p = 5$ nebo $p = 10$. Tento postup můžeme zapsat pomocí algoritmu, který funguje ve třech krocích.

1. Získat náhodnou $n \times (k + p)$ matici Ω ,
2. Spočítat matici $Y = A\Omega$,
3. Zkonstruovat matici Q , jejíž sloupce tvoří ortonormální bázi $\mathcal{R}(Y)$.

Tímto algoritmem jsme vyřešili první fázi celého procesu.

Ve druhé fázi je třeba pomocí matice Q spočítat požadovaný rozklad matice A . Tento postup je možné použít například výpočet singulárního rozkladu. Místo A budeme počítat s maticí

$$B = Q^*A. \tag{5.1}$$

Ta je malá a můžeme tedy spočítat její singulární rozklad s malými výpočetními náklady. Označíme-li $B = \hat{U}\Sigma V^*$, pak můžeme spočítat singulární rozklad A z rovnosti (5.1)

$$\begin{aligned} Q^*A &= \hat{U}\Sigma V^*, \\ A &= Q\hat{U}\Sigma V^*. \end{aligned}$$

Po označení $U = Q\hat{U}$ je singulární rozklad matice A hotov.

5.2 Blendenpik

Na pravděpodobnostních metodách je založen například řešič problému nejmenších čtverců Blendenpik. Jeho autoři v [1] právě tuto metodu použili, aby vylepšili výpočetní rychlost klasického řešiče problému nejmenších čtverců pro výrazně přeuročené úlohy.

5.2.1 Problém nejmenších čtverců

Uvažujme matici $A \in \mathbb{R}^{m \times n}$ a vektor $b \in \mathbb{R}^m$, která představují data pro soustavu lineárních rovnic

$$Ax = b.$$

Tato rovnice nemusí mít řešení, proto se snažíme najít alespoň co nejlepší přibližné řešení. To lze napsat například takto

$$\hat{x} = \arg \min_x \|Ax - b\|_2.$$

Předpokládejme nyní, že $m \gg n$, vyberme malý vzorek řádků matice a označme jej \mathcal{R} . Pokud spočítáme QR rozklad matice \mathcal{R} , můžeme jeho faktor R použít jako předpokládání pro iterační řešič LSQR. Pokud byla matice \mathcal{R} vybrána správně, s velkou pravděpodobností se zmenší počet iterací potřebných k nalezení řešení.

To, jak je matice R vhodná, závisí hlavně na matici A a také počtu řádků vzorku. Vlastnost, která dobře vystihuje, zda tento přístup bude dobře fungovat na matici A , se nazývá koherence.

5.2.2 Koherence

Představme si situaci, že matice a má v j -tém sloupci samé nuly, kromě i -té pozice. Pak by každý výběr neobsahující i -tý řádek matice byl singulární. To by ve výsledku způsobilo selhání LSQR. Počet vybraných řádků by se v tomto případě musel blížit k číslu m , aby k selhání nedošlo. Pak by ovšem celý postup ztrácel smysl, protože by nebyl efektivní.

Veličina, která ukazuje, do jaké míry je řešení závislé na určitých řádcích, se nazývá koherence.

Definice 9. *Bud' $A \in \mathbb{R}^{m \times n}$ s plnou sloupcovou hodností a $U \in \mathbb{R}^{m \times n}$ matice, jejíž sloupce tvoří ortonormální bázi sloupců A . Koherence matice A definujeme předpisem*

$$\mu(A) = \max_{1 \leq i \leq m} \|U_{i,*}\|_2^2.$$

Koherence matice je vždy mezi čísly n/m a 1 s tím, že čím je větší, tím je potřeba vybrat více řádků pro dobré předpokmínění. Následující věta dává do vztahu koherenci matice A , počet vybraných řádků a kvalitu předpokmínění.

Věta 8. *Bud' $A \in \mathbb{R}^{m \times n}$ matice s plnou sloupcovou hodností a \mathcal{S} operátor, který z matice A náhodně vybere $r \geq n$ řádků. Definujme $\tau = C\sqrt{m\mu(A)\log(r)}/r$, kde C je určitá konstanta. Předpokládejme, že $\delta^{-1}\tau \leq 1$. Pak s pravděpodobností alespoň $1 - \delta$ má matice $\mathcal{S}A$ plnou sloupcovou hodnost a pokud $\mathcal{S}A = QR$ je ekonomický QR rozklad $\mathcal{S}A$, platí*

$$\kappa(AR^{-1}) \leq \frac{1 + \delta^{-1}\tau}{1 - \delta^{-1}\tau}$$

Z věty 8 plyne několik důležitých pozorování. Bude-li počet vybraných řádků r malý, pak pravděpodobnost získání velkého čísla podmíněnosti δ bude vysoká. Potřebný počet vybraných řádků závisí na koherenci matice A . Pokud bude mít matice A koherenci rovnou 1, pak je pro potřeby předpokladů věty nutné r zvolit jako $O(m \log(m))$. Naproti tomu pro nejlepší případ $\mu(A) = n/m$ bude potřeba r pouze $\Theta(n \log(n))$. Z předpokladu $m \gg n$ je tedy zřejmé, že koherence je důležitým faktorem při vybírání počtu řádků.

5.2.3 Algoritmus

Jedním z problémů je, že $\mu(A)$ je složité určit předem. Algoritmus Blendepik tento problém řeší mícháním řádků. Tento přístup využívá pozorování, že unitární transformace mění koherenci, ale ne číslo podmíněnosti. Vhodné transformace pak s vysokou pravděpodobností zajišťují nízkou koherenci nezávisle na koherenci původní matice A .

Celý algoritmus lze pak zapsat pomocí tří kroků.

1. Míchání řádků matice A pomocí vhodné unitární matice F : $B = FA$,
2. Náhodný výběr r řádků matice B : $\hat{B} = \mathcal{S}B$,
QR rozklad matice \hat{B} : $\hat{B} = QR$,
3. Iterační řešení úlohy $Ax \approx b$ pomocí metody LSQR s předpokmíněním R .

Pravděpodobnostní algoritmy byly do výpočtové matematiky zavedeny poměrně nedávno, ačkoli jejich teoretické pozadí prošlo dlouhým vývojem. Jejich dosud nedostatečně vyzkoušená robustnost je důvodem, proč se experimentální část této práce zaměřuje na jeden z uvedených algebraických přístupů. Experimentální výsledky, kterých dosahuje algoritmus Blendepik v [1], jsou velmi slibné. Předpoklad $m \gg n$ je ale bohužel velmi omezující.

6. Experimenty

V této kapitole budeme řešit problém nejmenších čtverců pomocí pseudo-skeletního rozkladu z kapitoly 4. Tento přístup jsme si vybrali mezi diskutovanými technikami pro svoji inovativnost a realizovatelnost. Podmatici vypočítanou pomocí algoritmu Maxvol budeme používat k předpokládání metody sdružených gradientů pro řešení soustavy normálních rovnic.

6.1 Problém nejmenších čtverců

Mějme matici $A \in \mathbb{R}^{m \times n}$ s vektorem pravé strany $b \in \mathbb{R}^m$ a řešme úlohu

$$Ax \approx b. \quad (6.1)$$

Nemá-li soustava $Ax = b$ přesné řešení, budeme hledat aproximaci s co nejmenší odchylkou. V našem případě budeme předpokládat, že vektor b obsahuje chyby. Opravíme tedy pravou stranu o vektor f a budeme řešit rovnici

$$Ax = b + f. \quad (6.2)$$

Naším cílem je najít řešení, které je co nejbližší řešení původního problému (6.1). Proto se budeme snažit minimalizovat f . Z rovnosti (6.2) pak řešení splňuje

$$\min_x \|Ax - b\|_2. \quad (6.3)$$

Takové řešení se nazývá řešení ve smyslu nejmenších čtverců. Poznamenejme ještě, že pokud bychom uvažovali chyby nejen v b , ale i v A , pak by se jednalo o problém úplných nejmenších čtverců, jehož řešení je mnohem složitější.

6.2 Soustava normálních rovnic

Zabývejme se nyní otázkou, jak najít řešení ve smyslu nejmenších čtverců. Podmínku (6.3) je možné přeformulovat pomocí následující věty z [3].

Věta 9. Označme \mathcal{S} množinu všech řešení splňující (6.3). Pak $x \in \mathcal{S}$ právě tehdy, když platí

$$A^T(b - Ax) = 0. \quad (6.4)$$

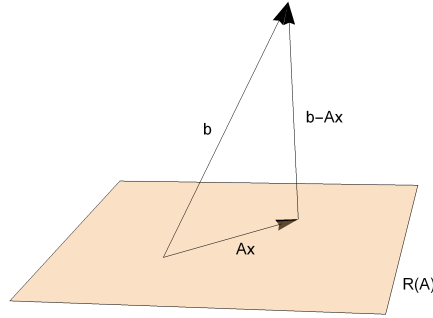
Důkaz. \Rightarrow : Předpokládejme, že \hat{x} splňuje (6.4) a označme $\hat{r} = b - A\hat{x}$. Pak pro každé $x \in \mathbb{R}^n$ platí

$$r = b - Ax = b - A\hat{x} + A\hat{x} - Ax = \hat{r} + A(\hat{x} - x).$$

Spočteme-li nyní $r^T r = \|r\|_2^2$, s využitím předpokladu $A^T \hat{r} = 0$ dostaneme

$$\begin{aligned} r^T r &= \hat{r}^T \hat{r} + \hat{r}^T A(\hat{x} - x) + (\hat{x} - x)^T A^T \hat{r} + (A(\hat{x} - x))^T A(\hat{x} - x) \\ \|r\|_2^2 &= \|\hat{r}\|_2^2 + \|A(\hat{x} - x)\|_2^2 \end{aligned}$$

Norma $r = b - Ax$ je tedy minimální pro $x = \hat{x}$.



Obrázek 6.1: Problém nejmenších čtverců

⇐: Předpokládejme nyní, že $A^T \hat{r} = z \neq 0$ a zvolme $x = \hat{x} + \varepsilon z$. Pak platí

$$r = b - Ax = b - A(\hat{x} + \varepsilon z) = \hat{r} - \varepsilon Az$$

a odtud využitím $A^T \hat{r} = z$ dostaneme

$$\begin{aligned} r^T r &= \hat{r}^T \hat{r} - \varepsilon \hat{r}^T Az - \varepsilon (Az)^T \hat{r} + \varepsilon^2 (Az)^T Az \\ \|r\|_2^2 &= \|\hat{r}\|_2^2 - 2\varepsilon \|z\|_2^2 + \varepsilon^2 \|Az\|_2^2. \end{aligned} \quad (6.5)$$

Epsilon bylo libovolně malé, z (6.5) tedy plyne $\|r\|_2^2 \leq \|\hat{r}\|_2^2$. Vektor \hat{x} tedy není řešení ve smyslu nejmenších čtverců. \square

Z věty 9 je zřejmé, že vektor $r = Ax - b$ musí být kolmý na $\mathcal{R}(A)$. To ilustruje obrázek 6.1.

Přeuspořádáním členů v (6.4) získáme soustavu normálních rovnic

$$A^T Ax = A^T b. \quad (6.6)$$

Díky větě 9 máme zajištěno, že x je řešením $Ax \approx b$ ve smyslu nejmenších čtverců právě tehdy, když je řešením soustavy normálních rovnic.

Matice soustavy normálních rovnic $A^T A$ je zřejmě symetrická. Pro použití metody sdružených gradientů je však potřeba, aby $A^T A$ byla také pozitivně definitní. Tento problém řeší následující věta.

Věta 10. *Matice $A^T A$ je pozitivně definitní právě tehdy, když sloupce A jsou lineárně nezávislé, neboli $\text{rank}(A) = n$.*

Důkaz. \Rightarrow : Pokud jsou sloupce A lineárně nezávislé, pak platí $Ax \neq 0$ pro všechny $x \neq 0$. Odtud pak plyne $x^T A^T Ax > 0$, což je definice pozitivní definitnosti matice $A^T A$.

\Leftarrow : Pokud jsou sloupce A lineárně závislé, pak existuje vektor $x_0 \neq 0$ takový, že $Ax_0 = 0$. Pak ale platí $x_0^T A^T Ax_0 = 0$, proto $A^T A$ není pozitivně definitní. \square

Věta 10 tedy zaručuje splnění předpokladů pro použití sdružených gradientů, pokud budeme pracovat s maticí plné sloupcové hodnotnosti.

Je třeba dodat, že řešení soustavy normálních rovnic není optimální způsob řešení problému nejmenších čtverců, protože soustava (6.6) má číslo podmíněnosti $\kappa(A^2)$. Pro špatně podmíněnou soustavu to může znamenat výrazný nárůst chyb.

Přesto jsme si tento způsob vybrali, protože není příliš náročný na realizaci, ale hlavně je základním postupem pro kombinaci s iterační metodou. Přináší také výhody pokud $m > n$. Matice soustavy $A^T A$ má pak rozměry $n \times n$ a je tedy menší než původní matice A .

6.3 Předpodmínění pomocí Maxvol

Pro předpodmínění budeme používat algoritmus Maxvol z kapitoly 4, který umí najít dominantní podmatice $A_D \in \mathbb{R}^{n \times n}$ matice $A \in \mathbb{R}^{m \times n}$. Obor hodnot matice A_D pak aproximuje obor hodnot matice A . Matici předpodmínění tedy budeme volit jako $A_D^T A_D$. Předpodmíněná soustava má pak tvar

$$(A_D^T A_D)^{-1} A^T A x = (A_D^T A_D)^{-1} b$$

Matice $A_D^T A_D$ je symetrická a pozitivně definitní, můžeme proto vypočítat její neúplný Choleského rozklad

$$A_D^T A_D \approx L^T L.$$

Tímto způsobem předpodmínění navazujeme na návrh přímé metody, kterou dle [3] v roce 1961 navrhl Läuchli. Náš hlavní záměr je ukázat, že tato transformace je efektivní a může umožnit kombinaci s dalšími způsoby předpodmínění.

6.4 Realizace

Kód pro experimenty jsme psali ve Fortranu a díky tomu jsme mohli použít některé existující spolehlivé rutiny. Zaměřili jsme se na řídké matice, proto matice ukládáme v řídkém formátu pomocí tří polí, viz [12]. Při implementaci jsme kladli důraz na funkčnost více než na efektivitu. Kód je však možné optimalizovat, aby byl použitelný i na matice větších rozměrů, než jsou námi používané testovací matice. To však není naším cílem. Kostra kódu je zachycena v algoritmu 4. Ta je z důvodu přehlednosti psána velmi schématicky pomocí metod diskutovaných níže.

6.4.1 Regularizace

Při hledání dominantní podmatice pomocí algoritmu Maxvol je potřeba v první iteraci počítat inverzi podmatice, která se nachází v prvních n řádcích matice A . Je tedy nutné zajistit, aby tato podmatice byla regulární. Tento problém jsme vyřešili přičtením konstantního násobku jednotkové matice k matici A . To zároveň zajišťuje plnou sloupcovou hodnotu matice A a tedy i pozitivní definitnost matice $A^T A$ dle věty 10.

V praxi dosáhneme téhož prvním přeuspořádáním řádků matice kombinatorickými metodami, což by ale šlo už nad rámec této práce.

6.4.2 Funkce Maxvol

Funkci Maxvol jsme implementovali podle algoritmu 3 v sekci 4.3. Její hlavní součástí je výpočet matice $B = A \hat{A}^{-1}$, který v dalším popíšeme.

Algoritmus 4: Pseudokód experimentů

```
Input:  $A \in \mathbb{R}^{m \times n}$ ,  $b \in \mathbb{R}^m$ ,  $c \in \mathbb{R}$ ,  $\text{maxk} \in \mathbb{N}$   
// regularizace:  
for  $i = 1, \dots, n$  do  
     $a_{ii} := a_{ii} + c$   
end  
// hledání dominantní podmatice s maximálně  $\text{maxk}$  iteracemi:  
 $A_D = \text{Maxvol}(A, \text{maxk})$   
// matice a pravá strana soustavy normálních rovnic:  
 $\tilde{A} = A^T A$   
 $\tilde{b} = A^T b$   
// předpodmínění:  
 $L = \text{icvl}(A_D^T A_D)$   
// řešení soustavy pomocí sdružených gradientů s  
    předpodmíněním L:  
 $\text{CG}(\tilde{A}, \tilde{b}, L)$ 
```

Díky řídkému formátu matice A máme jednoduchý přístup k řádkům matice, nikoli však k jejím sloupcům. Pomocí ekvivalentních úprav tedy rovnici pro matici B přepíšeme do tvaru

$$\hat{A}^T B^T = A^T.$$

Tuto rovnost je možné přepsat m soustav rovnic

$$\hat{A}^T b_i^T = a_i^T, \quad i = 1, \dots, m,$$

kde b_i a a_i značí i -tý řádek matice B a A . Tyto rovnice následně vyřešíme pomocí LU rozkladu matice \hat{A}^T .

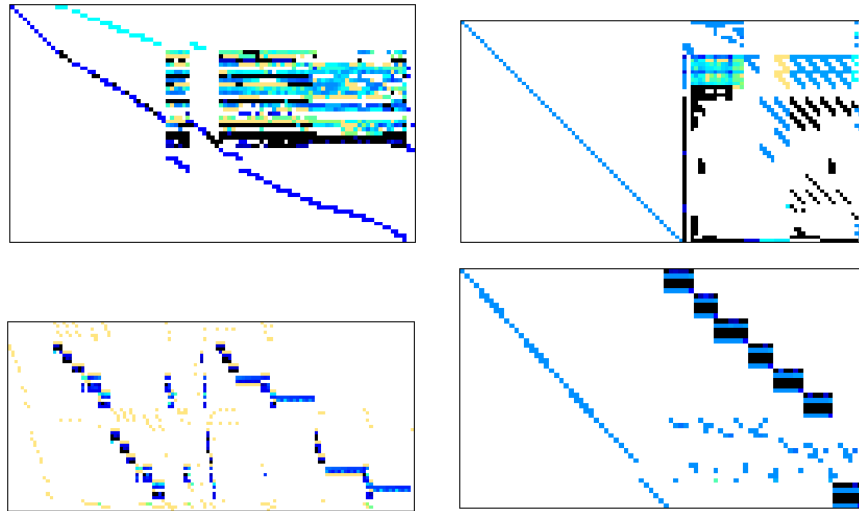
Efektivnější alternativou by bylo použít Sherman-Morrisonův vzorec z věty 7, jak je popsáno v sekci 4.3. Pro použití na naše testovací matice to však není nutné.

Pro zastavovací kritérium funkce Maxvol jsme volili $\max_{i,j} |b_{ij}| \geq 1 + 10^{-5}$. Tato volba prakticky neovlivní výslednou matici A_D , může však zabránit zbytečným iteracím zaviněným zaokrouhlovacími chybami.

6.5 Výsledky

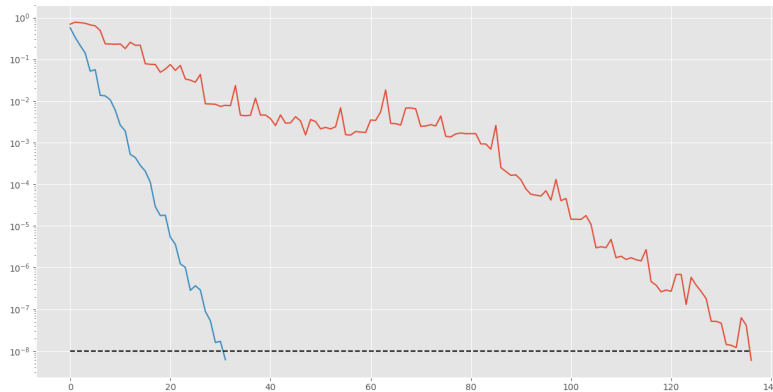
Matice pro experimenty jsme včetně jejich obrázků získali ze SuiteSparse Matrix Collection. Tyto matice jsme transponovali, aby odpovídaly našemu požadavkům $m > n$. Experimentovali jsme s maticemi `lp_beaconfd`, `lp_israel`, `lp_share1b` a `lp_share2b`, které jsou zobrazeny na obrázku 6.2. Všechny tyto matice mají hustší sloupce, které je výrazně charakterizují. Po transponování mají tedy husté řádky, což je postačujícím předpokladem pro to, aby naše metoda mohla fungovat dobře.

Pro každou matici jsme porovnávali počet iterací CG s předpodmíněním dominantní podmaticí A_D a původní podmaticí \hat{A} v prvních n řádcích matice. Pravou stranu b problému $Ax \approx b$ jsme zvolili jako $b = (1, 2, \dots, m)^T$. Jednotlivé počty iterací a grafy norm reziduí jsou zaznamenány v následujících tabulkách a obrázcích 6.3, 6.4, 6.5 a 6.6.



Obrázek 6.2: Testovací matice `lp_beaconfd`, `lp_israel`, `lp_share1b` a `lp_share2b`

<code>lp_beaconfd</code> s $c = 1.5$	iterace Maxvol	iterace CG
předpodmínění CG pomocí \hat{A} :	0	137
předpodmínění CG pomocí A_D :	19	32



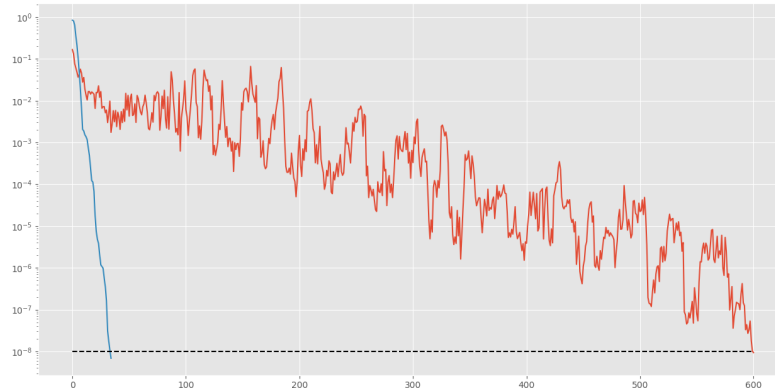
Obrázek 6.3: rezidua CG pro matici `lp_beaconfd`

Výsledky, které jsme z experimentů získali napovídají, že předpodmínění pomocí dominantní podmatice může výrazně snížit počet iterací sdružených gradientů. Záleží však velmi na struktuře původní matice. Pokud by například matice obsahovala náhodně rozmístěné prvky stejné hodnoty, předpodmínění by nejspíš nemělo moc velký účinek. Je tedy třeba problém znát předem a předpodmínění mu adaptivně uzpůsobit.

Je také potřeba uvážit výpočetní a časovou náročnost iterací funkce Maxvol, které jsou dražší než iterace CG. Tento postup by byl tedy výhodný, pokud bychom používali určitou matici pro více výpočtů s různými pravými stranami.

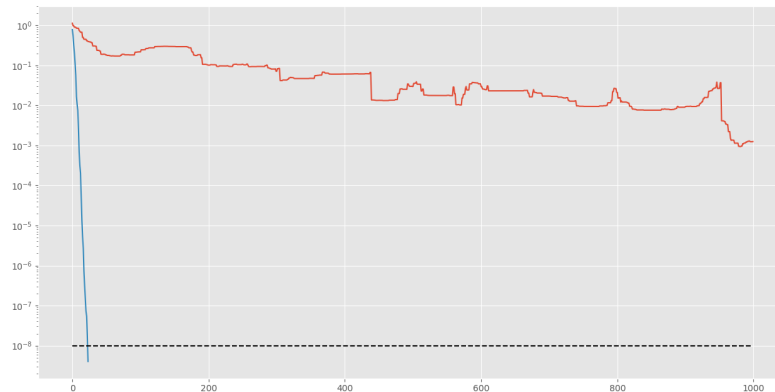
Další možností je počítat jen několik iterací Maxvol a výsledek použít pro škálování, nebo v kombinaci s dalšími předpodmíněními. Myslíme si, že v tom by mohl být hlavní nový přínos této metody.

lp_israel s $c = 0$	iterace Maxvol	iterace CG
předpodmínění CG pomocí \hat{A} :	0	601
předpodmínění CG pomocí A_D :	83	35



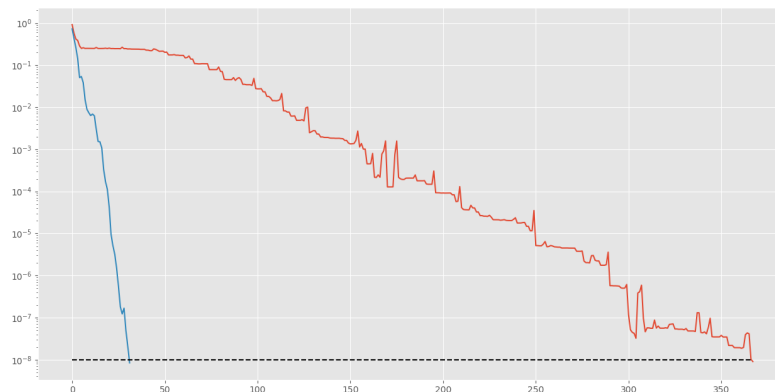
Obrázek 6.4: rezidua CG pro matici lp_israel

lp_share1b s $c = 10$	iterace Maxvol	iterace CG
předpodmínění CG pomocí \hat{A} :	0	2385
předpodmínění CG pomocí A_D :	35	24



Obrázek 6.5: rezidua CG pro matici lp_share1b

lp_share2b s $c = 0.5$	iterace Maxvol	iterace CG
předpodmínění CG pomocí \hat{A} :	0	368
předpodmínění CG pomocí A_D :	48	32



Obrázek 6.6: rezidua CG pro matici lp_share2b

Závěr

V této práci jsme se věnovali čtyřem různým přístupům k aproximaci pomocí matic nízkých hodnotí. Singulární rozklad je nejstarším z nich. Jeho výpočet je časově náročný, proto není příliš vhodný pro použití na matice velkých rozměrů. Jeho výhodou naopak je, že z něj lze vyčíst obor hodnot, jádro i singulární čísla rozložené matice.

Dále jsme se věnovali hierarchickým maticím. Ty jsou silným nástrojem například pro řešení integrálních rovnic či parciálních diferenciálních rovnic. Popsaný přístup využívá symetrií problému a tím snižuje potřebné paměťové i časové nároky. Způsob, jakým se tyto matice konstruují, je však velmi svazuje s konkrétními aplikacemi. Proto tento přístup není příliš vhodný pro použití na obecné matice.

Přístupy založené na pravděpodobnosti jsou vhodné pro aproximace oboru hodnot matice a řešení problému nejmenších čtverců, což dosvědčuje algoritmus Blendenpik, který využívá náhodný výběr řádků. Výsledky, které autoři článku [1] uvádějí, jsou velmi dobré, obáváme se však, že předpoklad $m \gg n$ je příliš omezující. Pravděpodobnostní metody však určitě naleznou své využití na nějakých užších třídách matic.

Algebraické přístupy, které v této práci reprezentuje pseudo-skeletní rozklad, jsou použitelné na jakýkoli typ matice. Pro správnou volbu řádků a sloupců prozatím nebyl nalezen ideální algoritmus, známe však alespoň odhady chyb takové aproximace a také algoritmus Maxvol, který umí najít dominantní podmatici $r \times r$, pokud je matice rozměrů $m \times r$ či $r \times n$. Tento algoritmus se ukázal jako užitečný při předpodmiňování metody sdružených gradientů, kterou jsme používali v experimentech. Výsledky, které jsme získali, napovídají, že pro matice s hustými řádky může předpodmiňování pomocí algoritmu Maxvol výrazně snížit počet iterací sdružených gradientů při řešení problému nejmenších čtverců. Pravděpodobně by bylo vhodné počítat jen několik iterací Maxvolu a následně ho kombinovat s dalším předpodmiňováním. To by mohlo být předmětem dalšího zkoumání v této oblasti.

Seznam použité literatury

- [1] AVRON, H., MAYMOUNKOV, P. a TOLEDO, S. (2010). Blendenpik: Supercharging lapack's least-squares solver. *SIAM Journal on Scientific Computing*, **32**(3), 1217–1236.
- [2] BARTO, L. a TŮMA, J. Skripta z lineární algebry. Studijní text k přednášce NMAG102, Matematicko-fyzikální fakulta Univerzity Karlovy, http://www.karlin.mff.cuni.cz/~tuma/LA2-17/skripta_la6.pdf.
- [3] BJÖRCK, A. K. (1996). *Numerical methods for least squares problems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA. ISBN 0-89871-360-9. doi: 10.1137/1.9781611971484. URL <https://doi.org/10.1137/1.9781611971484>.
- [4] BÖRM, S., GRASEDYCK, L. a HACKBUSCH, W. (2003). Hierarchical matrices. *Lecture notes of the Max Planck Institute for Mathematics in the Sciences*, **21**, 2003.
- [5] DUINTJER TEBBENS, J., HNĚTYNKOVÁ, I., PLEŠINGER, M., STRAKOŠ, Z. a TICHÝ, P. (2012). *Analýza metod pro maticové výpočty*. První vydání. Matfyzpress, Praha. ISBN 978-80-7378-201-6.
- [6] GOLUB, G. H. a VAN LOAN, C. F. (1996). *Matrix Computations. 3rd edition*. The Johns Hopkins University Press, Baltimore and London.
- [7] GOREINOV, S., OSELEDETS, I., SAVOSTYANOV, D., TYRTYSHNIKOV, E. a ZAMARASHKIN, N. (2010). How to find a good submatrix. In *Matrix Methods: Theory, Algorithms And Applications: Dedicated to the Memory of Gene Golub*, pages 247–256. World Scientific.
- [8] GOREINOV, S. A., TYRTYSHNIKOV, E. E. a ZAMARASHKIN, N. L. (1997). A theory of pseudoskeleton approximations. *Linear algebra and its applications*, **261**(1-3), 1–21.
- [9] HACKBUSCH, W. (1999). A sparse matrix arithmetic based on H-matrices. part I: Introduction to H-matrices. *Computing*, **62**(2), 89–108.
- [10] HALKO, N., MARTINSSON, P.-G. a TROPP, J. A. (2011). Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, **53**(2), 217–288.
- [11] MEURANT, G. a STRAKOŠ, Z. (2006). The Lanczos and conjugate gradient algorithms in finite precision arithmetic. *Acta Numerica*, **15**, 471–542.
- [12] SAAD, Y. (1990). SPARSKIT: A basic tool kit for sparse matrix computations. Technical Report 90–20, Research Institute for Advanced Computer Science, NASA Ames Research Center, Moffet Field, CA.
- [13] TYRTYSHNIKOV, E., GOREINOV, S. a ZAMARASHKIN, N. (1995). Pseudo skeleton approximations. *Doklady Akademii Nauk*, **343**(2), 151–152.