

ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA APLIKOVANÝCH VĚD
KATEDRA MATEMATIKY

DIPLOMOVÁ PRÁCE

Odhady normy chyby v metodě sdružených
gradientů

Prohlášení

Prohlašuji, že jsem svou diplomovou práci vypracovala samostatně s použitím literatury a pramenů uvedených v seznamu, který je součástí této práce

V Plzni dne _____

podpis

Poděkování

Na úvod bych chtěla poděkovat především vedoucímu diplomové práce RNDr. Petru Tichému Ph.D. za odborné vedení, za poskytování věcných připomínek, cenných rad a nápadů a hlavně za velkou trpělivost, se kterou mě zasvěcoval do dané problematiky a za veškerý čas, který mi věnoval.

Chtěla bych poděkovat i celé své rodině a příbuzným za obrovskou podporu nejen při psaní diplomové práce, ale i za to, že mě podporovali během celého studia a pomáhali zvláště s hlídáním mých dětiček.

Obsah

| | | |
|----------|--|-----------|
| 1 | Úvod | 6 |
| 2 | Metoda sdružených gradientů - Lanczosův algoritmus | 8 |
| 2.1 | Metody řešení soustavy lineárních rovnic | 8 |
| 2.2 | Metoda sdružených gradientů | 9 |
| 2.3 | Algoritmus metody sdružených gradientů | 11 |
| 2.4 | Konvergence metody sdružených gradientů | 11 |
| 2.5 | Zastavovací kritéria | 14 |
| 2.6 | Lanczosův algoritmus | 16 |
| 2.7 | Vztah mezi CG a Lanczosovým algoritmem | 17 |
| 3 | Ortogonální polynomy a Gaussova kvadratura | 20 |
| 3.1 | Ortogonální polynomy | 20 |
| 3.2 | Kvadraturní pravidla | 24 |
| 4 | Odhady normy chyby | 29 |
| 4.1 | Souvislosti mezi CG a Gaussovou kvadraturou | 29 |
| 4.1.1 | Faktorizace třídiagonální matice | 34 |
| 4.1.2 | Faktorizace posunuté třídiagonální matice | 34 |
| 4.1.3 | Jak počítat různé kvadratury pomocí modifikované Jacobiho matice | 36 |
| 4.2 | Algoritmická realizace | 36 |
| 5 | Numerické experimenty | 41 |
| 5.1 | Výhody a nevýhody dolního a horního odhadu | 42 |
| 5.1.1 | Dolní odhad | 42 |
| 5.1.2 | Horní odhad | 44 |
| 5.2 | Odhad založený na Gauss-Radau kvadratuře | 45 |
| 5.2.1 | Volba μ | 45 |
| 5.2.2 | „Místo nestability“ | 50 |
| 5.3 | Adaptivní volba d | 55 |
| 5.3.1 | Ideální d | 55 |
| 5.3.2 | Algoritmy pro adaptivní volbu d | 55 |
| 5.4 | Použité matice | 62 |

| | |
|----------------------------|-----------|
| 6 Závěr | 63 |
| Literatura | 65 |
| A Ovládání programu | 66 |

Kapitola 1

Úvod

Ve své diplomové práci jsem se zabývala odhadováním normy chyby v metodě sdružených gradientů. Pro počítání odhadů jsem používala různé druhy kvadraturních pravidel. Cílem mé práce bylo nastudovat metodu sdružených gradientů a její souvislosti s Gaussovou kvadraturou. Dále se pak věnovat hornímu odhadu A-normy chyby založeném na Gauss-Radau kvadratuře. Jak je tento odhad citlivý na vstupní data nebo jaký vliv má na výpočet konečná aritmetika. Pomocí horního odhadu se poté pokusit navrhnout adaptivní volbu parametru d .

Práce je rozdělena do čtyř částí. V první kapitole je popsána metoda sdružených gradientů (CG conjugate gradient) odvozená pomocí minimalizace funkcionálu. Při popisu jsem vycházela převážně ze skript [2], kapitola 8. Jsou zde popsány některé vlastnosti metody sdružených gradientů, jak je možné odhadovat rychlost konvergence metody nebo jaká lze použít zastavovací kritéria. Tedy jak můžeme měřit kvalitu aproximace x_k . Já jsem se hlavně věnovala sledování A-normy chyby, ale používají se i jiná kritéria. V další části této kapitoly je uveden Lanczosův algoritmus, který generuje ortogonální bázi vektorů pomocí tříčlenné rekurence a jeho souvislosti s metodou sdružených gradientů. Jaké jsou vztahy mezi koeficienty a vektory vznikajícími v CG a Lanczosově algoritmu.

V následující kapitole jsem popsala ortogonální polynomy vzhledem ke skalárnímu součinu definovanému Riemann-Stieltjesovým integrálem. Ukázala jsem, jak je možné počítat je pomocí tříčlenné rekurence, kterou je možné zapsat Jacobiho maticí. V další části jsem popsala obecné kvadraturní pravidlo a jak aproximací Riemann-Stieltjesova integrálu získáme Gaussovou kvadraturu, ve které jsou uzly určeny jako kořeny ortogonálních polynomů nebo Gauss-Radau a Gauss-Lobatto kvadraturu, ve které si jeden resp. dva uzly předepíšeme a zbývající uzly určíme tak, abychom dosáhli co nejvyšší algebraické přesnosti. Dále jsem uvedla, jak je možné pomocí tříčlenné rekurence generující ortonormální polynomy počítat uzly a váhy Gaussovy, Gauss-Radau a Gauss-Lobatto kvadratury a jaký tvar má příslušná Jacobiho matice na základě rekurentních vztahů.

Pochopení souvislostí těchto jednotlivých témat umožňuje ve třetí části pohled na metodu sdružených gradientů jako na Gaussovou kvadraturu, jejíž uzly jsou vlastní čísla

Jacobiho matice a váhy jsou určeny pomocí vlastních vektorů Jacobiho matice, kterou získáváme v průběhu CG pomocí koeficientů metody sdružených gradientů. Analogie mezi aproximací Riemann-Stieltjesova integrálu Gassovou kvadraturou a metodou sdružených gradientů dává možnost odhadovat A-normu chyby řešení $(x - x_k)$ v k -tém kroku CG. V poslední části této kapitoly je uvedena LDL^T faktorizace třídiagonální matice, protože v metodě sdružených gradientů nevzniká přímo Jacobiho matice, ale právě její LDL^T faktorizace. Souvislost mezi koeficienty metody sdružených gradientů a LDL^T faktorizací modifikované Jacobiho matice příslušné Gaussově, Gauss-Radau a Gauss-Lobatto kvadratuře nám umožňuje vytvořit efektivní algoritmus pro počítání horních a dolních odhadů pomocí jednotlivých kvadraturních pravidel.

V poslední kapitole jsou popsány numerické výpočty v programu MATLAB™. Počítala jsem horní a dolní odhady A-normy chyby založené na Gaussově, Gauss-Radau a Gauss-Lobatto kvadratuře. Dále jsem sledovala chování horního odhadu pomocí Gauss-Radau kvadratury a jeho citlivosti na předepsaný uzel. Zkoumala jsem jeho numerické chování při výpočtech v konečné aritmetice. Je-li předepsaný uzel blízký nejmenšímu vlastnímu číslu λ_1 matice A , může během výpočtu dojít k numerické nestabilitě. Jak odhalit místo nestability, popř. jak horní odhad v tomto místě vylepšit, tomu jsem se věnovala v dalších experimentech. Na závěr práce jsem se snažila navrhnout postup, jak volit adaptivně parametr d a tím zlepšit dolní odhad, o kterém víme, že je numericky stabilní.

Kapitola 2

Metoda sdružených gradientů - Lanczosův algoritmus

V této kapitole shrneme, jaké jsou základní metody pro řešení soustavy lineárních rovnic $Ax = b$ a jaké jsou mezi nimi rozdíly. Poté si odvodíme pomocí minimalizace funkcionálu $\mathcal{F}(x)$ metodu sdružených gradientů a ukážeme některé její vlastnosti. V poslední části této kapitoly uvedeme Lanczosův algoritmus a jeho souvislosti s metodou sdružených gradientů.

2.1 Metody řešení soustavy lineárních rovnic

Soustavy lineárních rovnic $Ax = b$ je možné řešit různými způsoby. Metody řešení můžeme rozdělit do dvou základních skupin - metody přímé a iterační.

1. Metody přímé

Pomocí přímé metody určíme řešení soustavy v konečném počtu kroků. Pokud řešíme soustavy přímou metodou, musíme vždy provést celý výpočet, na konci kterého získáme řešení. Přímé metody jsou většinou paměťově náročné. Proto jsou těžko použitelné pro velké systémy. Mezi základní metody z této skupiny patří Gaussova eliminační metoda, LU rozklad a jejich modifikace.

2. Metody iterační

Iterační metody řeší soustavy lineárních rovnic tak, že si zvolíme počáteční aproximaci a snažíme se postupně přibližovat k přesnému řešení. Oproti přímým metodám můžeme iterační metodu zastavit po k krocích, dosáhne-li aproximace řešení námi požadované přesnosti. Tyto metody jsou vhodné pro velké řídké matice. Iterační metody se dají ještě rozdělit na dvě skupiny

- klasické, jsou metody, které počítají novou aproximaci x_{k+1} pomocí předchozí iterace x_k . Sem můžeme zařadit např. Gauss-Seidlovu metodu, Jacobiho metodu nebo metodu SOR.

- krylovovské, jsou metody, které hledají aproximace v Krylovově podprostoru pomocí bázových vektorů. Můžeme je ještě rozdělit na metody, které řeší soustavu s nesymetrickou maticí \mathbf{A} nebo metody, které řeší soustavu se symetrickou maticí \mathbf{A} . Hlavní rozdíl mezi těmito metodami je, že metody s nesymetrickou maticí generují aproximaci pomocí všech předchozích bázových vektorů, kdežto metody řešící soustavu se symetrickou maticí generují aproximaci pouze pomocí dvou předchozích bázových vektorů, což je podstatně méně pamětově náročné. Mezi metody, které řeší soustavu se symetrickou pozitivně definitní maticí, patří metoda sdružených gradientů.

2.2 Metoda sdružených gradientů

Metoda sdružených gradientů - dále budeme používat zkratku CG (z anglického conjugate gradient) - řeší soustavy lineárních algebraických rovnic $\mathbf{A}x = b$ s reálnou symetrickou pozitivně definitní maticí popřípadě s hermitovskou pozitivně definitní maticí. Metodu sdružených gradientů lze odvodit několika způsoby, např. z Lanczosova algoritmu nebo pomocí minimalizace funkcionálu. Metodu sdružených gradientů zformulovali v roce 1952 Hestenes a Stiefel [8]. Již tehdy podrobně uvedli vztahy mezi minimalizací funkcionálu a CG, ale i vztahy mezi CG a Gaussovou kvadraturou. Dokonce naznačili, jaké mohou být problémy v konečné aritmetice počítače.

Nejprve ukážeme odvození CG pomocí minimalizace funkcionálu viz [2]. Uvažujme soustavu lineárních algebraických rovnic

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

kde $\mathbf{A} \in \mathbb{R}^{n \times n}$ je symetrická pozitivně definitní matice a $b \in \mathbb{R}^n$ je vektor pravé strany. Úlohu můžeme převést na problém minimalizace kvadratického funkcionálu

$$\mathcal{F}(x) = \frac{1}{2}x^T \mathbf{A}x - x^T b,$$

kde platí

$$\nabla \mathcal{F}(x) = \mathbf{A}x - b.$$

$\mathcal{F}(x)$ nabývá minima v bodě $\nabla \mathcal{F}(x) = 0$, tj. v bodě x , pro který se $\mathbf{A}x = b$. Uvažujme aproximaci x_k . Funkcionál v bodě x_k je pak určen vztahem

$$\begin{aligned} \mathcal{F}(x_k) &= \frac{1}{2}(x - x_k)^T \mathbf{A}(x - x_k) - \frac{1}{2}x^T \mathbf{A}x \\ &= \frac{1}{2}\|x - x_k\|_{\mathbf{A}}^2 - \frac{1}{2}\|x\|_{\mathbf{A}}^2. \end{aligned}$$

Aproximaci v k -tém kroku konstruujeme podle rekurentní formule

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

kde p_{k-1} je směrový vektor v kroku k a koeficient γ_{k-1} určíme tak, aby vektor x_k byl bodem minima $\mathcal{F}(z(\gamma))$ na přímce $z(\gamma) = x_{k-1} + \gamma p_{k-1}$. Z nutné a postačující podmínky minima položíme derivaci podle γ rovnou nule.

$$\begin{aligned}\mathcal{F}(x_{k-1} + \gamma p_{k-1}) &= \|x - x_{k-1}\|_{\mathbf{A}}^2 - 2\gamma p_{k-1}^T r_{k-1} + \gamma^2 p_{k-1}^T \mathbf{A} p_{k-1}, \\ 0 &= -2p_{k-1}^T r_{k-1} + 2\gamma p_{k-1}^T \mathbf{A} p_{k-1}.\end{aligned}$$

Minima je dosaženo pro koeficient γ , který je dán vztahem

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

Okamžitým důsledkem volby je ortogonalita rezidua v k -tém kroku r_k a směrového vektoru p_k

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

Musíme tedy ještě určit směrové vektory p_k . Nejjednodušší počáteční volbou je $p_0 \equiv r_0$. Pokud bychom volili $p_k = r_k$ pro $k = 1, 2, \dots$, dostali bychom metodu největšího spádu, která nám nedává žádné minimalizační vlastnosti na vícedimenzionálním prostoru. Nejjednodušší volbou, abychom takovéto vlastnosti dosáhli, je volit směrový vektor jako kombinaci nového rezidua a předchozího směrového vektoru

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

Nezávisle na volbě δ_k

$$p_k^T r_k = r_k^T r_k + \delta_k p_{k-1}^T r_k = r_k^T r_k = \|r_k\|^2.$$

Iterační proces se zastaví, pokud je buď $p_k = 0$ nebo $\gamma_k = 0$. Koeficienty δ_k volíme tak, aby posloupnost směrů $p_k, k = 0, 1, \dots$ tvořila systém \mathbf{A} -ortogonálních vektorů. Abychom podmínku \mathbf{A} -ortogonalitě směrových vektorů splnili, volíme δ_k tak, aby dva po sobě jdoucí vektory byly \mathbf{A} -ortogonální.

$$p_{k-1}^T \mathbf{A} p_k = 0$$

$$\delta_k = -\frac{p_{k-1}^T \mathbf{A} r_k}{p_{k-1}^T \mathbf{A} p_{k-1}},$$

což nám dává lokální \mathbf{A} -ortogonalitu dvou po sobě jdoucích směrových vektorů p_{k-1} a p_k . Lze ale ukázat viz např. [2], že zvolíme-li takovýto δ_k nový směrový vektor, bude \mathbf{A} -ortogonální nejen k předchozímu směrovému vektoru, ale i ke všem předchozím směrovým vektorům $p_j, j = 0, 1, \dots$ a tedy nám zaručuje globální \mathbf{A} -ortogonalitu. Dosažením do rekurentního vztahu pro p_k dostaneme

$$p_k = r_k - \frac{p_{k-1}^T \mathbf{A} r_k}{p_{k-1}^T \mathbf{A} p_{k-1}} p_{k-1},$$

$$p_{k-1}^T r_{k-1} = \|r_{k-1}\|^2.$$

Využijeme-li toho, že platí

$$-\mathbf{A}p_{k-1} = \frac{r_k - r_{k-1}}{\gamma_{k-1}} = (r_k - r_{k-1}) \frac{p_{k-1}^T \mathbf{A} p_{k-1}}{p_{k-1}^T r_{k-1}},$$

pak po dosazení získáme známé tvary pro koeficienty γ a δ :

$$\gamma_{k-1} = \frac{r_{k-1}^T r_{k-1}}{p_{k-1}^T \mathbf{A} p_{k-1}}$$

$$\delta_k = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}.$$

2.3 Algoritmus metody sdružených gradientů

Na základě uvedených vztahů můžeme napsat algoritmus pro řešení soustavy rovnic $\mathbf{A}x = b$ metodou sdružených gradientů:

Algoritmus 1: CG

vstupní data \mathbf{A}, b, x_0

$$r_0 = b - \mathbf{A}x_0$$

$$p_0 = r_0$$

$$k = 1$$

for $k=1,2,\dots$

$$\gamma_{k-1} = \frac{r_{k-1}^T r_{k-1}}{p_{k-1}^T \mathbf{A} p_{k-1}}$$

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

$$r_k = r_{k-1} - \gamma_{k-1} \mathbf{A} p_{k-1}$$

$$\delta_k = \frac{r_k^T r_k}{r_{k-1}^T r_{k-1}}$$

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

test kvality aproximace

end

2.4 Konvergence metody sdružených gradientů

Metoda sdružených gradientů se zastaví, kdykoliv je $\|r_k\| = 0$, tj. $x_k = x$, což musí v přesné aritmetice nastat v nejvýše n krocích. Nás ale většinou zajímá, jak rychle se aproximace x_1, x_2, \dots přiblíží „dostatečně blízko“ k přesnému řešení, kde pojem „dostatečně blízko“ většinou závisí na konkrétním řešeném problému.

Mějme k -tý Krylovův podprostor generovaný maticí \mathbf{A} a vektorem v

$$\mathcal{K}_k(\mathbf{A}, v) \equiv \text{span}\{v, \mathbf{A}v, \mathbf{A}^2v, \dots, \mathbf{A}^{k-1}v\}.$$

Pokud nedojde k zastavení výpočtu v důsledku dosažení přesného řešení, pak z ortogonalit reziduí r_{k-1} a \mathbf{A} -ortogonalit směrových vektorů p_{k-1} plyne, že jsou lineárně nezávislé a tedy prostor $\text{span}\{r_0, \dots, r_{k-1}\}$ a $\text{span}\{p_0, \dots, p_{k-1}\}$ jsou stejné dimenze k a tedy generují stejný Krylovův podprostor

$$\text{span}\{r_0, \dots, r_{k-1}\} = \mathcal{K}_k(\mathbf{A}, r_0) = \text{span}\{p_0, \dots, p_{k-1}\}.$$

Potom k -tá aproximace x_k splňuje vztah

$$x_k = x_0 + \sum_{i=1}^{k-1} \gamma_i p_i \in x_0 + \mathcal{K}_k(\mathbf{A}, r_0) \quad (2.1)$$

a vektor rezidua r_k je ortogonální ke k -tému Krylovovu podprostoru, neboli vektor $(x - x_k)$ je \mathbf{A} -ortogonální ke k -tému Krylovovu podprostoru

$$\begin{aligned} r_k \perp \mathcal{K}_k(\mathbf{A}, r_0) &= \text{span}\{r_0, \dots, r_{k-1}\}, \\ x - x_k \perp_{\mathbf{A}} \mathcal{K}_k(\mathbf{A}, r_0). \end{aligned} \quad (2.2)$$

Ze vztahů (2.1) a (2.2) vyplývá, že minimalizační vlastnost metody sdružených gradientů můžeme vyjádřit následovně

$$\|x - x_k\|_{\mathbf{A}} = \min_{y \in x_0 + \mathcal{K}_k(\mathbf{A}, r_0)} \|x - y\|_{\mathbf{A}}.$$

Krylovův podprostor $\mathcal{K}_k(\mathbf{A}, r_0)$ není nic jiného než množina všech lineárních kombinací vektorů $r_0, \mathbf{A}r_0, \dots, \mathbf{A}^{k-1}r_0$. Tzn. že libovolný vektor z tohoto prostoru lze psát ve tvaru $q(\mathbf{A})r_0$, kde q je polynom stupně nejvýše $k - 1$. Řešení x_k můžeme psát ve tvaru polynomu $x_k = x_0 + q(\mathbf{A})r_0$ a stejně tak chybu v k -tém kroku $e^{(k)} \equiv x - x_k$ lze psát ve tvaru

$$e^{(k)} = (x - x_0) - (x_k - x_0) = x - x_0 - q(\mathbf{A})r_0 = e^{(0)} - \mathbf{A}q(\mathbf{A})e^{(0)} = p(\mathbf{A})e^{(0)},$$

kde

$$p(\mathbf{A}) = \mathbf{I} - \mathbf{A}q(\mathbf{A})$$

je polynom stupně nejvýše k s vlastností $p(0) = 1$. Z minimalit \mathbf{A} -normy k -té chyby $e^{(k)}$ plyne

$$\|x - x_k\|_{\mathbf{A}} = \|e^{(k)}\|_{\mathbf{A}} = \min_{p \in \pi_k} \|p(\mathbf{A})e^{(0)}\|_{\mathbf{A}}, \quad (2.3)$$

kde π_k je množina všech polynomů stupně nejvýše k , pro které platí, že $p(0) = 1$. Uvažujeme spektrální rozklad matice $\mathbf{A} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$, kde $\mathbf{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_n)$ a $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}$ a definujeme si vektor

$$\omega = (\omega_1, \dots, \omega_n)^T \equiv \mathbf{Q}^T r_0,$$

který je projekcí počátečního rezidua r_0 do prostoru vlastních vektorů matice \mathbf{A} . Využijeme-li vztahů mezi euklidovskou normou a \mathbf{A} -normou vektoru v $\|v\|_{\mathbf{A}} = \|\mathbf{A}^{\frac{1}{2}}v\|$, můžeme psát

$$\begin{aligned}\|p(\mathbf{A})e^{(0)}\|_{\mathbf{A}}^2 &= \|p(\mathbf{A})\mathbf{A}^{1/2}e^{(0)}\|^2 = \|p(\mathbf{A})\mathbf{A}^{1/2}r_0\|^2 \\ &= \|p(\mathbf{A})\mathbf{A}^{1/2}\omega\|^2 = \sum_{j=1}^n \frac{|\omega_j|^2}{\lambda_j} p(\lambda_j)^2\end{aligned}$$

a chybu v k -té iteraci CG ve vztahu (2.3) můžeme přepsat

$$\|x - x_k\|_{\mathbf{A}} = \|e^{(k)}\|_{\mathbf{A}} = \min_{p \in \pi_k} \left(\sum_{j=1}^n \frac{|\omega_j|^2}{\lambda_j} p(\lambda_j)^2 \right)^{1/2}. \quad (2.4)$$

\mathbf{A} -normu chyby v k -tém kroku CG s využitím Čebyševových polynomů, lze shora omezit pomocí čísla podmíněnosti matice \mathbf{A} . Platí viz [12]

$$\|x - x_k\|_{\mathbf{A}} \leq 2 \left(\frac{\sqrt{\kappa_2(\mathbf{A})} - 1}{\sqrt{\kappa_2(\mathbf{A})} + 1} \right)^k \|x - x_0\|_{\mathbf{A}}$$

a κ_2 označuje číslo podmíněnosti matice \mathbf{A} vzhledem k $\|\cdot\|_2$ spektrální normě

$$\kappa_2(\mathbf{A}) = \|\mathbf{A}\|_2 \|\mathbf{A}^{-1}\|_2.$$

Relativní \mathbf{A} -normu chyby můžeme odhadnout shora následovně

$$\frac{\|e^{(k)}\|_{\mathbf{A}}}{\|e^{(0)}\|_{\mathbf{A}}} \leq 2 \left(\frac{\sqrt{\kappa_2(\mathbf{A})} - 1}{\sqrt{\kappa_2(\mathbf{A})} + 1} \right)^k. \quad (2.5)$$

Tento odhad nám říká, že pokud je číslo podmíněnosti $\kappa_2(\mathbf{A})$ velmi malé, konverguje metoda sdružených gradientů rychle. Je-li ale číslo podmíněnosti velké, žádnou takovou informaci nemáme, protože lze najít matice, které mají velmi velké číslo podmíněnosti a přesto metoda sdružených gradientů konverguje rychle.

V praktických výpočtech se snažíme zvýšit rychlost konvergence tím, že použijeme předpodmínění maticí \mathbf{P}

$$\mathbf{P}^{-1}\mathbf{A}\mathbf{P}^{-T}\mathbf{P}^T x = \mathbf{P}^{-1}b$$

Matici \mathbf{P} volíme tak, aby byla regulární a umožnila nám zrychlit konvergenci CG. Jednou z možností je snažit se, aby číslo podmíněnosti matice $\mathbf{P}^{-1}\mathbf{A}\mathbf{P}^{-T}$ bylo menší než číslo podmíněnosti matice \mathbf{A} . Pro předpodmínování je možné použít např. neúplný Choleského rozklad.

2.5 Zastavovací kritéria

Existují různé možnosti, jak testovat kvalitu aproximace x_k a podle toho rozhodovat o zastavení metody sdružených gradientů. Ukázali jsme, že je přirozené soustředit se na \mathbf{A} -normu chyby, protože zachovává minimalizační vlastnosti metody sdružených gradientů. V článku [1] bylo ukázáno, že pokud např. chceme numericky pomocí metody konečných prvků řešit eliptickou parciální diferenciální rovnici v samoadjungovaném tvaru, je vhodné jako zastavovací kritérium použít relativní \mathbf{A} -normu chyby, protože odpovídá relativní chybě mezi přesným a diskrétním řešením dané parciální diferenciální rovnice. Relativní \mathbf{A} -normu chyby

$$\frac{\|x - x_k\|_{\mathbf{A}}}{\|x\|_{\mathbf{A}}}$$

během CG iterací neznáme, protože neznáme přesné řešení x . Je ale možné tuto hodnotu odhadovat pomocí kvadratických pravidel. Dále ukážeme odvození odhadu algebraickou cestou viz [2]. Jeho souvislost s Gaussovou kvadraturou si ukážeme později.

Na chybu $(x - x_j)$ lze pohlížet jako na rozklad \mathbf{A} -ortogonálních komponent a pomocí Pythagorovy věty jí můžeme zapsat takto

$$\|x - x_j\|_{\mathbf{A}}^2 - \|x - x_{j+1}\|_{\mathbf{A}}^2 = \gamma_k^2 \|p_k\|_{\mathbf{A}}^2 = \gamma_k \|r_k\|^2. \quad (2.6)$$

Sečtením vztahů (2.6) pro $j = k$ až $j = k + d$ pro $d > 0$ dostaneme

$$\|x - x_k\|_{\mathbf{A}}^2 - \|x - x_{k+d}\|_{\mathbf{A}}^2 = \sum_{j=k}^{k+d-1} \gamma_j \|r_j\|^2. \quad (2.7)$$

Výraz (2.7) lze použít k odhadu \mathbf{A} -normy chyby. Víme-li, že během d kroků poklesne \mathbf{A} -norma chyby tak, že platí

$$\|x - x_k\|_{\mathbf{A}}^2 \gg \|x - x_{k+d}\|_{\mathbf{A}}^2, \quad (2.8)$$

potom odmocnina ze sumy na pravé straně dobře aproximuje \mathbf{A} -normu chyby v k -tém kroku CG iterací

$$\|x - x_k\|_{\mathbf{A}} \approx \left(\sum_{j=k}^{k+d-1} \gamma_j \|r_j\|^2 \right)^{1/2}.$$

Dále se budeme snažit o odvození dolního odhadu relativní \mathbf{A} -normy chyby. Využijeme-li následujících vztahů

$$\begin{aligned} \|x - x_0\|_{\mathbf{A}}^2 &= \|x\|_{\mathbf{A}}^2 - 2b^T x_0 + x_0^T \mathbf{A} x_0 = \|x\|_{\mathbf{A}}^2 - b^T x_0 - (b - \mathbf{A} x_0)^T x_0 \\ \|x\|_{\mathbf{A}}^2 &= \|x - x_0\|_{\mathbf{A}}^2 + (b + r_0)^T x_0 \end{aligned}$$

a dosadíme do (2.7) dostaneme

$$\|x\|_{\mathbf{A}}^2 - \|x - x_{k+d}\|_{\mathbf{A}}^2 = (b + r_0)^T x_0 + \sum_{j=0}^{k+d-1} \gamma_j \|r_j\|^2. \quad (2.9)$$

Pokud je splněna podmínka $\|x - x_0\|_{\mathbf{A}} \leq \|x\|_{\mathbf{A}}$ platí

$$0 < \varrho_{k,d}^2 \equiv \frac{\|x - x_k\|_{\mathbf{A}}^2 - \|x - x_{k+d}\|_{\mathbf{A}}^2}{\|x\|_{\mathbf{A}}^2 - \|x - x_{k+d}\|_{\mathbf{A}}^2} \leq \frac{\|x - x_k\|_{\mathbf{A}}^2}{\|x\|_{\mathbf{A}}^2}.$$

Nerovnost platí z algebraického vztahu pro reálná čísla α, β, γ , kde $\beta > \gamma \geq 0$ a $\beta \geq \alpha$

$$\frac{\alpha - \gamma}{\beta - \gamma} \leq \frac{\alpha}{\beta}.$$

S použitím vztahů (2.7) a (2.9) je možné vyjádřit $\varrho_{k,d}$ následujícím vztahem

$$\varrho_{k,d} = \left(\frac{\sum_{j=k}^{k+d-1} \gamma_j \|r_j\|^2}{(b + r_0)^T x_0 + \sum_{j=0}^{k+d-1} \gamma_j \|r_j\|^2} \right)^{1/2}.$$

Pokud během d iterací dojde k dostatečnému poklesu \mathbf{A} -normy chyby, $\varrho_{k,d}$ nám dává dobrý dolní odhad relativní \mathbf{A} -normy chyby v k -tém kroku. Otázkou ale zůstává, jak volit parametr d , neboli jak poznat, že došlo k dostatečnému poklesu \mathbf{A} -normy chyby. Touto otázkou se budu zabývat dále, zvláště v experimentální části.

Jednou z dalších možností jak sledovat kvalitu aproximace je testovat *normovanou relativní zpětnou chybu*. Aproximaci x_k si můžeme představit jako přesné řešení porušené (perturbované) soustavy

$$(\mathbf{A} + \Delta \mathbf{A})x_k = b + \Delta b. \quad (2.10)$$

a nás zajímá nejmenší velikost možných poruch měřených pomocí $\frac{\|\Delta \mathbf{A}\|}{\|\mathbf{A}\|}$ a $\frac{\|\Delta b\|}{\|b\|}$ tak, aby x_k bylo přesným řešením (2.10). Normovanou zpětnou chybu $\sigma(x_k)$ lze vyjádřit ve tvaru viz např [2]

$$\sigma(x_k) = \frac{\|r_k\|}{\|b\| + \|\mathbf{A}\| \|x_k\|}.$$

Další možností jak testovat kvalitu řešení je chápat rezidum r_k v k -tém kroku CG, pro které platí $\mathbf{A}x_k = b - r_k$, jako změnu pravé strany

$$\mathbf{A}x_k = b - \Delta b. \quad (2.11)$$

Potom hodnota $\frac{\|r_k\|}{\|b\|}$ nám udává relativní velikost změny pravé strany a řešení x_k je přesným řešením (2.11). Některé výhody či nevýhody těchto kritérií je možné nalézt např. viz [1]

2.6 Lanczosův algoritmus

Lanczosův algoritmus nám generuje ortonormální báze v_1, \dots, v_k Krylovova podprostoru $\mathcal{K}_k(\mathbf{A}, v_1)$. Lze ho snadno odvodit z Arnoldiho algoritmu, který je založen na Gram-Schmidtově ortogonalizaci. Arnoldiho algoritmus lze reprezentovat maticově následujícím vztahem

$$\mathbf{A}\mathbf{V}_k = \mathbf{V}_k\mathbf{H}_k + h_{k+1,k}v_{k+1}e_k^T,$$

kde matice $\mathbf{A} \in \mathbb{R}^{n \times n}$, vektor $v \in \mathbb{R}^n$, $\mathbf{V}_k = [v_1, \dots, v_k]$ je matice s ortogonálními sloupci, které jsou generovány algoritmem.

$\mathbf{H}_k \in \mathbb{C}^{k \times k}$ je horní Hessenbergova matice.

$$\mathbf{H} = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & \dots & h_{1,k} \\ h_{2,1} & h_{2,2} & h_{2,3} & \dots & h_{2,k} \\ & \ddots & \ddots & \ddots & \vdots \\ & & \ddots & \ddots & h_{k-1,k} \\ & & & h_{k,k-1} & h_{k,k} \end{bmatrix}.$$

Je-li \mathbf{A} symetrická, pak platí

$$\mathbf{H}_k = \mathbf{V}_k^* \mathbf{A} \mathbf{V}_k = \mathbf{V}_k^* \mathbf{A}^* \mathbf{V}_k = (\mathbf{V}_k^* \mathbf{A} \mathbf{V}_k)^* = \mathbf{H}_k^*.$$

Potom matice \mathbf{H}_k je pouze třídiagonální. V dalším textu je označována jako \mathbf{T}_k .

$$\mathbf{T}_k = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_k \\ & & & \beta_k & \alpha_k \end{bmatrix}$$

Použijeme-li Arnoldiho algoritmus pro symetrickou matici \mathbf{A} , získáváme ortogonální bázi pouze pomocí tříčlenné rekurence. Takovýto algoritmus se nazývá Lanczosův. Maticově můžeme Lanczosův algoritmus zapsat takto

$$\mathbf{A}\mathbf{V}_k = \mathbf{V}_k\mathbf{T}_k + \beta_{k+1}v_{k+1}e_k^T.$$

Rozepíšeme-li si maticový zápis po jednotlivých sloupcích, dostaneme následující rekurentní vztah

$$\beta_{j+1}v_{j+1} = \mathbf{A}v_j - \alpha_jv_j - \beta_jv_{j-1}, \quad \text{kde } j = 1, \dots, k. \quad (2.12)$$

Lanczosův algoritmus nám tříčlennou rekurencí generuje ortogonální bázi Krylovova podprostoru, kde β_{j+1} je normalizační koeficient a koeficient α_j určíme z podmínky ortogonality vektorů $v_{j+1} \perp v_j$

$$\alpha_j = v_j^* \mathbf{A} v_j.$$

Protože matice \mathbf{A} je symetrická, lokální ortogonalita vektorů v_{j+1}, v_j, v_{j-1} nám zaručuje globální ortogonalitu vektoru v_{j+1} ke všem předchozím vektorům.

Algoritmus 2. Lanczosův algoritmus

vstupní data \mathbf{A}, v

$$v_0 = 0$$

$$v_1 = v / \|v\|$$

$$\beta_1 = 0$$

for $k = 1, 2, \dots$

$$w = \mathbf{A}v_k - \beta_{k-1}v_{k-1}$$

$$\alpha_k = v_k^T w$$

$$w = w - \alpha_k v_k$$

$$\beta_{k+1} = \|w\|$$

$$v_{k+1} = w / \beta_{k+1}$$

end

2.7 Vztah mezi CG a Lanczosovým algoritmem

Lanczosův algoritmus aplikovaný na matici \mathbf{A} a vektor v generuje ortonormální bázi $\{v_1, \dots, v_k\}$ Krylova podprostoru $\mathcal{K}_k(\mathbf{A}, r_0)$. Ukázali jsme, že pro aproximaci x_k platí $x_k \in x_0 + \mathcal{K}_k(\mathbf{A}, r_0)$, lze ji proto vyjádřit ve tvaru

$$x_k = x_0 + \mathbf{V}_k y_k,$$

kde $\mathbf{V}_k = [v_1, \dots, v_k]$ a y_k je vektor souřadnic. Reziduum v k -tém kroku můžeme zapsat

$$r_k = b - \mathbf{A}x_k = b - \mathbf{A}(x_0 + \mathbf{V}_k y_k) = r_0 - \mathbf{A}\mathbf{V}_k y_k.$$

Reziduum je ortogonální k prostoru $\mathcal{K}_k(\mathbf{A}, r_0)$ a tedy musí být ortogonální i ke všem bázovým vektorům v_1, \dots, v_k a platí

$$0 = \mathbf{V}_k^T r_k = \mathbf{V}_k^T (r_0 - \mathbf{A}\mathbf{V}_k y_k) = \|r_0\| e_1 - \mathbf{V}_k^T \mathbf{A}\mathbf{V}_k y_k = \|r_0\| e_1 - \mathbf{T}_k y_k,$$

kde \mathbf{T}_k je Jacobiho matice vznikající v průběhu Lanczosova algoritmu. Aproximaci x_k v metodě sdružených gradientů lze počítat pomocí Lanczosova algoritmu. Můžeme ji vyjádřit ve tvaru

$$x_k = x_0 + \mathbf{V}_k y_k, \quad \|r_0\| e_1 = \mathbf{T}_k y_k. \quad (2.13)$$

Je tedy vidět, že jak metoda sdružených gradientů s počátečním x_0 , tak i Lanczosův algoritmus s počátečním vektorem r_0 generují ortogonální bázové vektory w_j Krylovových podprostorů tak, že platí

$$\omega_{j+1} \in \mathcal{K}_{j+1}(\mathbf{A}, r_0) \quad \omega_{j+1} \perp \mathcal{K}_j(\mathbf{A}, r_0), \quad \text{kde } j = 1, \dots, k.$$

V metodě sdružených gradientů jsou těmito vektory rezidua r_j , v Lanczosově algoritmu to jsou ortogonální vektory v_{j+1} . Vektory w_j jsou určeny jednoznačně až na násobek. Z toho již vyplývá, že normalizovaná rezidua jsou rovna vektorům v_{j+1} až na znaménko. To určíme tak, že porovnáme algoritmy 1.CG a 2.Lanczosův algoritmus a z toho jak jsou tyto vektory počítány, dostaneme vztah mezi nimi

$$v_{j+1} = (-1)^j \frac{r_j}{\|r_j\|}, \quad j = 0, \dots, k.$$

Ještě určíme jaké jsou vztahy mezi koeficienty v Lanczosově algoritmu α_j a β_j a koeficienty vystupujícími v metodě sdružených gradientů γ_j a δ_j . Rezidua r_j a směrové vektory p_j jsou v metodě sdružených gradientů počítány ve dvou dvoučlenných rekurentních vztazích

$$r_j = r_{j-1} - \gamma_{j-1} \mathbf{A} p_{j-1}, \quad p_j = r_j + \delta_j p_{j-1}.$$

Budeme-li tyto vztahy dále upravovat, můžeme reziduum r_j vyjádřit pomocí jedné tříčlenné rekurence.

$$\begin{aligned} r_j &= r_{j-1} - \gamma_{j-1} \mathbf{A} p_{j-1} \\ &= r_{j-1} - \gamma_{j-1} \mathbf{A} (r_{j-1} + \delta_{j-1} p_{j-2}) \\ &= r_{j-1} - \gamma_{j-1} \mathbf{A} r_{j-1} - \gamma_{j-1} \delta_{j-1} \frac{r_{j-2} - r_{j-1}}{\gamma_{j-2}} \\ &= -\gamma_{j-1} \mathbf{A} r_{j-1} + \left(1 + \frac{\gamma_{j-1} \delta_{j-1}}{\gamma_{j-2}}\right) r_{j-1} - \frac{\gamma_{j-1} \delta_{j-1}}{\gamma_{j-2}} r_{j-2}. \end{aligned} \tag{2.14}$$

Využijeme-li toho, že platí

$$\sqrt{\delta_j} = \frac{\|r_j\|}{\|r_{j-1}\|} \quad \text{a} \quad v_{j+1} = (-1)^j \frac{r_j}{\|r_j\|},$$

můžeme rekurenci (2.14) viz [2] upravit na následující tvar

$$\frac{\sqrt{\delta_j}}{\gamma_{j-1}} v_{j+1} = \mathbf{A} v_j - \left(\frac{1}{\gamma_{j-1}} + \frac{\delta_{j-1}}{\gamma_{j-2}} \right) v_j - \frac{\sqrt{\delta_{j-1}}}{\gamma_{j-2}} v_{j-1}. \tag{2.15}$$

Dostali jsme tedy tvar, jehož koeficienty u v_j a v_{j-1} můžeme porovnat s tříčlennou rekurencí (2.12) a tím získáme následující vztahy

$$\beta_{j+1} = \frac{\sqrt{\delta_j}}{\gamma_{j-1}} \quad \alpha_{j+1} = \frac{1}{\gamma_{j-1}} + \frac{\delta_{j-1}}{\gamma_{j-2}}. \tag{2.16}$$

Vztahy můžeme zapsat v maticovém tvaru pomocí Jacobiho matice \mathbf{T}_k

$$\mathbf{T}_k = \begin{bmatrix} \alpha_1 & \beta_2 & 0 & & \\ & \beta_2 & \ddots & & \\ & & & \ddots & \beta_k \\ 0 & & & \beta_k & \alpha_k \end{bmatrix} \tag{2.17}$$

$$\mathbf{L}_k = \begin{bmatrix} 1 & 0 & & & \\ \sqrt{\delta_1} & \ddots & & & \\ 0 & \ddots & \ddots & & \\ & & & \ddots & \\ \sqrt{\delta_{k-1}} & & & & 1 \end{bmatrix} \quad \mathbf{D}_k = \begin{bmatrix} \frac{1}{\gamma_0} & 0 & & & \\ 0 & \frac{1}{\gamma_1} & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \frac{1}{\gamma_{k-1}} \end{bmatrix} \quad (2.18)$$

a je tedy zřejmé, že \mathbf{LDL}^T faktorizace Jacobiho matice \mathbf{T}_k můžeme zapsat pomocí koeficientů γ_j a δ_j a platí

$$\mathbf{T}_k = \mathbf{L}_k \mathbf{D}_k \mathbf{L}_k^T. \quad (2.19)$$

Kapitola 3

Ortogonalní polynomy a Gaussova kvadratura

3.1 Ortogonalní polynomy

V této části uvedeme ortogonalní polynomy na konečném intervalu¹ $[a, b] \in \mathbb{R}$ vzhledem ke skalárnímu součinu definovanému pomocí Riemann-Stieltjesova integrálu viz [6].

Definice 3.1.1. Riemann-Stieltjesův integrál reálné spojitě funkce $f(\lambda)$ podle funkce $\omega(\lambda)$ na intervalu $[a, b]$ je zapsán takto

$$\int_a^b f(\lambda) d\omega(\lambda). \quad (3.1)$$

Uvažujeme omezený uzavřený interval $[a, b]$. Necht' π je libovolné dělení tohoto intervalu tak, že $a = \rho_0 < \rho_1 < \dots < \rho_n = b$ a normu toho dělení si definujeme jako

$$\|\pi\| = \max_{j=1, \dots, n} (\rho_j - \rho_{j-1}).$$

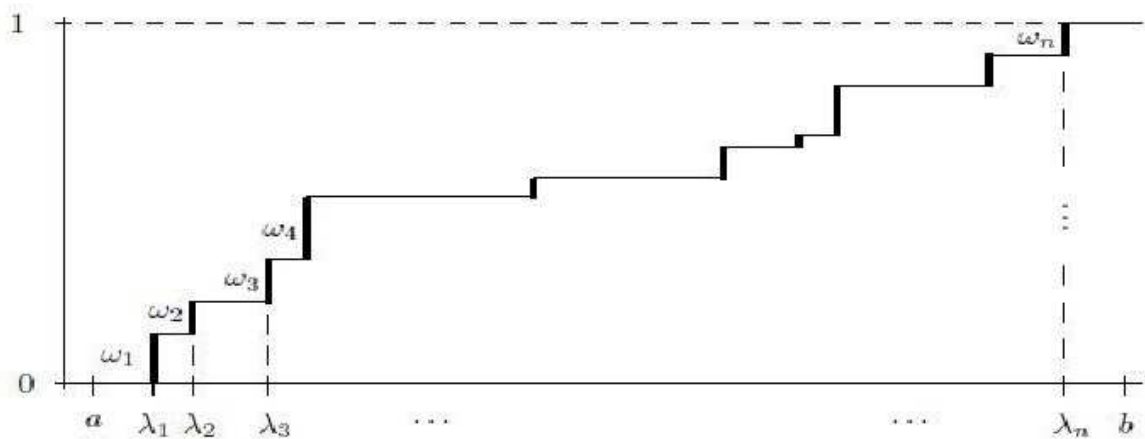
Riemann-Stieltjesův integrál je definován jako limita (pokud existuje), normy $\|\pi\|$ na intervalu $[a, b]$ jdoucí k nule, ze sumy

$$\sum_{\rho_i \in \pi} f(c_i)(\omega(\rho_{i+1}) - \omega(\rho_i)),$$

kde c_i je libovolný bod z intervalu $[\rho_i, \rho_{i+1}]$. Tedy

$$\int_a^b f(\lambda) d\omega(\lambda) \equiv \lim_{\|\pi\| \rightarrow 0} \sum_{j=1}^n f(c_j)(\omega(\rho_{j+1}) - \omega(\rho_j)).$$

¹V této kapitole b značí hranici intervalu, ne vektor pravé strany jak tomu bylo v předešlé kapitole.



Obrázek 3.1: Distribuční funkce $\omega(\lambda)$

Riemann-Stieltjesův integrál existuje, pokud funkce f je spojitá a funkce $\omega(\lambda)$ má konečnou variaci na intervalu $[a, b]$.

Riemannův integrál dostaneme, jestliže $d\omega(\lambda) = d\lambda$. Jestliže ω je spojitá a diferencovatelná, potom integrál (4.24) je rovný

$$\int_a^b f(\lambda)\omega'(\lambda)d\lambda.$$

Mějme uzavřený interval $[a, b]$. V dalším textu budeme uvažovat funkci $\omega(\lambda)$, která je neklesající, po částech konstantní a je určena uzly λ_j , které jsou uspořádány následovně $a < \lambda_1 < \dots < \lambda_n < b$ viz obrázek 3.1.

Nechť $\omega_1, \dots, \omega_n$ jsou kladná čísla, která budeme nazývat *váhy* a jsou normalizovaná tak, že

$$\sum_{j=1}^n \omega_j = 1$$

a dále budeme uvažovat, že funkce $\omega(\lambda)$ je určena body λ_i a kladnými čísly $\omega_1, \dots, \omega_n$ následovně

$$\omega(\lambda) = \begin{cases} 0 & \text{pro } \lambda < \lambda_1, \\ \sum_{i=1}^k \omega_i & \text{pro } \lambda_k \leq \lambda < \lambda_{k+1}, \quad 1 \leq k \leq n, \\ 1 & \text{pro } \lambda_n \leq \lambda. \end{cases} \quad (3.2)$$

Takovouto po částech konstantní funkci $\omega(\lambda)$ budeme nazývat *distribuční funkcí*. Potom

Riemann-Stieltjesův integrál lze zapsat ve tvaru sumy

$$\int_a^b f(\lambda) d\omega(\lambda) = \sum_{i=1}^n \omega_i f(\lambda_i). \quad (3.3)$$

Definujeme skalární součin dvou polynomů pomocí Riemann-Stieltjesova integrálu s takovouto distribuční funkcí

$$\langle p, q \rangle_\omega = \int p(\lambda) q(\lambda) d\omega(\lambda) = \sum_{i=1}^n \omega_i p(\lambda_i) q(\lambda_i). \quad (3.4)$$

Lze snadno ověřit, že $\langle p, q \rangle_\omega$ je skutečně skalární součin.

Definice 3.1.2. Mějme uzavřený interval $[a, b]$ a distribuční funkci $\omega(\lambda)$ tak, jak jsme si ji definovali výše. Řekneme, že posloupnost polynomů $\psi_i, i = 0, 1, \dots$ je ortogonální vzhledem ke skalárnímu součinu (3.4), jestliže je stupeň polynomu ψ roven i a platí

$$\int_a^b \psi_i(\lambda) \psi_j(\lambda) d\omega(\lambda) = 0 \quad \text{pro } i \neq j.$$

Norma polynomu ψ je definována takto

$$\|\psi\|_\omega = \left(\int_a^b \psi(\lambda)^2 d\omega(\lambda) \right)^{1/2}.$$

Víme, že množina takovýchto polynomů ψ_i tvoří vektorový prostor.

Věta 3.1.1. V takovémto vektorovém prostoru se skalárním součinem existuje posloupnost ortogonálních polynomů ψ_i .

Důkaz. Důkaz lze provést matematickou indukcí. Zvolme polynom ψ_0 rovný nenulové konstantní funkci. Takový vektor tvoří ortogonální bázi. Dále budeme předpokládat, že polynomy ψ_0, \dots, ψ_k tvoří ortogonální bázi a chceme ukázat, že lze určit polynom ψ_{k+1} tak, aby tvořil ortogonální bázi

$$\psi_{k+1} = \sum_{i=0}^k c_i \psi_i + c_{k+1} x^{k+1}.$$

Protože požadujeme, aby polynomy byly ortogonální dostaneme podmínky

$$\langle \psi_j, \sum_{i=0}^k c_i \psi_i + c_{k+1} x^{k+1} \rangle_\omega = 0, \quad j = 1, 2, \dots, k.$$

Indukčním předpokladem získáme

$$c_j \langle \psi_j, \psi_j \rangle_\omega = c_{k+1} \langle \psi_{j+1}, x^{k+1} \rangle_\omega, \quad j = 1, 2, \dots, k$$

a tedy

$$c_j = -\frac{\langle \psi_j, x^{k+1} \rangle_\omega}{\langle \psi_j, \psi_j \rangle_\omega} \quad c_{k+1} = -\frac{\int_a^b x^{k+1} \psi_j d\omega}{\int_a^b \psi_j^2 d\omega}.$$

Koeficient c_{k+1} zvolíme jako libovolnou nenulovou konstantu. □

Uvažujeme monické ortogonální polynomy.

Věta 3.1.2. *Monické ortogonální polynomy $\psi_0, \psi_1, \dots, \psi_{k+1}$ lze počítat tříčlennou rekurencí*

$$\psi_{k+1}(x) = (x - \alpha_k)\psi_k - \beta_k\psi_{k-1}(x) \quad (3.5)$$

$$\psi_0(x) = 1, \quad \psi_1 = x - \alpha_1.$$

Koeficienty α_k a β_k jsou dány

$$\alpha_k = \frac{\int_a^b x \psi_k^2(x) d\omega}{\int_a^b \psi_k^2(x) d\omega}, \quad \beta_k = \frac{\int_a^b \psi_k^2(x) d\omega}{\int_a^b \psi_{k-1}^2(x) d\omega}.$$

Důkaz. Mějme dán polynom $x\psi_k$, což je polynom stupně $k+1$ a můžeme jej tedy zapsat jako lineární kombinaci předchozích polynomů

$$x\psi_k = c_{k+1}\psi_{k+1} + c_k\psi_k + \dots + c_1\psi_1 + c_0\psi_0.$$

Pokud tento vztah přenásobíme polynomem ψ_j , kde $j = 0, 1, \dots, k-2$ a celou rovnost zintegrujeme od a do b podle $d\omega(x)$, dostaneme

$$\langle \psi_k, x\psi_j \rangle_\omega = c_{k+1}\langle \psi_{k+1}, \psi_j \rangle_\omega + c_k\langle \psi_k, \psi_j \rangle_\omega + \dots + c_1\langle \psi_1, \psi_j \rangle_\omega + c_0\langle \psi_0, \psi_j \rangle_\omega. \quad (3.6)$$

Využijeme ortogonality a potom levá strana rovnice bude rovna nule. Stejně tak na pravé straně rovnice využitím ortogonality budou všechny členy nulové, až na člen $c_j\langle \psi_j, \psi_j \rangle_\omega$. Z toho plyne, že koeficient c_j je roven nule pro $j = 0, 1, \dots, k-2$. A je tedy zřejmé, že nenulové jsou pouze koeficienty c_{k+1}, c_k, c_{k-1} . Jednoduchými úpravami vzorce (3.6) získáme vztahy pro koeficienty α_k a β_k . Koeficient c_{k+1} je roven jedné. □

Budeme-li uvažovat polynomy, které jsou vzájemně ortonormální dostaneme následující rekurentní vztah viz [6]

$$\begin{aligned} \sqrt{\beta_{k+1}}\psi_{k+1}(\lambda) &= (\lambda - \alpha_{k+1})\psi_k(\lambda) - \sqrt{\beta_k}\psi_{k-1}(\lambda), \quad k = 0, 1, \dots \\ \psi_{-1}(\lambda) &\equiv 0, \quad \psi_0(\lambda) \equiv \frac{1}{\sqrt{\beta_0}}, \quad \beta_0 = \int_a^b d\omega, \end{aligned}$$

kde koeficienty α_k a β_k jsou stejné jako v případě rekurentního vztahu pro ortogonální polynomy a důkaz by se provedl analogicky.

Tříčlennou rekurenci pro vektor $P_k = [\psi_0(\lambda), \psi_1(\lambda), \dots, \psi_{k-1}(\lambda)]^T$ lze potom pomocí Jacobiho matice

$$\mathbf{T}_k = \begin{bmatrix} \alpha_0 & \sqrt{\beta_1} & 0 & & \\ \sqrt{\beta_1} & \alpha_1 & \sqrt{\beta_2} & \ddots & \\ 0 & \ddots & \ddots & \ddots & \\ & & \sqrt{\beta_{k-1}} & \alpha_{k-1} & \sqrt{\beta_k} \\ & & & \sqrt{\beta_k} & \alpha_k \end{bmatrix} \quad (3.7)$$

zapsat následovně

$$\lambda P_k = \mathbf{T}_k P_k + \sqrt{\beta_k} \psi_k(\lambda) e^k, \quad (3.8)$$

kde e^k je k -tý sloupec jednotkové matice velikosti k .

Je-li μ kořenem polynomu ψ_k platí, že $\psi_k(\mu) = 0$ a tříčlenná rekurence (3.9) pomocí Jacobiho matice nám dává vztah

$$\mu P_k = \mathbf{T}_k P_k, \quad (3.9)$$

protože z definice víme, že $\psi_0 \neq 0$ musí být P_k netriviální vlastní vektor a μ vlastním číslem Jacobiho matice \mathbf{T}_k . Navíc lze ukázat, že kořeny ortonormálního polynomu ψ_k jsou reálné a jednoduché a tedy všechny jeho kořeny jsou vlastními čísly matice \mathbf{T}_k .

3.2 Kvadrurní pravidla

V této části definujeme kvadrurní pravidla v souvislosti s Riemann-Stieltjesovým integrálem viz [3] a ukážeme, jak je možné počítat uzly a váhy jednotlivých kvadrurních pravidel viz [6].

Mějme distribuční funkci ω na uzavřeném intervalu $[a, b]$. Potom n -bodové kvadrurní pravidlo funkce $f(t)$ je určeno následovně

$$\int_a^b f(t) d\omega(t) = \sum_{\nu=1}^n \omega_\nu f(\tau_\nu) + R_n(f), \quad (3.10)$$

kde suma na pravé straně aproximuje integrál na levé straně a R_n je chyba aproximace, jejíž velikost obvykle není známa přesně. τ_ν jsou uzly, které jsou vzájemně různé a ω_ν se nazývají váhy kvadrurního pravidla. Říkáme, že kvadrurní pravidlo (3.10) má stupeň přesnosti h , jestliže

$$R_n(p) = 0 \quad \text{pro všechny polynomy} \quad p \in \mathbb{P}_h,$$

kde \mathbb{P}_h je množina všech polynomů stupně h a zároveň existuje nějaký polynom $q \in \mathbb{P}_{h+1}$, pro který $R_n(q) \neq 0$. Kvadrurní pravidlo (3.10) se stupněm přesnosti $h = n - 1$ se nazývá *interpoláční*. Funkce f může být aproximována interpoláčním polynomem, např. Lagrangeovým polynomem.

Věta 3.2.1. Mějme $k \in \mathbb{N}$ takové, že $0 \leq k \leq n$, kvadrurní pravidlo (3.10) má stupeň přesnosti $h = n - 1 + k$ právě tehdy, když jsou splněny obě následující podmínky

1. Pravidlo (3.10) je interpolační

2. pro polynom $r_n(t) = \prod_{\tau=1}^n (t - \tau_j)$ platí

$$\int_a^b r_n(t)p(t)d\omega(t) = 0 \quad \text{pro všechny} \quad p \in \mathbb{P}_{k-1}.$$

Důkaz můžeme nalézt např. viz[3].

Jestliže funkce ω je kladná a $k = n$, tedy $h = 2n - 1$ kvadrurní pravidlo (3.10) má nejvyšší možný stupeň přesnosti. Pokud bychom položili $k = n + 1$, znamenalo by to, že bychom vyžadovali podmínku ortogonality ω všech polynomů stupně $h \leq n$, což by ale znamenalo, že polynom

$$\prod_{k=1}^n (t - \tau_\nu)$$

je ortogonální sám na sebe a to není možné. Optimální kvadrurní pravidlo s $k = n$ má tedy stupeň přesnosti $h = 2n - 1$ a nazývá se Gaussovo kvadrurní pravidlo. Podmínka ortogonality nám potom ukazuje, že uzly τ_ν jsou kořeny ortogonálního polynomu stupně n s váhovou funkcí ω .

Využijeme-li předchozích poznatků můžeme aproximaci Riemann-Stieltjesova integrálu zapsat pomocí Gaussovy kvadratury následovně

$$\int_a^b f(\lambda)d\omega(\lambda) = \sum_{\nu=1}^n \omega_\nu f(\tau_\nu) + R_n(f), \quad \text{kde} \quad R_n(\mathbb{P}_{2n-1}) = 0. \quad (3.11)$$

$R_n(f)$ je chyba aproximace v n -tém kroku Gaussovy kvadratury, která má následující tvar např. viz [3]

$$R_n(f) = \frac{f^{(2n)}(\nu)}{(2n)!} \int_a^b \left(\prod_{\nu=1}^n (\lambda - \tau_\nu) \right)^2 d\omega(\lambda),$$

kde τ_ν jsou uzly Gaussovy kvadratury, které obecně neznáme. Někdy je ale výhodné použít kvadrurní pravidlo, ve kterém si předepíšeme některý uzel resp. uzly $\tilde{\tau}_j$ a ostatní určíme tak, aby kvadrurní pravidlo mělo co nejvyšší stupeň přesnosti. Pokud volíme jeden uzel $\tilde{\tau}_1$, většinou si předepisujeme $\tilde{\tau}_1 = a$ nebo $\tilde{\tau}_1 = b$ nebo se předepisují oba současně. Pro aproximaci Riemann-Stieltjesova integrálu s předepsanými uzly dostaneme následující tvar

$$\int_a^b f(\lambda)d\omega = \sum_{i=1}^k \omega_i f(\tau_i) + \sum_{j=1}^m \tilde{\omega}_j f(\tilde{\tau}_j) + R_k(f)$$

a chyba aproximace $R_k(f)$ má tvar

$$R_k(f) = \frac{f^{(2k+m)}(\nu)}{(2k+m)!} \int_a^b \prod_{j=1}^m (\lambda - \tilde{\tau}_j) \left[\prod_{i=1}^k (\lambda - \tau_i) \right]^2 d\omega(\lambda).$$

- Pro $m = 0$ dostaneme Gaussovo pravidlo tak, jak jsme si ho definovali výše. Pro chybu $R_n(f)$ je ze vztahu vidět, že pokud je $f^{(2k)} > 0$ pro všechny $\lambda \in [a, b]$, potom $R_k^{(G)}(f) > 0$
- pro $m = 1$ dostaneme Gauss-Radau pravidlo a pro chybu aproximace $R_n(f)$ je ze vztahu vidět, předepíšeme-li $\tilde{\tau}_1 = a$, pak pokud $f^{(2k+1)} < 0$ pro všechny $\lambda \in [a, b]$ a $a \leq \lambda_1$, $R_k^{(GRa)}(f) < 0$. Analogicky, pokud $\tilde{\tau}_1 = b$ a $f^{(2k+1)} > 0$ pro všechny $\lambda \in [a, b]$ a $b \geq \lambda_n$, potom $R_k^{(GRb)}(f) > 0$
- pro $m = 2$ dostaneme Gauss-Lobatto pravidlo a pro chybu aproximace $R_n(f)$ platí, předepíšeme-li si $\tilde{\tau}_1 = a$ a $\tilde{\tau}_2 = b$ a $f^{(2k+2)} > 0$ pro všechny $\lambda \in [a, b]$ a $a \leq \lambda_1$ a $b \geq \lambda_n$, potom $R_k^{(GL)}(f) < 0$.

Chceme-li počítat uzly a váhy Gaussovy kvadratury, jednou z možností je využít předešlých poznatků o ortogonálních polynomech a jejich souvislostí s Jacobiho maticí. Jak již bylo řečeno, kořeny ortogonálních polynomů jsou uzly Gaussovy kvadratury. Mějme posloupnost polynomů $p_0(\lambda), p_1(\lambda), \dots$, které jsou ortonormální s distribuční funkcí ω a tedy platí

$$\int_a^b p_i(\lambda) p_j(\lambda) d\omega(\lambda) = \begin{cases} 1 & \text{pro } i = j, \\ 0 & \text{pro } i \neq j \end{cases}$$

a polynom p_k má stupeň přesnosti $h = k$. Víme, že kořeny polynomu p_k leží v intervalu $[a, b]$. Ukázali jsme, že posloupnost těchto polynomů je určena tříčlennou rekurencí (3.7), kterou můžeme zapsat pomocí Jacobiho matice \mathbf{T}_k (3.9). Z toho vyplývá, že vlastní čísla Jacobiho matice (Ritzova čísla θ_j), která, jak jsme ukázali, jsou také kořeny polynomů p_k , jsou uzly Gaussovy kvadratury. Váhy ω_j lze určit pomocí vlastních vektorů Jacobiho matice.

Chceme-li počítat uzly a váhy Gauss-Radau kvadratury, znamená to, že si předepíšeme jeden uzel τ_1 a ostatní uzly určíme tak, abychom dosáhli co největší algebraické přesnosti. Použijeme Gaussovu kvadraturu, o které víme, že má optimální stupeň přesnosti $2n - 1$ a zkonstruujeme nový polynom ϑ_{k+1} tak, aby námi předepsaný uzel např. $\tau_1 = \mu$ byl kořenem tohoto polynomu a aby tento nový polynom ϑ_{k+1} byl ortogonální ke všem předchozím polynomům ψ_k . Takovýto polynom můžeme vytvořit pomocí tříčlenné rekurence (3.5), což zapíšeme následovně

$$0 = \beta_{k+1} \vartheta_{k+1}(\mu) = (\mu - \tilde{\alpha}_{k+1}) \psi_k(\mu) - \beta_k \psi_{k-1}(\mu), \quad k = 0, 1, \dots \quad (3.12)$$

To znamená, že chceme určit $\tilde{\alpha}_{k+1}$ tak, aby $\tau_1 = \mu$ bylo vlastním číslem modifikované matice $\tilde{\mathbf{T}}_{k+1}$. Vztah (3.12) nám dává následující podmínku

$$\tilde{\alpha}_{k+1} = \mu - \beta_k \frac{\psi_{k-1}(\mu)}{\psi_k(\mu)}.$$

Označíme si $\delta_k^{(\mu)} = -\beta_k \frac{\psi_{k-1}(\mu)}{\psi_k(\mu)}$. Dá se ukázat viz např. [6], že $\delta_k^{(\mu)}$ určíme jako poslední složku vektoru, který získáme jako řešení soustavy s posunutou Jacobiho maticí $(\mathbf{T}_k - \mu\mathbf{I})\delta^{(\mu)} = \beta_k^2 e^k$, což můžeme maticově zapsat následovně

$$\begin{bmatrix} \alpha_1 - \mu & \beta_1 & 0 & & \\ \beta_1 & \ddots & \ddots & & \\ 0 & \ddots & \ddots & \beta_{k-1} & \\ & & \beta_{k-1} & \alpha_k - \mu & \\ & & & & \end{bmatrix} \begin{bmatrix} \delta_1^{(\mu)} \\ \vdots \\ \delta_{k-1}^{(\mu)} \\ \delta_k^{(\mu)} \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \beta_k^2 \end{bmatrix}.$$

Z Choleského rozkladu posuté Jacobiho matice $(\mathbf{T}_k - \mu\mathbf{I})$ se již dá snadno ukázat, že pro $\tilde{\alpha}_{k+1}$ platí

$$\tilde{\alpha}_{k+1} = \mu + \delta_k^{(\mu)}, \quad \delta_k^{(\mu)} = \frac{\beta_k^2}{\alpha_k - \mu}. \quad (3.13)$$

Jacobiho matice $\tilde{\mathbf{T}}_{k+1}$ odpovídající Gauss-Radau kvadratuře s předepsaným uzlem $\tau_1 = \mu$ má potom následující tvar

$$\tilde{\mathbf{T}}_{k+1}^{(\mu)} = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \ddots & & \\ & \ddots & \ddots & \beta_{k-1} & \\ & & \beta_{k-1} & \alpha_k & \beta_k \\ & & & \beta_k & \tilde{\alpha}_{k+1}^{(\mu)} \end{bmatrix}. \quad (3.14)$$

Chceme-li použít Gauss-Lobatto pravidlo, předepíšeme si $\tau_1 = \mu$ a $\tau_2 = \eta$ a budeme postupovat stejně jako v předchozím odstavci. Budeme chtít, aby platilo

$$\begin{aligned} 0 &= \beta_{k+1} \vartheta_{k+1}(\mu) = (\mu - \tilde{\alpha}_{k+1})\psi_k(\mu) - \tilde{\beta}_k \psi_{k-1}(\mu), \quad k = 0, 1, \dots, \\ 0 &= \beta_{k+1} \vartheta_{k+1}(\eta) = (\eta - \tilde{\alpha}_{k+1})\psi_k(\eta) - \tilde{\beta}_k \psi_{k-1}(\eta), \quad k = 0, 1, \dots, \end{aligned}$$

tím získáme soustavu dvou rovnic pro neznámé $\tilde{\alpha}_{k+1}^{(\mu, \eta)}$ a $\tilde{\beta}_k^{(\mu, \eta)}$

$$\tilde{\alpha}_{k+1}^{(\mu, \eta)} = \mu + \left(\tilde{\beta}_k^{(\mu, \eta)} \right)^2 \delta_k^{(\mu)}, \quad \tilde{\alpha}_{k+1}^{(\mu, \eta)} = \eta + \left(\tilde{\beta}_k^{(\mu, \eta)} \right)^2 \delta_k^{(\eta)}, \quad (3.15)$$

kde $\delta_k^{(\mu)}$ a $\delta_k^{(\eta)}$ opět určíme jako poslední složku řešení systému rovnic s posunutou Jacobiho maticí

$$\begin{aligned} (\mathbf{T}_k - \mu\mathbf{I})\delta^{(\mu)} &= \tilde{\beta}_k^2 e^k \quad \text{a} \\ (\mathbf{T}_k - \eta\mathbf{I})\delta^{(\eta)} &= \tilde{\beta}_k^2 e^k, \quad \text{kde} \\ \delta^{(\mu)} &= [\delta_1^{(\mu)}, \dots, \delta_k^{(\mu)}] \quad \text{a} \quad \delta^{(\eta)} = [\delta_1^{(\eta)}, \dots, \delta_k^{(\eta)}]. \end{aligned}$$

Kapitola 4

Odhady normy chyby

V této kapitole ukážeme souvislosti mezi CG, Gaussovou kvadraturou, Jacobiho maticí a ortogonálními polynomy viz např. [11] a jak díky těmto souvislostem je možné odhadovat \mathbf{A} -normu chyby $\|x - x_k\|_{\mathbf{A}}$ popř. euklidovskou normu chyby $\|x - x_k\|_2$ v k -tém kroku CG.

4.1 Souvislosti mezi CG a Gaussovou kvadraturou

Mějme vlastní čísla matice \mathbf{A} uspořádaná vzestupně tak, že $a \leq \lambda_1 < \lambda_2 < \dots < \lambda_n \leq b$. Uvažujme distribuční funkci $\omega(\lambda)$, pro kterou platí

$$\omega(\lambda) = \begin{cases} 0 & \text{pro } \lambda < \lambda_1, \\ \sum_{i=1}^k \omega_i & \text{pro } \lambda_k \leq \lambda < \lambda_{k+1}, \quad \leq k \leq n, \\ 1 & \text{pro } \lambda_n \leq \lambda, \end{cases}$$

potom Riemann-Stieltjesův integrál má tvar

$$\int_a^b f(\lambda) d\omega = \sum_{i=1}^n \omega_i f(\lambda_i).$$

V kapitole o metodě sdružených gradientů jsme uvedli, že $x_k = x_0 + q(\mathbf{A})r_0$, kde $q_k(\mathbf{A})$ je polynom stupně k , tedy reziduum r_k můžeme zapsat jako polynom matice \mathbf{A} a platí $r_k = q_k(\mathbf{A})r_0$. Podobně je možné odvodit následující vztah pro vektory v_{j+1} vznikající v Lanczosově algoritmu

$$v_{j+1} = \psi_j(\mathbf{A})v_1 \cdot \frac{1}{\beta_1\beta_2, \dots, \beta_{j+1}},$$

kde ψ_j je polynom stupně j . Užitím ortogonalitly můžeme odvodit

$$\|\psi_j(\mathbf{A})v_1\| = \min_{\psi \in \mathcal{M}_j} \|\psi(\mathbf{A})v_1\| = \min_{\psi \in \mathcal{M}_j} \left(\sum_{i=1}^n (v_1, u_i)^2 \psi^2(\lambda_i) \right)^{\frac{1}{2}}, \quad (4.1)$$

kde \mathcal{M}_j označuje množinu monických polynomů stupně j . Uvažujeme-li CG nebo Lanczosův algoritmus, získáváme posloupnost monických polynomů $1, \psi_1, \psi_2, \dots, \psi_{j+1}$, které jsou ortogonální vzhledem ke skalárnímu součinu (3.4), jak jsme si ho definovali v kapitole o ortogonálních polynomech

$$\langle f(\lambda)g(\lambda) \rangle_\omega = \sum_{i=1}^n \omega_i f(\lambda_i)g(\lambda_i),$$

kde váhy ω_i jsou určeny takto

$$\omega_i = (v_1, u_i)^2, \quad \sum_{i=1}^n \omega_i = 1.$$

Potom vztah (4.1) můžeme přepsat ve smyslu Riemann-Stieltjesova integrálu

$$\psi_j = \arg \min_{\psi \in \mathcal{M}_j} \left(\int_a^b \psi^2(\lambda) d\omega \right)^{\frac{1}{2}}, \quad j = 1, 2, \dots, n.$$

Jak již bylo řečeno, iterace CG resp. Lanczosův algoritmus se startovacím vektorem $\|r_0\|v_1$ resp. v_1 určují symetrickou třídiagonální Jacobiho matici \mathbf{T}_j , jejíž vlastní čísla (Ritzova čísla θ_j) aproximují vlastní čísla matice \mathbf{A} . Rozklad Jacobiho matice můžeme psát

$$\mathbf{T}_j = \mathbf{S}_j \mathbf{\Theta}_j \mathbf{S}_j^T, \quad \mathbf{S}_j^T \mathbf{S}_j = \mathbf{S}_j \mathbf{S}_j^T = \mathbf{I},$$

kde $\mathbf{\Theta}_j = \text{diag}(\theta_1^{(j)}, \dots, \theta_j^{(j)})$, $\mathbf{S}_j = [s_1^{(j)}, \dots, s_j^{(j)}]$. Analogicky jako jsme zkonstruovali váhy ω_i a distribuční funkci $\omega(\lambda)$ pro matici \mathbf{A} , zkonstruujeme váhy $\omega_i^{(j)}$ a distribuční funkci $\omega^{(j)}$ pro Jacobiho matici \mathbf{T}_j

$$\omega_i^{(j)} = (e_1, s_i^{(j)})^2, \quad \sum_{i=1}^j \omega_i^{(j)} = 1,$$

a pro $a < \theta_1^{(j)}, \dots, \theta_j^{(j)} < b$, je prvních j vektorů z posloupnosti $\{1, \psi_1, \psi_2, \dots, \psi_n\}$ určeno podmínkou

$$\psi_j = \min_{\psi \in \mathcal{M}_j} \left(\int_a^b \psi^2(\lambda) d\omega^{(j)}(\lambda) \right)^{1/2}.$$

Potom j -tou aproximaci Riemann-Stieltjesova integrálu pomocí Gaussovy kvadratury zapsanou takto

$$\int_a^b f(\lambda) d\omega^{(j)}(\lambda) = \sum_{i=1}^k \omega_i^{(j)} f(\theta_i^{(j)}), \quad (4.2)$$

nazýváme j -bodovým Gaussovým kvadraturním pravidlem. Pro polynomy, které jsou stupně $2n - 1$, dostáváme přesnou hodnotu Riemann-Stieltjesova integrálu. Necht' tedy máme matici \mathbf{A} a počáteční vektor rezidua r_0 , potom Riemann-Stieltjesův integrál a jeho aproximace pomocí Gaussovy kvadratury jsou pro $j = 1, 2, \dots, n$ jednoznačně určeny. Pak distribuční funkce $\omega^{(j)}$ jednoznačně určuje matici \mathbf{T}_j a ta společně se vztahy (2.13) určuje CG aproximace x_j . Budeme-li uvažovat funkci $f(\lambda) = \lambda^{-1}$ resp. $f(\lambda) = \lambda^{-2}$, můžeme vztah (2.4) psát ve smyslu Riemann-Stieltjesova integrálu a dostaneme

$$\begin{aligned}\|x - x_0\|_{\mathbf{A}}^2 &= \|r_0\|^2 \sum_{i=1}^n \frac{\omega_i}{\lambda_i} = \|r_0\|^2 \int_a^b \lambda^{-1} d\omega(\lambda) \\ \|x - x_0\|^2 &= \|r_0\|^2 \sum_{i=1}^n \frac{\omega_i}{\lambda_i^2} = \|r_0\|^2 \int_a^b \lambda^{-2} d\omega(\lambda)\end{aligned}\quad (4.3)$$

s použitím vztahů

$$\begin{aligned}\|x - x_0\|_{\mathbf{A}}^2 &= (r_0, \mathbf{A}^{-1}r_0) = \|r_0\|^2 (e_1, \mathbf{T}_n^{-1}e_1) \equiv \|r_0\|^2 (\mathbf{T}_n^{-1})_{1,1} \\ \|x - x_0\|^2 &= (r_0, \mathbf{A}^{-2}r_0) = \|r_0\|^2 (e_1, \mathbf{T}_n^{-2}e_1) \equiv \|r_0\|^2 (\mathbf{T}_n^{-2})_{1,1}\end{aligned}\quad (4.4)$$

můžeme Riemann-Stieltjesův integrál pro funkci $f(\lambda) = \lambda^{-1}$ resp. $f(\lambda) = \lambda^{-2}$ zapsat následovně

$$\int_a^b \lambda^{-1} d\omega(\lambda) = (\mathbf{T}_n^{-1})_{1,1}, \quad \int_a^b \lambda^{-2} d\omega(\lambda) = (\mathbf{T}_n^{-2})_{1,1}.$$

Analogicky můžeme psát pro j -tý krok

$$\int_a^b \lambda^{-1} d\omega^{(j)}(\lambda) = (\mathbf{T}_j^{-1})_{1,1}, \quad \int_a^b \lambda^{-2} d\omega^{(j)}(\lambda) = (\mathbf{T}_j^{-2})_{1,1}.$$

Je zřejmé, že použijeme-li $f(\lambda) = \lambda^{-2}$, můžeme odvozovat i vztahy pro euklidovskou normu $\|x - x_k\|_2$, ale to není cílem této práce. Další podrobnosti je možné najít např. viz [9] a dále se budeme již věnovat pouze \mathbf{A} -normě chyby.

Označíme-li $R_k(f)$ chybu k -té aproximace funkce f Gaussovou kvadraturou, můžeme Riemann-Stieltjesův integrál zapsat takto

$$\int_a^b f(\lambda) d\omega(\lambda) = \int_a^b f(\lambda) d\omega^{(k)}(\lambda) + R_k(f),$$

což můžeme přepsat pomocí Jacobiho matice \mathbf{T}

$$(\mathbf{T}_n^{-1})_{1,1} = (\mathbf{T}_k^{-1})_{1,1} + R_k^{(G)}(\lambda^{-1}).$$

Přenásobíme rovnici $\|r_0\|^2$ a užitím vztahu (4.4) dostaneme

$$\|x - x_0\|_{\mathbf{A}}^2 = \|r_0\|^2 (\mathbf{T}_n^{-1})_{1,1} = \|r_0\|^2 (\mathbf{T}_k^{-1})_{1,1} + \|r_0\|^2 R_k^{(G)}(\lambda^{-1}).\quad (4.5)$$

Chybu aproximace v k -tém kroku pro funkci $f(\lambda) = \frac{1}{\lambda}$ viz [7] lze zapsat ve tvaru

$$R_k^{(G)}(\lambda^{-1}) = \frac{\|x - x_k\|_{\mathbf{A}}^2}{\|r_0\|^2}.$$

Dosadíme-li chybu aproximace do vztahu (4.5) dostaneme

$$\|x - x_0\|_{\mathbf{A}}^2 = \|r_0\|^2 (\mathbf{T}_k^{-1})_{1,1} + \|x - x_k\|_{\mathbf{A}}^2. \quad (4.6)$$

Užitím vztahů CG iterací viz [11] lze ukázat, že rovnici (4.6) můžeme přepsat

$$\|x - x_0\|_{\mathbf{A}}^2 = \sum_{j=0}^{k-1} \gamma_j \|r_j\|^2 + \|x - x_k\|_{\mathbf{A}}^2.$$

Z tohoto vztahu a ze vztahu (4.6) můžeme vyjádřit rovnici pro $(\mathbf{T}_k^{-1})_{1,1}$

$$(\mathbf{T}_k^{-1})_{1,1} = \frac{1}{\|r_0\|^2} \sum_{j=0}^{k-1} \gamma_j \|r_j\|^2. \quad (4.7)$$

Pro odhadování \mathbf{A} -normy chyby v CG použijeme Gaussovo kvadrurní pravidlo v k -tém kroku

$$\|x - x_0\|_{\mathbf{A}}^2 = \|r_0\|^2 (\mathbf{T}_k^{-1})_{1,1} + \|x - x_k\|_{\mathbf{A}}^2. \quad (4.8)$$

Stejně tak můžeme zapsat odhad v $(k+d)$ -tém kroku pro $d > 0$

$$\|x - x_0\|_{\mathbf{A}}^2 = \|r_0\|^2 (\hat{\mathbf{T}}_{k+d}^{-1})_{1,1} + R_{k+d}(\lambda^{-1}), \quad (4.9)$$

kde $\hat{\mathbf{T}}_{k+d}$ značí matici vypočtenou v kroku $k+d$ pomocí Gaussovy, Gauss-Radau nebo Gauss-Lobatto kvadratury. Odečtením vztahů (4.11) a (4.8) dostaneme

$$\|x - x_k\|_{\mathbf{A}}^2 = \|r_0\|^2 \left((\hat{\mathbf{T}}_{k+d}^{-1})_{1,1} - (\mathbf{T}_k^{-1})_{1,1} \right) - R_{k+d}(\lambda^{-1}). \quad (4.10)$$

Označíme si

$$\hat{Q}_{k,d} = \|r_0\|^2 \left((\hat{\mathbf{T}}_{k+d}^{-1})_{1,1} - (\mathbf{T}_k^{-1})_{1,1} \right). \quad (4.11)$$

V části o kvadrurních pravidlech jsme uvedli, jaké podmínky musí být splněny, abychom mohli říct, zda-li je $R_{k+d} > 0$ nebo $R_{k+d} < 0$.

- Pro Gaussovu kvadraturu je pro funkci $f = \lambda^{-1}$ derivace $f^{(2k)} > 0$. Proto $R_{k+d} > 0$ a dostaneme dolní odhady \mathbf{A} -normy chyby $\|x - x_k\|_{\mathbf{A}}$.
- Pro Gauss-Radauⁿ předpokládáme $\lambda_n \leq \eta$ a pro funkci $f = \lambda^{-1}$ je derivace $f^{(2k+1)} > 0$. Proto $R_{k+d} > 0$ a dostaneme dolní odhady \mathbf{A} -normy chyby $\|x - x_k\|_{\mathbf{A}}$.
- Pro Gauss-Radau^μ předpokládáme $\mu \leq \lambda_1$ a pro funkci $f = \lambda^{-1}$ je derivace $f^{(2k+1)} < 0$. Proto $R_{k+d} < 0$ a dostaneme horní odhady \mathbf{A} -normy chyby $\|x - x_k\|_{\mathbf{A}}$.

- Pro Gauss-Lobatto předpokládáme ($\mu \leq \lambda_1 \wedge \lambda_n \leq \eta$) a pro funkci $f = \lambda^{-1}$ je derivace $f^{(2k+2)} > 0$. Proto $R_{k+d} < 0$ a dostaneme horní odhady \mathbf{A} -normy chyby $\|x - x_k\|_{\mathbf{A}}$.

Výpočet \hat{Q}_{k+d} nám dá horní resp. dolní odhad \mathbf{A} -normy chyby podle toho jaké kvadraturní pravidlo použijeme. Z hlediska počítačové aritmetiky není vhodné počítat prvky $(\mathbf{T}_k^{-1})_{1,1}$ a $(\hat{\mathbf{T}}_{k+d}^{-1})_{1,1}$ a potom je odečíst, protože můžeme ztratit přesnost výsledku. Místo odečítání těchto prvků si rozdíl ještě upravíme

$$(\hat{\mathbf{T}}_{k+d}^{-1})_{1,1} - (\mathbf{T}_k^{-1})_{1,1} = (\hat{\mathbf{T}}_{k+d}^{-1})_{1,1} - (\mathbf{T}_{k+d-1}^{-1})_{1,1} + \sum_{j=k}^{k+d-2} [(\mathbf{T}_{j+1}^{-1})_{1,1} - (\mathbf{T}_j^{-1})_{1,1}].$$

Z výrazu (4.7) dostaneme, že

$$\|r_0\|^2 [(\mathbf{T}_{j+1}^{-1})_{1,1} - (\mathbf{T}_j^{-1})_{1,1}] = \gamma_j \|r_j\|^2.$$

Potom vztah pro $\hat{Q}_{k,d}$ můžeme psát následovně

$$\hat{Q}_{k,d} = \|r_0\|^2 \left((\hat{\mathbf{T}}_{k+d}^{-1})_{1,1} - (\mathbf{T}_{k+d-1}^{-1})_{1,1} \right) + \sum_{j=k}^{k+d-2} \gamma_j \|r_j\|^2 \quad (4.12)$$

a \mathbf{A} -normu chyby v k -tém kroku CG můžeme zapsat

$$\|x - x_k\|_{\mathbf{A}}^2 = \|r_0\|^2 \left((\hat{\mathbf{T}}_{k+d}^{-1})_{1,1} - (\mathbf{T}_{k+d-1}^{-1})_{1,1} \right) + \sum_{j=k}^{k+d-2} \gamma_j \|r_j\|^2 - R_{k+d}(\lambda^{-1}). \quad (4.13)$$

Koeficienty γ_j a rezidua r_j získáváme v průběhu metody sdružených gradientů tzn., že problém odhadování \mathbf{A} -normy chyby se nám redukuje na problém, jak vypočítat rozdíl dvou prvků Jacobiho matic

$$\left((\hat{\mathbf{T}}_{k+d}^{-1})_{1,1} - (\mathbf{T}_{k+d-1}^{-1})_{1,1} \right). \quad (4.14)$$

V části o metodě sdružených gradientů a jejích souvislostech s Lanczosovým algoritmem jsme ukázali, že v průběhu CG získáváme koeficienty γ_j a δ_j . Tyto koeficienty nám neurčují přímo Jacobiho matici \mathbf{T}_k , ale její \mathbf{LDL}^T faktorizaci. Protože se chceme vyhnout explicitnímu vyjádření Jacobiho matice, odvodíme si, jaký tvar má \mathbf{LDL}^T faktorizace Jacobiho třídiagonální matice \mathbf{T}_k a faktorizace modifikované Jacobiho matice $\hat{\mathbf{T}}_{k+1}$ odpovídající Gauss-Radau a Gauss-Lobatto kvadratuře tak, jak jsme si je uvedli ve vztazích (3.14) resp. (3.16). Těchto \mathbf{LDL}^T rozkladů poté využijeme k počítání rozdílu (4.14), tj. budeme počítat (4.14) bez explicitních znalostí prvků modifikované Jacobiho matice a tedy ukážeme jak odhadovat \mathbf{A} -normu chyby.

4.1.1 Faktorizace třídiagonální matice

Mějme symetrickou třídiagonální matici \mathbf{T}_k s prvky na hlavní diagonále α_j a s prvky $\beta_j \neq 0$ na vedlejší diagonále.

$$\mathbf{T}_k = \begin{bmatrix} \alpha_1 & \beta_1 & & & & \\ \beta_1 & \alpha_2 & \ddots & & & \\ & \ddots & \ddots & \beta_{k-2} & & \\ & & \beta_{k-2} & \alpha_{k-1} & \beta_{k-1} & \\ & & & \beta_{k-1} & \alpha_k & \end{bmatrix}. \quad (4.15)$$

Potom faktorizaci LDL^T matice $\mathbf{T}_k = \mathbf{L}_k \mathbf{D}_k \mathbf{L}_k^T$ zapíšeme takto

$$\begin{bmatrix} 1 & 0 & & & & \\ l_1 & \ddots & \ddots & & & \\ & \ddots & & & & \\ & & & 0 & & \\ & & & l_{k-1} & 1 & \end{bmatrix} \begin{bmatrix} d_1 & 0 & & & & \\ 0 & d_2 & \ddots & & & \\ & \ddots & \ddots & & & \\ & & & & & \\ & & & & & d_k \end{bmatrix} \begin{bmatrix} 1 & l_1 & & & & \\ 0 & \ddots & \ddots & & & \\ & \ddots & & & & \\ & & & & & \\ & & & & & l_{k-1} \\ & & & & 0 & 1 \end{bmatrix} \quad (4.16)$$

Jednotlivé prvky můžeme počítat následujícími rekurentními vztahy viz [10]

$$d_1 = \alpha_1, \quad l_j = \frac{\beta_j}{d_j}, \quad d_{j+1} = \alpha_{j+1} - \beta_j l_j, \quad j = 1, \dots, k-1.$$

4.1.2 Faktorizace posunuté třídiagonální matice

Použijeme-li Gauss-Radau nebo Gauss-Lobatto kvadraturu, předepíšeme si jeden resp. dva uzly, tedy jedno resp. dvě vlastní čísla Jacobiho matice \mathbf{T}_{k+1} . Ukázali jsme, že hledané prvky $\tilde{\alpha}_{k+1}^{(\mu)}$ Jacobiho matice $\tilde{\mathbf{T}}_{k+1}$ ve vztahu (3.13) a hledané prvky $\tilde{\alpha}_{k+1}^{(\mu, \eta)}$ a $\left(\tilde{\beta}_k^{(\mu, \eta)}\right)^2$ ve vztahu (3.15) počítáme pomocí řešení soustavy s posunutou Jacobiho maticí $\mathbf{T}_k - \mu \mathbf{I}$ a $\mathbf{T}_k - \eta \mathbf{I}$. Tedy i LDL^T faktorizaci modifikované Jacobiho matice $\tilde{\mathbf{T}}_{k+1}$ budeme počítat pomocí LDL^T faktorizace posunuté Jacobiho matice.

LDL^T faktorizaci matice $\mathbf{T}_k - \mu \mathbf{I}$ zapíšeme takto

$$\mathbf{T}_k - \mu \mathbf{I} = \bar{\mathbf{L}}_k^{(\mu)} \bar{\mathbf{D}}_k^{(\mu)} (\bar{\mathbf{L}}_k^{(\mu)})^T.$$

Jednotlivé prvky lze počítat následujícími rekurentními vztahy

$$\bar{d}_1 = \alpha_1 - \mu, \quad \bar{l}_j = \frac{\beta_j}{\bar{d}_j^{(\mu)}}, \quad \bar{d}_{j+1} = \alpha_{j+1} - \mu - \beta_j \bar{l}_j^{(\mu)}, \quad j = 1, \dots, k-1.$$

$\mathbf{L}_{k+1} \mathbf{D}_{k+1} \mathbf{L}_{k+1}^T$ faktorizace modifikované matice $\tilde{\mathbf{T}}_{k+1}$ s předepsaným jedním vlastním číslem, která vznikne rozšířením o jeden sloupec a jeden řádek prvky $\tilde{\alpha}_{k+1}$ a β_k pomocí

posunutá matice $\mathbf{T}_k - \mu\mathbf{I}$ má tvar

$$\tilde{\mathbf{T}}_{k+1}^{(\mu)} = \mathbf{L}_{k+1} \begin{bmatrix} d_1 & & & & \\ & \ddots & & & \\ & & \ddots & & \\ & & & d_k & \\ & & & & \tilde{d}_{k+1}^{(\mu)} \end{bmatrix} \mathbf{L}_{k+1}^T, \quad (4.17)$$

kde

$$\tilde{d}_{k+1}^{(\mu)} = \tilde{\alpha}_{k+1}^\mu - \beta_k l_k = \mu + \frac{\beta_k^2}{\tilde{d}_k^{(\mu)}} - d_k l_k^2.$$

My ale chceme využít pouze prvků \mathbf{LDL}^T rozkladu modifikované Jacobiho matice $\tilde{\mathbf{T}}_k$ a pomocí nich počítat prvky $\tilde{d}_{k+1}^{(\mu)}$.

Lemma 4.1.1. *Mějme μ rozdíl od nějakého vlastního čísla matice \mathbf{T}_k a odpovídající \mathbf{LDL}^T faktorizace matice \mathbf{T}_{k+1} a matice posunutá $\mathbf{T}_{k+1} - \mu\mathbf{I}$ zapíšeme následujícími vztahy*

$$\mathbf{T}_{k+1} = \mathbf{L}_{k+1} \mathbf{D}_{k+1} \mathbf{L}_{k+1}^T, \quad \mathbf{T}_{k+1} - \mu\mathbf{I} = \bar{\mathbf{L}}_{k+1} \bar{\mathbf{D}}_{k+1} \bar{\mathbf{L}}_{k+1}^T.$$

Mějme $\tilde{\mathbf{T}}_{k+1}$ modifikovanou matici tak, že předepíšeme μ jako vlastní číslo matice $\tilde{\mathbf{T}}_{k+1}$. Potom platí

$$\tilde{d}_{k+1}^{(\mu)} = d_{k+1} - \tilde{d}_{k+1}^{(\mu)}.$$

Důkaz můžeme nalézt viz [10].

V tomto článku[10] je také dokázána následující rekurentní formule pro prvky $\tilde{d}_{k+1}^{(\mu)}$, které vůbec nemusíme počítat pomocí prvků \mathbf{LDL}^T rozkladu posunuté matice $\mathbf{T}_{k+1} - \mu\mathbf{I}$, ale pouze z prvků \mathbf{LDL}^T rozkladu Jacobiho matice $\tilde{\mathbf{T}}_k^{(\mu)}$

$$\tilde{d}_1 = \mu, \quad \tilde{d}_{k+1}^{(\mu)} = \mu + l_k^2 \frac{d_k \tilde{d}_k}{d_k - \tilde{d}_k^{(\mu)}} \quad \text{pro } k \leq 1. \quad (4.18)$$

$\mathbf{L}_{k+1} \mathbf{D}_{k+1} \mathbf{L}_{k+1}^T$ faktorizace modifikované matice $\tilde{\mathbf{T}}_{k+1}$ s dvěma předepsanými vlastními čísly μ a η , která vznikne rozšířením o jeden sloupec a jeden řádek prvky $\tilde{\alpha}_{k+1}$ a $\tilde{\beta}_k$ pomocí posunutých matic $\mathbf{T}_k - \mu\mathbf{I}$ a $\mathbf{T}_k - \eta\mathbf{I}$ má tvar

$$\begin{bmatrix} 1 & & & & \\ l_1 & 1 & & & \\ 0 & \ddots & \ddots & & \\ & \ddots & l_k & 1 & \\ & & & \tilde{l}_k^{(\mu,\eta)} & 1 \end{bmatrix} \begin{bmatrix} d_1 & 0 & & & \\ 0 & \ddots & \ddots & & \\ & \ddots & \ddots & d_k & \\ & & & & \tilde{d}_{k+1}^{(\mu,\eta)} \end{bmatrix} \begin{bmatrix} 1 & l_1 & & & \\ 0 & \ddots & \ddots & & \\ & \ddots & & l_{k-1} & \\ & & & 1 & \tilde{l}_k^{(\mu,\eta)} \\ & & & & 1 \end{bmatrix}. \quad (4.19)$$

Prvky \tilde{d}_{k+1} a \tilde{l}_k lze počítat následujícími vztahy viz [10] opět bez znalosti prvků posunutých matic $\mathbf{T}_k - \mu\mathbf{I}$ a $\mathbf{T}_k - \eta\mathbf{I}$, ale pouze z prvků LDL^T rozkladu matice $\tilde{\mathbf{T}}_k^{(\mu,\eta)}$,

$$\begin{aligned} \left(\tilde{l}_k^{(\mu,\eta)}\right)^2 &= \frac{(d_k - \tilde{d}_k^{(\mu)})(d_k - \tilde{d}_k^{(\eta)})(\eta - \mu)}{2} (\tilde{d}_k^{(\mu)} - \tilde{d}_k^{(\eta)}) d_k^2, \\ \tilde{d}_{k+1}^{(\mu,\eta)} &= \frac{d_k(\eta - \mu) + \mu\tilde{d}_k^{(\mu)} - \eta\tilde{d}_k^{(\eta)}}{\tilde{d}_k^{(\mu)} - \tilde{d}_k^{(\eta)}} - d_k \left(\tilde{l}_k^{(\mu,\eta)}\right)^2. \end{aligned} \quad (4.20)$$

Z předchozích uvedených vztahů vyplývá, že když máme \tilde{l}_{k-1} a \tilde{d}_k prvky LDL^T faktorizace modifikované Jacobiho matice $\tilde{\mathbf{T}}_k$ pomocí Gaussova, Gauss-Radau nebo Gauss-Lobatto pravidla, umíme spočítat prvky \tilde{l}_k a \tilde{d}_{k+1} pouze pomocí předchozích prvků LDL^T faktorizace bez implicitního vyjádření prvků matice $\tilde{\mathbf{T}}_k$.

4.1.3 Jak počítat různé kvadratury pomocí modifikované Jacobiho matice

Uvedli jsme, že pro počítání odhadů \mathbf{A} -normy chyby pomocí různých kvadraturních pravidel budeme potřebovat počítat rozdíl mezi prvkem $(1, 1)$ modifikované Jacobiho matice $\hat{\mathbf{T}}_{k+1}^{-1}$ a Jacobiho matice \mathbf{T}_k^{-1} . Rozdíl těchto dvou prvků lze zapsat následujícím vztahem viz [6]

$$\left(\hat{\mathbf{T}}_{k+1}^{-1}\right)_{1,1} - \left(\mathbf{T}_k^{-1}\right)_{1,1} = \frac{\hat{l}_k^2}{\hat{d}_{k+1}} (l_1^2 \cdots l_{k-1}^2), \quad (4.21)$$

kde \hat{l}_k a \hat{d}_{k+1} jsou prvky LDL^T rozkladu modifikované Jacobiho matice $\tilde{\mathbf{T}}_{k+1}$ pomocí Gauss-Radau a Gauss-Lobatto kvadratury, které umíme počítat pomocí odvozených formulí 4.18 a 4.20.

4.2 Algoritmická realizace

Navážeme na výsledky předchozí části, kde jsme ukázali, jak počítat prvky l_k a d_k v LDL^T faktorizaci matice \mathbf{T}_k a modifikované matice $\tilde{\mathbf{T}}_{k+1}$. Využijeme algebraických vztahů s koeficienty γ_j a δ_j počítaných metodou sdružených gradientů a použijeme je k počítání horních a dolních odhadů \mathbf{A} -normy chyby. Koeficienty l_k a d_k ve vztahu (4.16) odpovídající koeficientům γ_j a δ_j ve vztahu (2.19) si můžeme definovat takto

$$l_k^2 = \delta_k, \quad d_k = \gamma_{k-1}^{-1}.$$

Pro předepsané uzly μ a η dostaneme různě modifikované \tilde{d}_k , podle toho jaké kvadraturní pravidlo použijeme. Následující tvary koeficientů γ_k pro Gauss-Radau^(μ), Gauss-Radau^(η) a Gauss-Lobatto kvadraturu si definujeme takto

$$\tilde{\gamma}_k^{(\mu)} \equiv \left(\tilde{d}_{k+1}^{(\mu)}\right)^{-1}, \quad \tilde{\gamma}_k^{(\eta)} \equiv \left(\tilde{d}_{k+1}^{(\eta)}\right)^{-1}, \quad \tilde{\gamma}_k^{(\mu,\eta)} \equiv \left(\tilde{l}_k^{(\mu,\eta)}\right)^2 \left(\tilde{d}_{k+1}^{(\mu,\eta)}\right)^{-1}.$$

Užitím vztahů pro \tilde{d}_{k+1} (4.18) a (4.20) můžeme rozepsat vztahy pro $\tilde{\gamma}_k$ následovně

$$\begin{aligned}\tilde{\gamma}_0^{(\mu)} &= \frac{1}{\mu}, & \tilde{\gamma}_k^{(\mu)} &= \frac{\tilde{\gamma}_{k-1}^{(\mu)} - \gamma_{k-1}}{\mu \left(\tilde{\gamma}_{k-1}^{(\mu)} - \gamma_{k-1} \right) + \delta_k}, \\ \tilde{\gamma}_0^{(\eta)} &= \frac{1}{\eta}, & \tilde{\gamma}_k^{(\eta)} &= \frac{\tilde{\gamma}_{k-1}^{(\eta)} - \gamma_{k-1}}{\eta \left(\tilde{\gamma}_{k-1}^{(\eta)} - \gamma_{k-1} \right) + \delta_k},\end{aligned}\tag{4.22}$$

$$\tilde{\gamma}_k^{(\mu,\eta)} = \frac{\left(\tilde{\gamma}_{k-1}^{(\mu)} - \gamma_{k-1} \right) \left(\tilde{\gamma}_{k-1}^{(\eta)} - \gamma_{k-1} \right) (\eta - \mu)}{\eta \left(\tilde{\gamma}_{k-1}^{(\eta)} - \gamma_{k-1} \right) - \mu \left(\tilde{\gamma}_{k-1}^{(\mu)} - \gamma_{k-1} \right)}.\tag{4.23}$$

Použijeme-li odvozeného vztahu (4.21) pro rozdíl mezi prvkem $(1, 1)$ matice $\hat{\mathbf{T}}_{k+1}^{-1}$ a matice $\hat{\mathbf{T}}_{k+1}^{-1}$ a toho, že platí

$$\|r_0\|^2 (l_1 \dots l_{k-1})^2 = \|r_0\|^2 \delta_1 \dots \delta_{k-1} = \|r_0\|^2 \frac{\|r_1\|^2}{\|r_0\|^2} \dots \frac{\|r_{k-1}\|^2}{\|r_{k-2}\|^2} = \|r_{k-1}\|^2,$$

můžeme vztah (4.11) přepsat takto

$$\|r_0\|^2 \left(\left[\left(\tilde{\mathbf{T}}_{k+1}^{(\mu)} \right)^{-1} \right]_{1,1} - (\mathbf{T}_k^{-1})_{1,1} \right) = \frac{\hat{l}_k^2}{\hat{d}_{k+1}} \|r_{k-1}\|^2 = \frac{\|r_k\|^2}{\hat{d}_{k+1}},$$

kde \hat{d}_{k+1} a \hat{l}_k nám označuje modifikované \tilde{d}_{k+1} a \tilde{l}_k v závislosti na tom, jestli jsme použili Gaussovu, Gauss-Radau^(μ), Gauss-Radau^(η) nebo Gauss-Lobatto kvadraturu. Označíme si

$$\begin{aligned}g_k &\equiv \|r_0\|^2 \left(\left[\left(\tilde{\mathbf{T}}_{k+1} \right)^{-1} \right]_{1,1} - (\mathbf{T}_k^{-1})_{1,1} \right) = \gamma_k \|r_k\|^2 \\ g_k^{(\mu)} &\equiv \|r_0\|^2 \left(\left[\left(\tilde{\mathbf{T}}_{k+1}^{(\mu)} \right)^{-1} \right]_{1,1} - (\mathbf{T}_k^{-1})_{1,1} \right) = \tilde{\gamma}_k^{(\mu)} \|r_k\|^2 \\ g_k^{(\eta)} &\equiv \|r_0\|^2 \left(\left[\left(\tilde{\mathbf{T}}_{k+1}^{(\eta)} \right)^{-1} \right]_{1,1} - (\mathbf{T}_k^{-1})_{1,1} \right) = \tilde{\gamma}_k^{(\eta)} \|r_k\|^2 \\ g_k^{(\mu,\eta)} &\equiv \|r_0\|^2 \left(\left[\left(\tilde{\mathbf{T}}_{k+1}^{(\mu,\eta)} \right)^{-1} \right]_{1,1} - (\mathbf{T}_k^{-1})_{1,1} \right) = \tilde{\gamma}_k^{(\mu,\eta)} \|r_{k-1}\|^2.\end{aligned}\tag{4.24}$$

Díky blízkému vztahu mezi koeficienty $\gamma_k^{(\mu)}$, $\gamma_k^{(\eta)}$, $\gamma_k^{(\mu,\eta)}$ a hodnotami g_k , $g_k^{(\mu)}$, $g_k^{(\eta)}$, $g_k^{(\mu,\eta)}$ viz (4.24) lze z (4.22) jednoduše odvodit rekurentní vztahy pro g_k , $g_k^{(\mu)}$, $g_k^{(\eta)}$, $g_k^{(\mu,\eta)}$ a můžeme

je přepsat takto

$$\begin{aligned}
g_k^{(\mu)} &= \|r_k\|^2 \frac{g_{k-1}^{(\mu)} - g_{k-1}}{\mu(g_{k-1}^{(\mu)} - g_{k-1}) + \|r_k\|^2} \\
g_k^{(\eta)} &= \|r_k\|^2 \frac{g_{k-1}^{(\eta)} - g_{k-1}}{\eta(g_{k-1}^{(\eta)} - g_{k-1}) + \|r_k\|^2} \\
g_k^{(\mu,\eta)} &= \frac{(g_{k-1}^{(\mu)} - g_{k-1})(g_{k-1}^{(\eta)} - g_{k-1})(\eta - \mu)}{\eta(g_{k-1}^{(\eta)} - g_{k-1}) - \mu(g_{k-1}^{(\mu)} - g_{k-1})}.
\end{aligned} \tag{4.25}$$

Nyní můžeme popsat samotný výpočet horních a dolních odhadů **A**-normy chyby pomocí hodnot $g_k, g_k^{(\mu)}, g_k^{(\eta)}, g_k^{(\mu,\eta)}$, které dosadíme do vztahu (4.13).

1. Nejprve vypočteme aproximaci řešení x_k , pomocí **metody sdružených gradientů** tak, jak jsme uvedli v první kapitole.
2. Ve druhé části, kterou jsme si nazvali **Kvadrurní část** spočítáme hodnoty $g_k, g_k^{(\mu)}, g_k^{(\eta)}, g_k^{(\mu,\eta)}$ z odvozených vzorců (4.25), které použijeme k odhadům **A**-normy chyby pomocí Gaussova, Gauss-Radau a Gauss-Lobatto pravidla.
3. Ve třetí části vypočítáme jednotlivé odhady v kroku $k - d$ metody sdružených gradientů. Hodnoty jednotlivých odhadů určíme následujícími vztahy.

Jestliže $k > d$ pak

$$\begin{aligned}
GQ_{k-d,d} &= \sum_{j=k-d+1}^k g_j \\
\text{Odhad_}GQ_{(k-d)} &= \sqrt{GQ_{k-d,d}} && \text{Gauss} \\
\text{Odhad_}GQ_{(k-d)}^{(\mu)} &= \sqrt{GQ_{k-d,d} + g_k^{(\mu)}} && \text{Gauss-Radau}^{(\mu)} \\
\text{Odhad_}GQ_{(k-d)}^{(\eta)} &= \sqrt{GQ_{k-d,d} + g_k^{(\eta)}} && \text{Gauss-Radau}^{(\eta)} \\
\text{Odhad_}GQ_{(k-d)}^{(\mu,\eta)} &= \sqrt{GQ_{k-d,d} + g_k^{(\mu,\eta)}} && \text{Gauss-Lobatto},
\end{aligned} \tag{4.26}$$

kde $\text{Odhad_}GQ_{(k-d)}$ a $\text{Odhad_}GQ_{(k-d)}^{(\eta)}$ nám počítají dolní odhad **A**-normy chyby pokud $\eta \geq \lambda_n$ a $\text{Odhad_}GQ_{(k-d)}^{(\mu)}$ a $\text{Odhad_}GQ_{(k-d)}^{(\mu,\eta)}$ nám počítají horní odhad **A**-normy chyby pokud $\mu \leq \lambda_1$ a $\eta \geq \lambda_n$. Můžeme tedy zapsat algoritmus pro výpočet horních a dolních odhadů **A**-normy chyby.

Algoritmus 3. CGQ

input $\mathbf{A}, b, x_0, \mu, \eta$

$$r_0 = b - Ax_0, p_0 = r_0$$

$$g_0^{(\mu)} = \frac{\|r_0\|^2}{\mu}, g_0^{(\eta)} = \frac{\|r_0\|^2}{\eta}$$

for $k = 1, 2, \dots$ k -tý krok metody sdružených gradientů

$$g_{k-1} = \gamma_{k-1} \|r_{k-1}\|^2,$$

$$g_k^{(\mu)} = \|r_k\|^2 \frac{g_{k-1}^{(\mu)} - g_{k-1}}{\mu(g_{k-1}^{(\mu)} - g_{k-1}) + \|r_k\|^2},$$

$$g_k^{(\eta)} = \|r_k\|^2 \frac{g_{k-1}^{(\eta)} - g_{k-1}}{\eta(g_{k-1}^{(\eta)} - g_{k-1}) + \|r_k\|^2},$$

$$\gamma_k^{(\mu, \eta)} = \frac{(g_{k-1}^{(\mu)} - g_{k-1})(g_{k-1}^{(\eta)} - g_{k-1})(\eta - \mu)}{\eta(g_{k-1}^{(\eta)} - g_{k-1}) - \mu(g_{k-1}^{(\mu)} - g_{k-1})}$$

odhady(k,d)

Tento algoritmus, který byl odvozen a popsán v [10] je zjednodušením následujícího algoritmu 4.CGQL, který byl uveden v [4] a [5]. V [10] bylo také ukázáno, že formule počítané novým algoritmem dávají přesnější výsledky ve smyslu relativní chyby.

Algoritmus 4. CGQL

input \mathbf{A}, b, x_0, μ

$$r_0 = b - Ax_0, p_0 = r_0$$

$$\delta_0 = 0, \gamma_{-1} = 1, c_1 = 1, \beta_0 = 0, d_0 = 1, \tilde{\alpha}_1^{(\mu)} = \mu, \tilde{\alpha}_1^{(\eta)}$$

for $k = 1$ **to** dokud konvergence nedosáhne námi požadované přesnosti

k-tý krok metody sdružených gradientů

$$\alpha_k = \frac{1}{\gamma_{k-1}} + \frac{\delta_{k-1}}{\gamma_{k-2}}, \beta_k^2 = \frac{\delta_k}{\gamma_{k-1}^2}$$

$$d_k = \alpha_k - \frac{\beta_{k-1}^2}{d_{k-1}},$$

$$g_k = \|r_0\|^2 \frac{c_k^2}{d_k}$$

$$\bar{d}_k^{(\mu)} = \alpha_k - \tilde{\alpha}_k^{(\mu)}, \quad \tilde{\alpha}_{k+1}^{(\mu)} = \mu + \frac{\beta_k^2}{\bar{d}_k^{(\mu)}}, \quad g_k^{(\mu)} = \|r_0\|^2 \frac{\beta_k^2 c_k^2}{d_k \left(\alpha_{(k+1)}^{(\mu)} d_k - \beta_k^2 \right)}$$

$$\bar{d}_k^{(\eta)} = \alpha_k - \tilde{\alpha}_k^{(\eta)}, \quad \tilde{\alpha}_{k+1}^{(\eta)} = \eta + \frac{\beta_k^2}{\bar{d}_k^{(\eta)}}, \quad g_k^{(\eta)} = \|r_0\|^2 \frac{\beta_k^2 c_k^2}{d_k \left(\alpha_{(k+1)}^{(\eta)} d_k - \beta_k^2 \right)}$$

$$\tilde{\alpha}_{k+1}^{(\mu, \eta)} = \frac{\bar{d}_k^{(\mu)} \bar{d}_k^{(\eta)}}{\bar{d}_k^{(\eta)} - \bar{d}_k^{(\mu)}} \left(\frac{\eta}{\bar{d}_k^{(\mu)}} - \frac{\mu}{\bar{d}_k^{(\eta)}} \right),$$

$$\left(\tilde{\beta}_k^{(\mu, \eta)} \right)^2 = \frac{\bar{d}_k^{(\mu)} \bar{d}_k^{(\eta)}}{\bar{d}_k^{(\eta)} - \bar{d}_k^{(\mu)}} (\eta - \mu)$$

$$g_k^{(\mu, \eta)} = \|r_0\|^2 \frac{\left(\tilde{\beta}_k^{(\mu, \eta)} \right)^2 c_k^2}{d_k \left(\tilde{\alpha}_{k+1}^{(\mu)} d_k - \left(\tilde{\beta}_k^{(\mu, \eta)} \right)^2 \right)}$$

$$c_{k+1}^2 = \frac{\beta_k^2 c_k^2}{d_k^2}$$

odhady(k,d)

Pro úplnost uvádíme i původní algoritmus, který explicitně počítá prvky modifikované Jacobiho matice $\tilde{\mathbf{T}}_{k+1}$. Já tento algoritmus ve své práci používala k některým numerickým experimentům právě proto, že přímo počítá koeficienty $\tilde{\alpha}_{k+1}, \tilde{\beta}_k$.

Kapitola 5

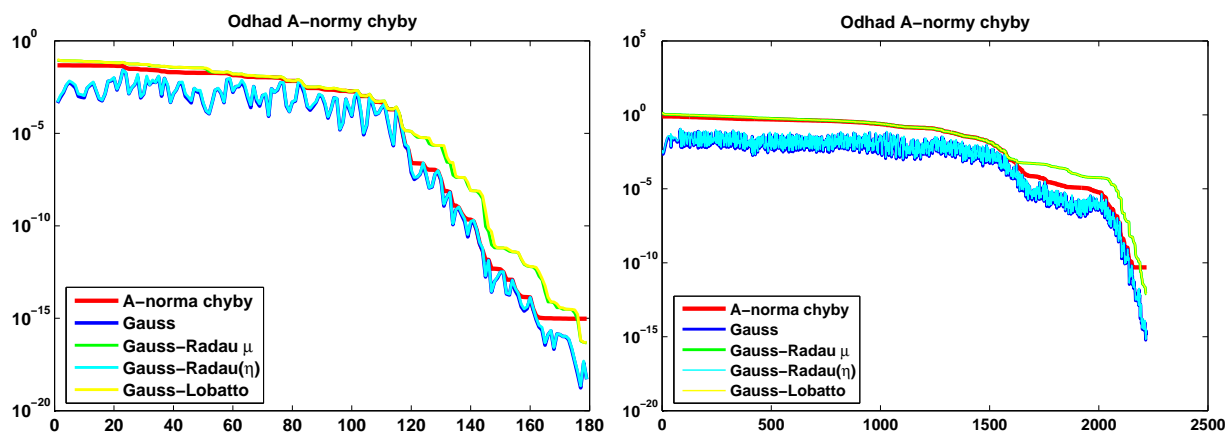
Numerické experimenty

V této kapitole se budeme věnovat odhadování A -normy chyby pomocí různých kvadrurních pravidel vytvořených v programu MATLABTM podle algoritmu 3.CGQ tak, jak jsme ho uvedli v předešlé kapitole. Ukážeme, jak pomocí tohoto algoritmu můžeme získat dobré dolní i horní odhady A -normy chyby v k -tém kroku CG. Ukážeme ale také příklady, ve kterých můžeme získat velmi nepřesný dolní i horní odhad. Pokusíme se říci, jak tyto nepřesnosti odhalit, popř. jak je odstranit např. vhodnou volbou parametru d . Ke všem experimentům jsem používala matice, jejichž zdrojem byl MATRIX MARKET¹ a THE UNIVERSITY OF FLORIDA SPARSE MATRIX COLLECTION², jednotlivé parametry matic jsou uvedeny v samostatné části v závěru kapitoly. Ve všech experimentech jsem používala vektor pravých stran b jako vektor samých jedniček. Pokud není v grafu uvedena jiná hodnota parametrů, volila jsem vždy $\mu = \lambda_1$, $\eta = \lambda_n$ a $d = 1$, kde λ_1 a λ_n je nejmenší a největší vlastní číslo dané matice A vypočtené standardní funkcí MATLABTM u `eig.m`.

Na úvod této kapitoly si ukážeme, jaké odhady pomocí Gaussových kvadratur můžeme získat viz obrázek 5.1. Gaussova a Gauss-Radau^(η) kvadratura nám dává dolní odhad a Gauss-Radau^(μ) a Gauss-Lobatto kvadratura horní odhad A -normy chyby. Z obrázku je vidět, že dolní odhady Gauss a Gauss-Radau^(η) jsou téměř shodné, stejně tak horní odhady Gauss-Radau^(μ) a Gauss-Lobatto. Protože se na první pohled zdá, že dolní odhad Gauss-Radau^(η) ani horní odhad Gauss-Lobatto nám nedávají výrazně jiné výsledky oproti zbylým dvěma odhadům, v další části se proto budeme zajímat především o chování dolního odhadu založeném na Gaussově kvadratuře a o chování horního odhadu založeném na Gauss-Radau^(μ) kvadratuře. Navíc z hlediska praktického využití, kde chceme volit $\mu = \lambda_1$ a $\eta = \lambda_n$, použijeme-li dolní odhad založený na Gauss-Radau^(η) kvadratuře, musíme nejprve získat aproximaci λ_n , kdežto dolní odhad založený na Gaussově kvadratuře získáme přímo z hodnot počítaných CG. Podobně v praktickém využití horního odhadu Gauss-Lobatto, kde předepisujeme dva uzly a musíme tedy znát aproximaci λ_1 i λ_n oproti Gauss-Radau^(μ), kde předepisujeme pouze

¹<http://math.nist.gov/MatrixMarket>

²<http://www.cise.ufl.edu/research/sparse/matrices>



Obrázek 5.1: Horní a dolní odhady matic *Bcsstk01* a *Nos1*

jeden uzel a hledáme tedy aproximaci λ_1 .

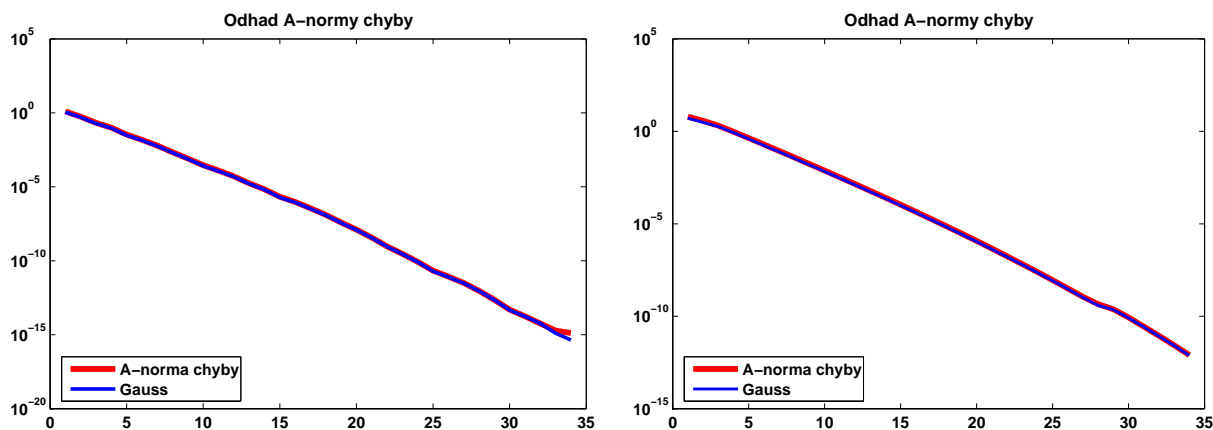
5.1 Výhody a nevýhody dolního a horního odhadu

Nejprve si uvedeme a na obrázcích ukážeme jaké mohou být výhody a nevýhody dolního a horního odhadu založeném na Gaussově a Gauss-Radau^(μ) kvadratuře. Ukážeme, kdy nám tyto odhady dávají kvalitní informaci o A -normě chyby a můžeme se tedy na výpočet spolehnout a použít ho k zastavení metody sdružených gradientů. Také ale ukážeme příklady, ve kterých uvidíme, že pokud bychom se spolehli na získanou informaci a metodu zastavili, dostali bychom aproximaci, která by byla velmi vzdálená od naší požadované přesnosti.

5.1.1 Dolní odhad

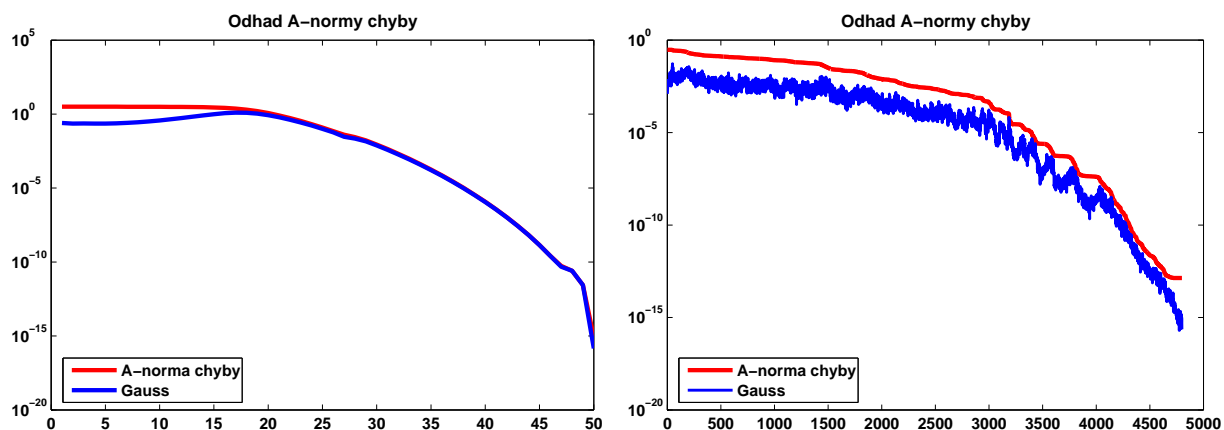
Víme, že dolní odhad pomocí Gaussovy kvadratury je numericky stabilní viz [11] v tom smyslu, že nám dává vždy dolní odhad aktuální A -normy chyby v k -tém kroku CG i přesto, že vlivem konečné aritmetiky dochází ke ztrátě ortogonality reziduí r_k . Je-li splněna podmínka (2.8), jak jsme uvedli v první kapitole o metodě sdružených gradientů $\|x - x_k\|_A \gg \|x - x_{k+d}\|_A$, získáme dolní odhad, který dobře aproximuje A -normu chyby.

Na obrázku 5.2 je vidět, že pokud A -norma chyby „rozumně“ klesá, dostáváme velmi dobrý dolní odhad A -normy chyby, dokud nedosáhneme limitní hladiny přesnosti A -normy chyby. Také jsme ale řekli, že parametr d obecně neznáme.



Obrázek 5.2: Dolní odhad pro matice *Mesh1em6* a *Strakos(48,0.1,1,0.99)*

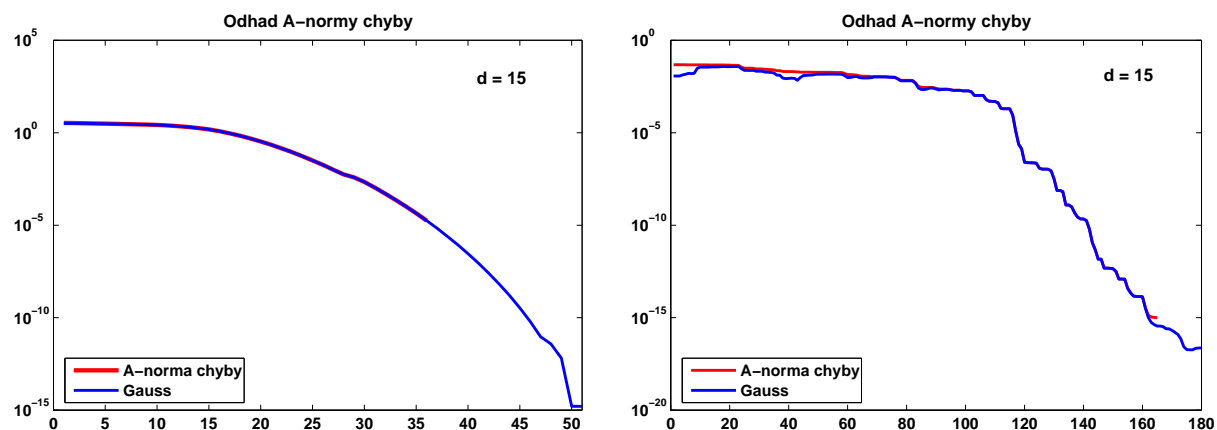
Na obrázku 5.3 je vidět, že pokud A-norma chyby stagnuje, tedy není splněna podmínka (2.8), dolní odhad podhodnocuje aktuální A-normu chyby. Kdybychom v těchto místech výpočet zastavili, nedosáhli bychom námi požadované přesnosti a skutečná A-norma chyby může být větší i o několik řádů.



Obrázek 5.3: Dolní odhad pro matice *Strakos(48,0.01,1000,0.99)* a *Bcsstk06*

Tento problém se můžeme pokusit řešit vhodnou volbou parametru d . Zvolíme-li na začátku výpočtu $d > 1$, je z obrázku 5.4 zřejmé, že ačkoliv máme dvě stejně velké matice $n = 48$, stejná volba d nám u jedné matice zajistí dobrý dolní odhad, ale u jiné matice bychom museli zvolit d větší. Nemáme tedy žádný návod, jak parametr d na začátku výpočtu zvolit tak, abychom dosáhli co nejpřesnějšího dolního odhadu. Navíc v praktickém použití, pokud bychom zvláště u velkých matic volili parametr d pro celý výpočet velký, museli bychom počítat příliš mnoho iterací navíc. Jiný přístup je, že se můžeme pokusit pouze rozpoznat místa, kde A-norma chyby stagnuje a v těchto

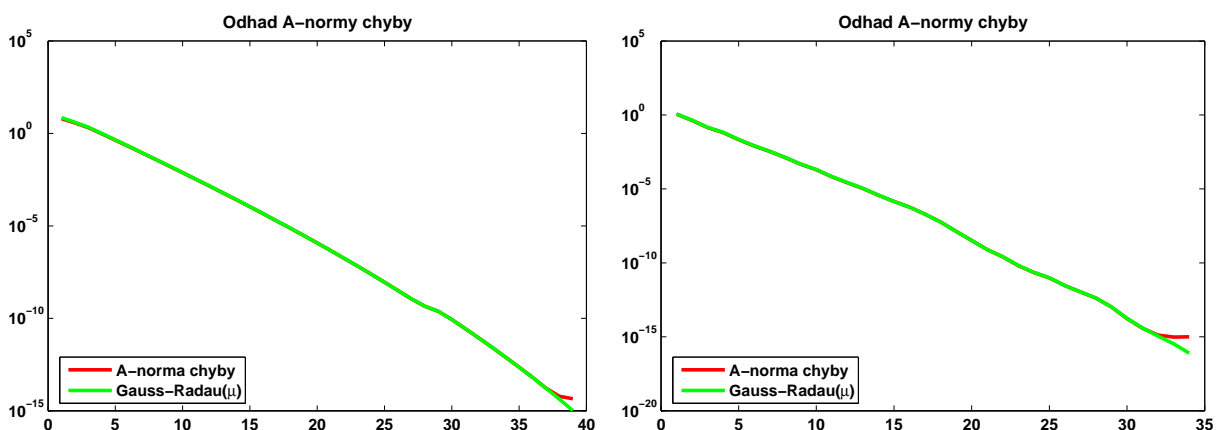
místech zvětšit parametr d . Protože k takovému přístupu se pokusíme využít horní odhad Gauss-Radau^(μ), nejprve si ukážeme vlastnosti tohoto odhadu a poté se budeme věnovat adaptivní volbě parametru d .



Obrázek 5.4: Parametr $d = 15$ pro matice *Strakos*(48,0.01,1000,0.99) a *Bcsstk01*

5.1.2 Horní odhad

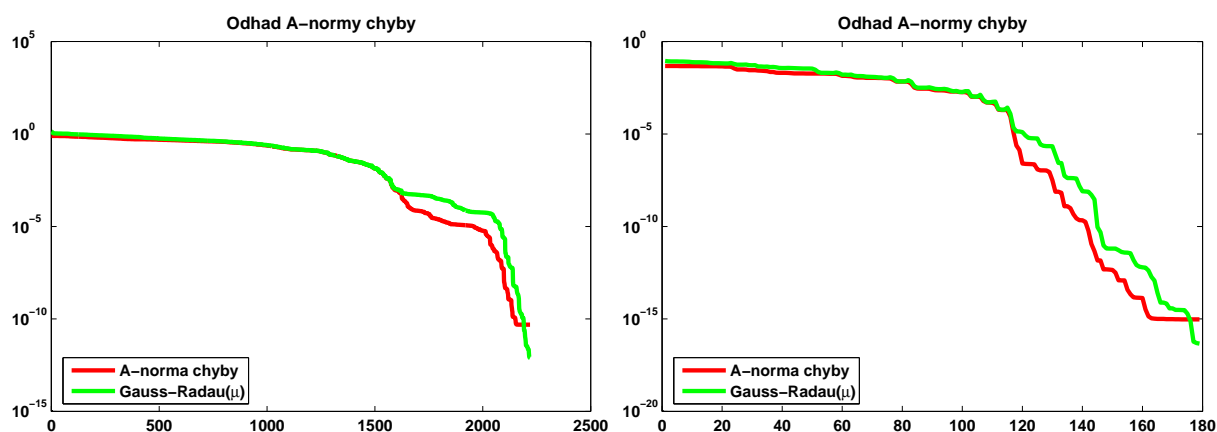
Odhad pomocí Gauss-Radau^(μ) kvadratury nám dává většinou horní odhad. Na obrázku 5.5 můžeme pozorovat, že pokud A-norma chyby „rozumně“ klesá, dostáváme téměř přesnou hodnotu A-normy chyby.



Obrázek 5.5: Horní odhad pro matice *Strakos*(48,0.1,1,0.99) a *Mesh1e1*

Jak je vidět na obrázku 5.6, může se ale stát, že v některých místech dochází k nestabilitě. Odhad, který byl velmi přesný, se v nějakém místě výrazně zhorší. Nevýhodou použití tohoto odhadu je, že na začátku musíme zvolit vstupní parametr μ . Abychom

mohli použít teoretické poznatky, které jsme si odvodili, chceme volit $\mu \leq \lambda_1$. V praktických úlohách ale většinou λ_1 neznáme. Pokud chceme použít tento odhad, musíme nejprve určit alespoň aproximaci λ_1 . Existují ale metody, které se touto problematikou zabývají a dokáží velmi rychle a efektivně aproximaci λ_1 získat.



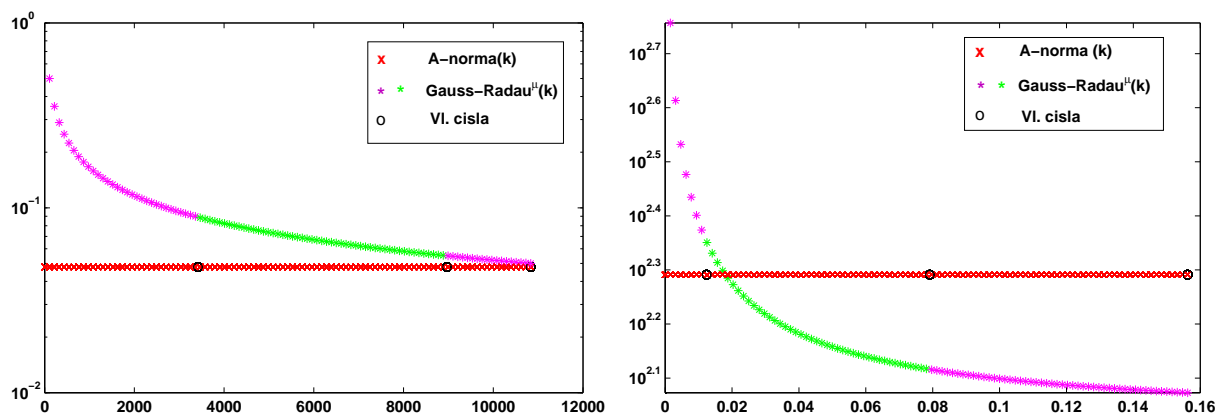
Obrázek 5.6: Horní odhad pro matice *Nos1* a *Bcsstk01*

5.2 Odhad založený na Gauss-Radau kvadratuře

Uvedli jsme, že dolní odhad pomocí Gaussovy kvadratury je numericky stabilní. Že i při ztrátě ortogonality reziduí r_k vlivem konečné aritmetiky nám dává vždy dolní odhad aktuální A-normy chyby. Ale jak již bylo řečeno, nevíme jak volit parametr d . Tedy jak rozpoznat, že je splněna podmínka (2.8) a během d kroků došlo k dostatečnému poklesu A-normy chyby. K volbě tohoto parametru bychom chtěli využít horního odhadu A-normy chyby založeného na Gauss-Radau^(μ) kvadratuře a proto se dále budeme věnovat vlastnostem a chování tohoto odhadu. Pomocí experimentů jsem ale zjistila různé překážky, které tento záměr komplikují.

5.2.1 Volba μ

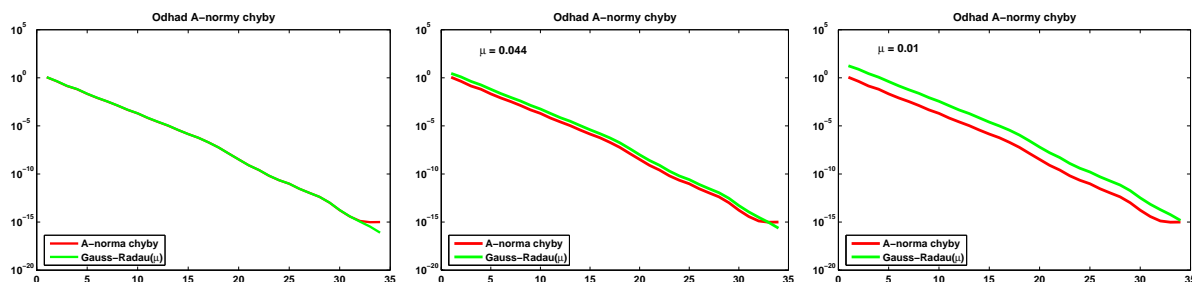
Volba předepsaného uzlu μ je pro tento odhad samozřejmě klíčová. Aby platily teoretické vlastnosti, které jsme si ukázali v předešlých kapitolách, chceme volit $\mu \leq \lambda_1$. Protože ale v praktických výpočtech můžeme získat aproximaci nejmenšího vlastního čísla λ_1 , pro kterou tato podmínka nemusí platit, ukážeme i příklady odhadů, kde volíme $\mu \in I = [\lambda_1, \lambda_n]$. Na obr. 5.7 můžeme vidět, jak různá volba μ na začátku algoritmu nám dává různé hodnoty odhadu v prvním kroku $k = 1$ CG pro volby $\mu \in (0, \lambda_3)$, kde λ_3 je třetí nejmenší vlastní číslo dané matice A .



Obrázek 5.7: Horní odhady pro matice *Bcsstk01* a *494_bus* v $k = 1$ pro volby $\mu \in (0, \lambda_3)$

- $\mu \leq \lambda_1$

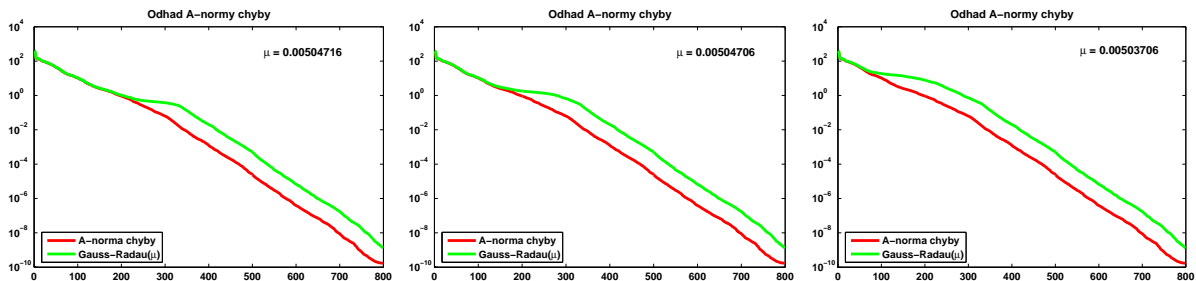
Pokud pro danou matici A *nedochází ke zpoždění konvergence CG*, tzn. nedochází k velkým zaokrouhlovacím chybám vlivem konečné aritmetiky, zvolíme-li $\mu \leq \lambda_1$, dostaneme horní odhad A-normy chyby. Čím blíže μ volíme k λ_1 , tím lepší odhad získáme tak, jak bychom očekávali (obrázek 5.8).



Obrázek 5.8: Horní odhady pro matici *Mesh1e1* pro 3 různá μ

Pokud ale pro danou matici A *dochází ke zpoždění konvergence CG*, tzn. dochází ke ztrátě ortogonalit reziduí $r_{k,r}$, nastává v některém „místě nestabilita“, tj. výrazné zhoršení odhadu, který do té doby byl velice přesný. Tohle místo zůstává, i když volíme μ co nejbližší k λ_1 . Pokud budeme μ volit menší, dochází k tomu, že se „místo nestability“ objevuje v dřívějších iteracích viz obrázek 5.9. Pokud zvolíme $\mu \ll \lambda_1$ dostaneme od začátku odhad, který velmi nadhodnocuje A-normu chyby. Mohlo by se zdát, že alespoň kopíruje tvar křivky A-normy chyby a neobjevuje se v něm „místo nestability“, což by nám dávalo dobrou informaci o poklesu či stagnaci A-chyby. Když ale společně s odhadem vykreslíme relativní chybu odhadu

$$\left| \frac{\|x - x_k\|_A - \text{Odhad_GQ}_{(k-d)}}{\|x - x_k\|_A} \right|$$

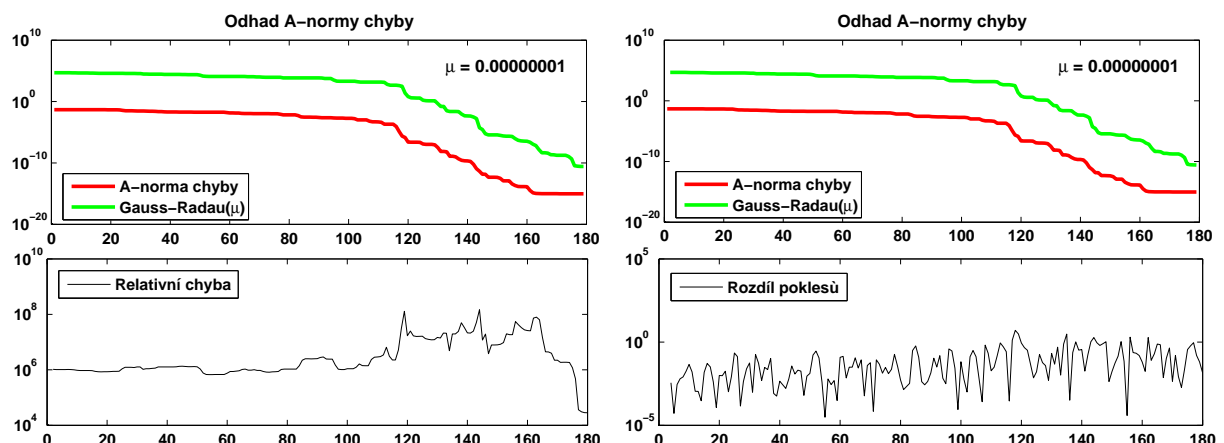


Obrázek 5.9: Horní odhady pro matici *662_bus* pro 3 různá μ

vidíme na obr. 5.10 vlevo, že v „místě nestability“, tj. přibližně v iteraci $k = 118$ dochází stále ke zhoršení odhadu jako na obr. 5.6 vpravo. Pokud by pokles A-normy chyby byl téměř stejný jako pokles horního odhadu, platilo by pro relativní chybu

$$\left| \frac{\frac{\|x - x_{k+d}\|_A}{\|x - x_k\|_A} - \frac{\text{Odhad_GQ}_{k+d}}{\text{Odhad_GQ}_k}}{\frac{\|x - x_{k+d}\|_A}{\|x - x_k\|_A}} \right| \approx 0.$$

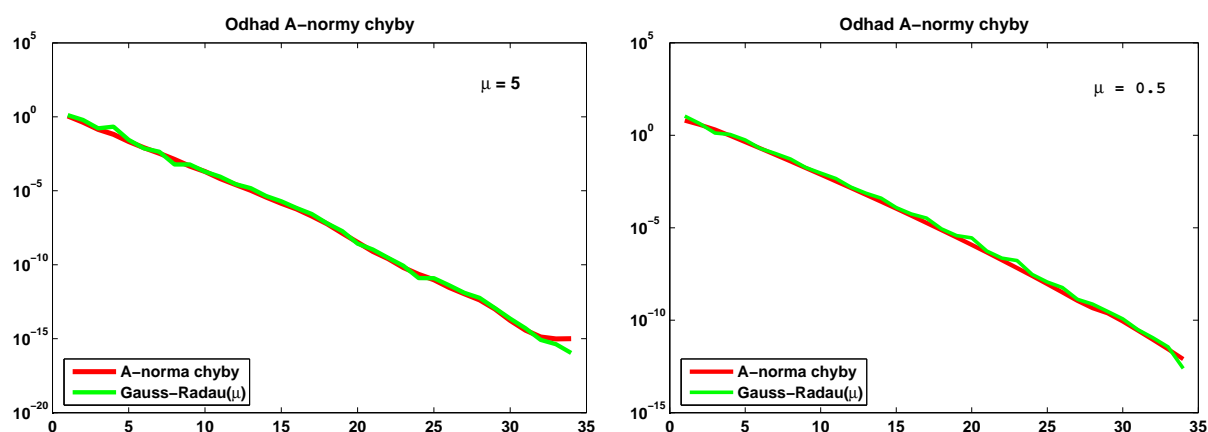
Vykreslíme-li relativní chybu, vidíme na obr. 5.10 vpravo, že přibližně v iteraci $k = 118$ rozdíl poklesů dosáhne nejvyšší hodnoty a ta se příliš nemění až do konce výpočtu. Tedy v „místě nestability“ klesá horní odhad pomaleji než A-norma chyby, ale dál už je pokles horního odhadu přibližně stejný jako pokles A-normy chyby. Takže i pokud zvolíme „špatné“ μ , tj. $\mu \ll \lambda_1$ dostaneme velmi nepřesný horní odhad, ve kterém se nám projeví „místo nestability“.



Obrázek 5.10: Relativní chyba a rozdíl poklesů horního odhadu pro matici *Bcsstk01*

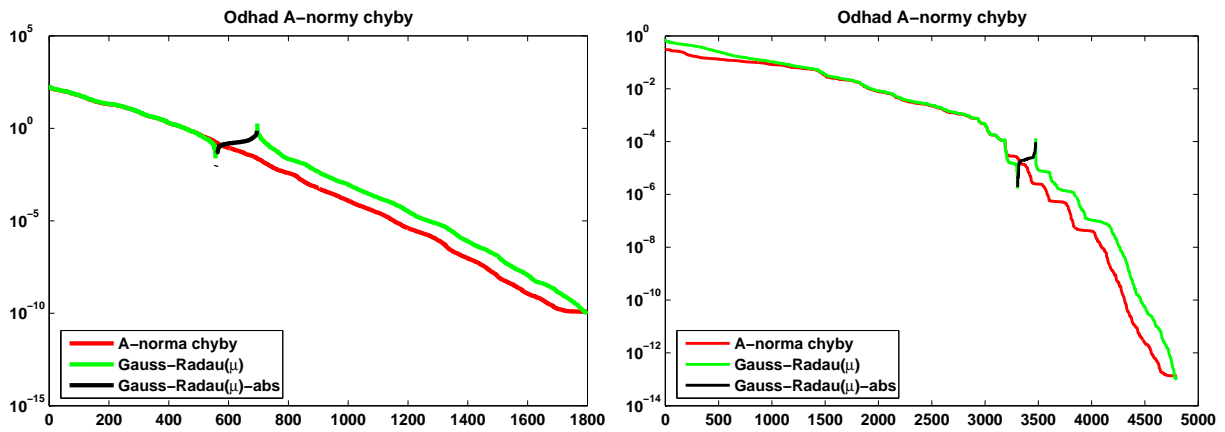
- $\mu \in [\lambda_1, \lambda_n]$

Další možností, kterou jsem ve svých experimentech testovala, bylo volit μ uvnitř intervalu $[\lambda_1, \lambda_n]$. Pokud zvolíme μ uvnitř intervalu, z hlediska teorie, kterou jsme si odvodili, už nemáme žádné předpoklady, že takovou volbou získáme horní odhad. Pokud pro danou matici A *nedochází ke zpoždění konvergence*, dostaneme odhad A-normy chyby, který nemusí být vždy horní. Může se stát, že v některých místech je g_k^μ ve vztahu (4.26) záporné a dostaneme dolní odhad. Při výpočtu ale nedochází k žádné nestabilitě viz obrázek 5.11.



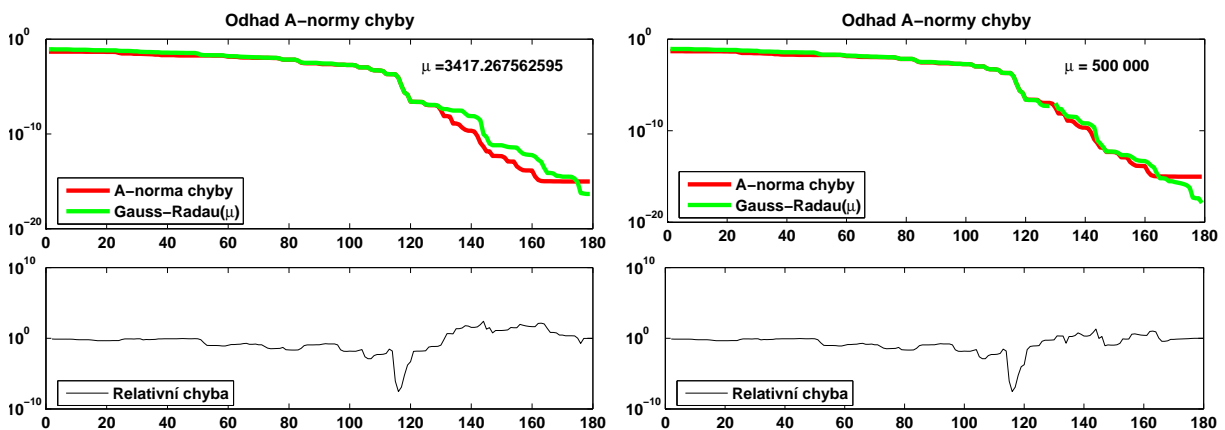
Obrázek 5.11: Horní odhad pro $\mu \in [\lambda_1, \lambda_n]$, matice *Strakos(48,0.1,1,0.99), Mesh1e1*

Pokud pro danou matici A *dochází ke zpoždění konvergence* dostáváme odhad, který není vždy horní, ale může být i dolní stejně jako v předchozích příkladech, ale v „místě nestability“ je g_k^μ záporné a zároveň je větší než $GQ_{(k-d)}$. Potom bychom $Odhad_{GQ_{(k-d)}}$ ve vztahu (4.26) počítali jako odmocninu ze záporného čísla, což by v přesné aritmetice nemohlo nastat. V grafu jsem použila absolutní hodnotu a vykreslila odhad černou barvou a tak je vidět, kde je $g_k^{(\mu)}$ záporné a zároveň je větší než $GQ_{(k-d)}$ viz obrázek 5.12.



Obrázek 5.12: Horní odhad pro matice *494_bus*, *Bcsstk06*

- Poslední možností, kterou jsem testovala při volbě uzlu μ , bylo předepsat na začátku $\mu_1 \leq \lambda_1$ a v průběhu výpočtu zvolit jiné $\mu = \mu_2$ z intervalu $[\lambda_1, \lambda_n]$ viz obr. 5.13 a 5.18. Z hlediska teorie, kterou jsme ukázali, nevím, co přesně tahle změna volby předepsaného uzlu μ znamená a jak tuto změnu popsat pomocí modifikované Jacobiho matice. Volba μ_2 byla tedy pouze experimentální.



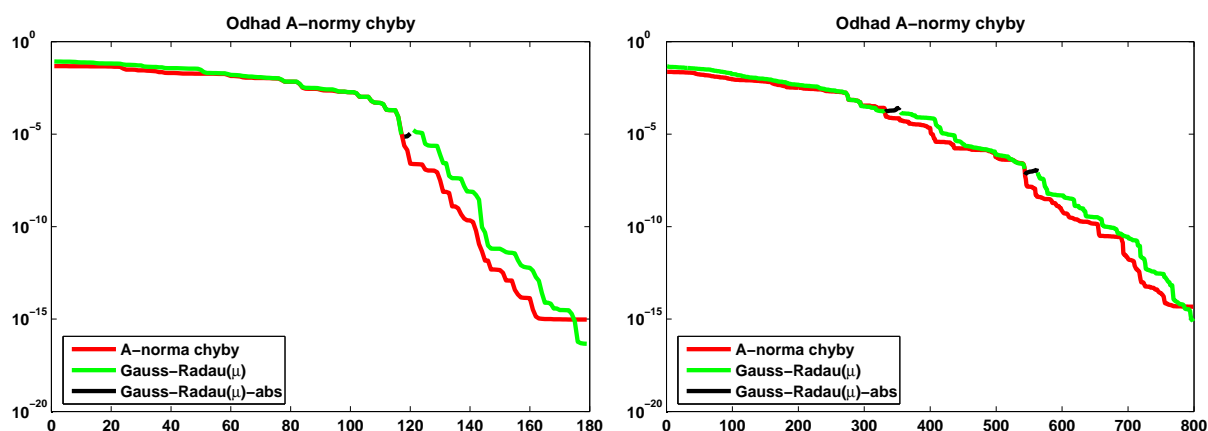
Obrázek 5.13: Volba μ_2 v iteraci $k = 118$ pro matici *Bcsstk01*

5.2.2 „Místo nestability“

Dalším problémem při výpočtu horního odhadu pomocí Gauss-Radau^(μ) kvadratury je „místo nestability“, tj. místo, ve kterém dochází k výraznému zhoršení doposud velmi přesného horního odhadu. V předchozích příkladech jsme ukázali, že vzniká u matic A , pro které dochází ke zpoždění konvergence CG. V této části se tedy pokusíme navrhnout, jak toto místo rozpoznat, popř. jak v něm horní odhad vylepšit.

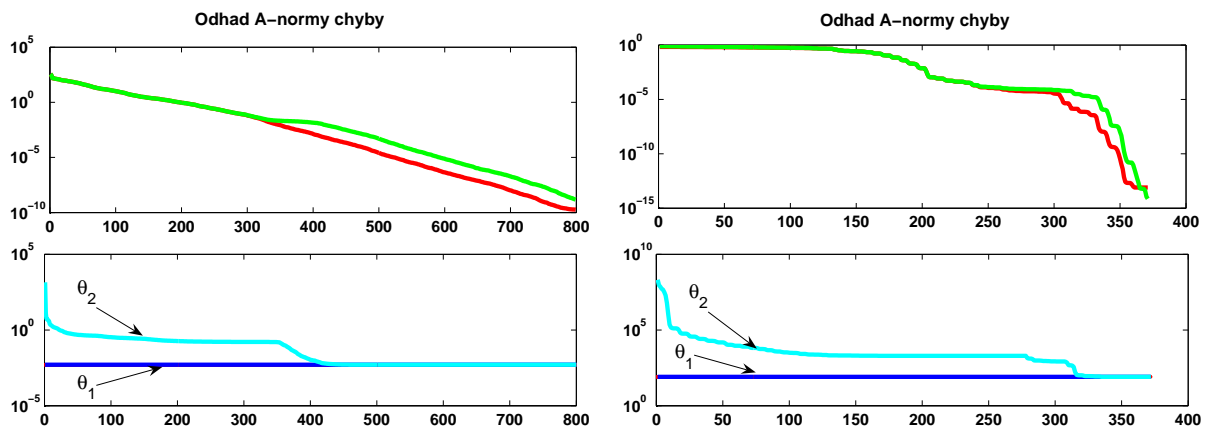
1. Jak rozpoznat „místo nestability“.

- Pokud volíme $\mu \leq \lambda_1$ a někde ve výpočtu se objeví $g_k^{(\mu)} < 0$ víme, že se jedná o místo nestability, protože, jak jsme odvodili pro takovou volbu μ , platí, že $g_k^{(\mu)}$ je kladné. Volíme-li μ uvnitř intervalu $[\lambda_1, \lambda_n]$, jak jsme ukázali na příkladech, může se ve výpočtu někde objevit $g_k^{(\mu)} < 0$, které zároveň je větší než $GQ_{(k-d)}$. Potom počítáme $Odhad_GQ_{(k-d)}^{(\mu)}$ jako odmocninu ze záporného čísla a víme tedy, že se jedná o „místo nestability“ viz obr. 5.14.



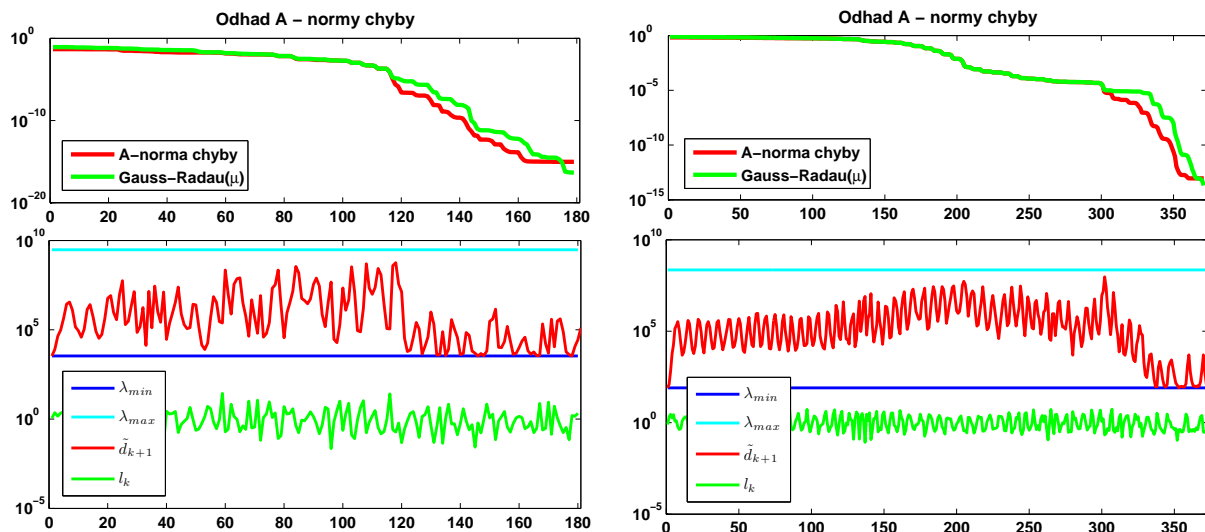
Obrázek 5.14: Horní odhad a místo nestability pro matice $Bcsstk01$, $Bcsstk03$

- „Místo nestability“ nastává přibližně v té iteraci, kde druhé nejmenší vlastní číslo modifikované Jacobiho matice \tilde{T}_{k+1} začne konvergovat k nejmenšímu vlastnímu číslu Jacobiho matice viz obr. 5.15, což odpovídá tomu, že se snažíme předepsat vlastní číslo modifikované Jacobiho matice \tilde{T}_{k+1} , které už je vlastním číslem. Nebo-li implicitně řešíme systém s posunutou maticí $T_k - \mu I$, která je téměř singulární, což právě v konečné aritmetice může způsobovat problémy. Hledat ale místo nestability během výpočtu pomocí vlastních čísel Jacobiho matice by nám výpočet velice zpomalovalo.



Obrázek 5.15: Horní odhad a konvergence dvou nejmenších vlastních čísel Jacobiho matice - pro matice *662bus*, *Lund_a*

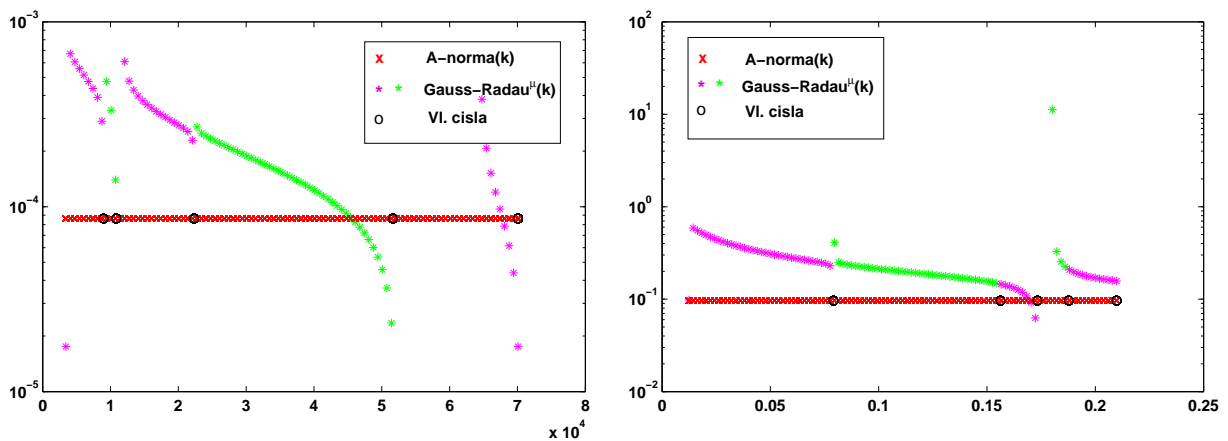
- Vznik „místa nestability“ bychom předpokládali, že se projeví v prvcích modifikované Jacobiho matice $\tilde{\mathbf{T}}_{k+1}$. V dalších experimentech jsem tedy sledovala velikost prvků LDL^T rozkladu modifikované Jacobiho matice \tilde{d}_{k+1} a l_k . Z grafů je vidět, že pokud se prvek \tilde{d}_{k+1} přiblíží k největšímu vlastnímu číslu, vzniká „místo nestability“ viz obr.5.16. Prvky \tilde{d}_{k+1} můžeme snadno získávat v algoritmu 4.CGQL.



Obrázek 5.16: Horní odhad a velikost prvků \tilde{d}_{k+1} pro matice *Bssctk01*, *Lund_a*

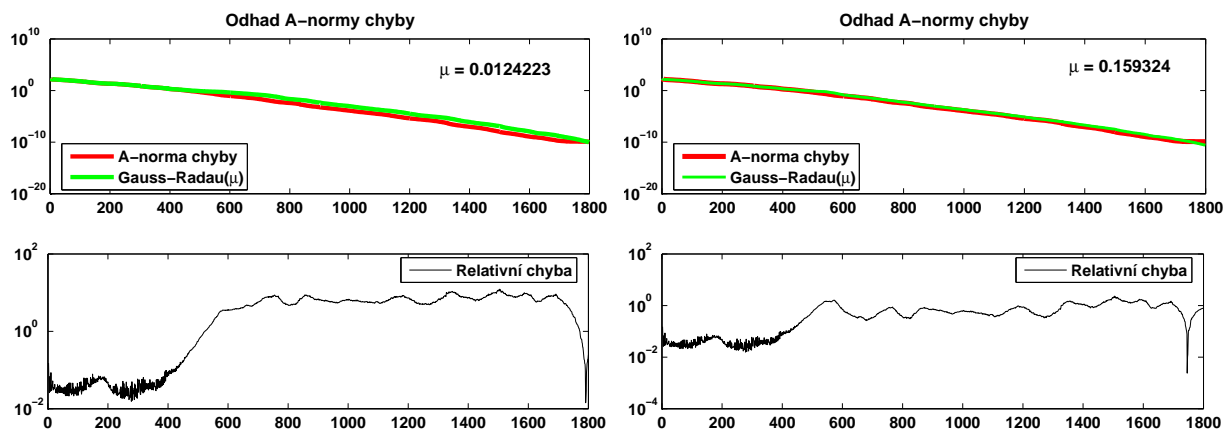
2. Jak vylepšit odhad v „místě nestability“.

- Jednou z možností je zvolit na začátku výpočtu $\mu = \mu_1$ a v průběhu výpočtu zvolit jiné $\mu = \mu_2 \in [\lambda_1, \lambda_n]$. Ale otázka je, kdy zvolit μ_2 a jakou hodnotu. Nejprve zkusme říct, kdy zvolit μ_2 . Pokud zvolíme druhé μ_2 příliš brzy, tzn. před „místem nestability“, dojde v tomto místě stejně ke zhoršení odhadu a změna μ nám nijak nepomůže. Když jej zvolíme příliš pozdě, až za „místem nestability“, hodnoty μ_2 , pro které bychom se horním odhadem přiblížili k přesné hodnotě A-normy chyby, bychom museli volit záporné. Což vzhledem k tomu, že matice A je pozitivně definitní není vhodné. Pokud si vykreslíme relativní chybu odhadu, vidíme, kde máme nejpřesnější odhad A-normy chyby a kdy dochází ke zhoršení odhadu. Než tedy dojde ke zhoršování horního odhadu, chceme zvolit $\mu = \mu_2$. Další otázka je, jaké μ_2 zvolit. Na obrázku 5.17 můžeme vidět hodnoty horních odhadů v iteraci $k = 118$, což je iterace, ve které volíme μ_2 pro matici *Bcsstk01* a v iteraci $k = 452$, ve které volíme μ_2 pro matici *494_bus*, kde počítáme odhady pro volby $\mu \in [\lambda_1, \lambda_6]$. Z obrázku vidíme, že můžeme najít několik hodnot, které pro pevně zvolené k , jsou velmi blízké A-normě chyby a to jsme na obrázku vykreslili pouze prvních šest vlastních čísel.



Obrázek 5.17: Horní odhady pro matici *Bcsstk01* v $k = 118$ a matici *494_bus* v $k = 452$

To znamená, že nějaké jednoduché a jednoznačné určení μ_2 není zřejmé a já ho určovala pouze experimentálně viz obr. 5.13 a 5.18.



Obrázek 5.18: Volba μ_2 v iteraci $k = 452$ pro matici 494_bus

- Další možností, jak se snažit vylepšit odhad v kritickém místě, je použít jakýsi „odhad dopředu“. Odvodili jsme, že platí

$$\|x - x_k\|_{\mathbf{A}}^2 = \sum_{i=k}^{k+d-1} g_i + \|x - x_{k+d}\|_{\mathbf{A}}^2$$

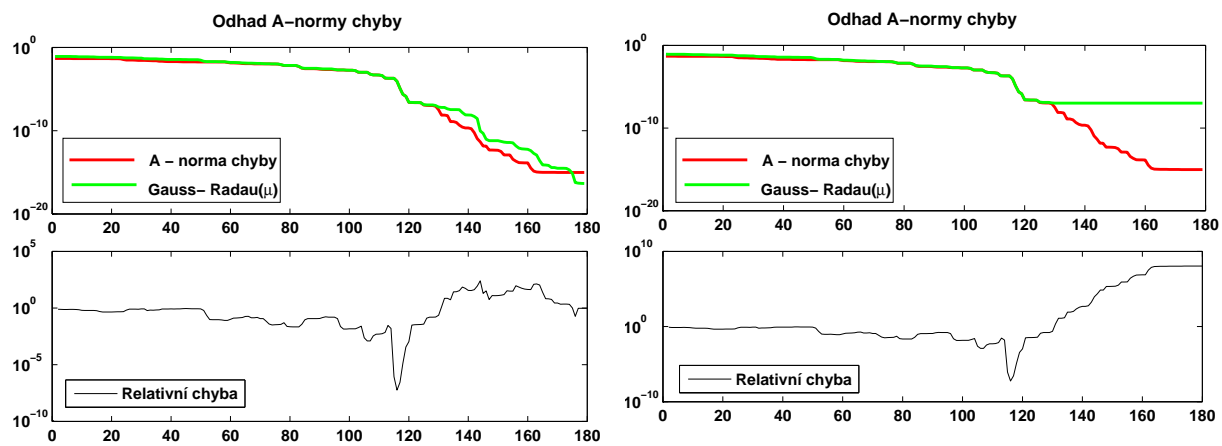
a pro $d = 1$ můžeme vztah přepsat

$$\|x - x_k\|_{\mathbf{A}}^2 - g_k = \|x - x_{k+1}\|_{\mathbf{A}}^2.$$

\mathbf{A} -normu chyby v kroku $(k+1)$ můžeme tedy zapsat pomocí \mathbf{A} -normy chyby v kroku k ve tvaru

$$\|x - x_{k+1}\|_{\mathbf{A}} = \sqrt{\|x - x_k\|_{\mathbf{A}}^2 - g_k}. \quad (5.1)$$

Pokud budeme mít dobrý horní odhad \mathbf{A} -normy chyby $\|x - x_k\|_{\mathbf{A}}$ v k -tém kroku CG ve smyslu relativní chyby, mohli bychom doufat, že i výraz (5.1) nám dá dobrý horní odhad ve stejném řádu chyby až do místa, kde odhad dosáhne této přesnosti viz obr. 5.19. Z obrázku je ale vidět, že takovýto postup nám horní odhad nezlepší, protože největší přesnost, které jsme dosáhli co nejlepší volbou $\mu = \lambda_1$, byla v řádu 10^{-7} a to nestačilo na to, aby nám tento postup zlepšil odhad v „místě nestability“.



Obrázek 5.19: Matice *Bssctk01* - „Odhad dopředu“

5.3 Adaptivní volba d

Jedním z cílů této práce bylo také navrhnout heuristiku, pomocí které by bylo možné volit v průběhu metody CG parametr d tak, abychom získali co nejpřesnější dolní odhad A -normy chyby. Tedy abychom dokázali navrhnout nějaký postup, jak rozpoznat, zda-li došlo během d -kroků CG k dostatečnému poklesu A -normy chyby a mohli říct, že máme dobrý dolní odhad chyby. Je možné na začátku zvolit pevné velké $d > 1$ a to použít během celého výpočtu. Potom, abychom dosáhli limitní přesnosti konvergence CG, musíme počítat $(k + d)$ iterací, čímž bychom u velkých matic museli provést velký počet iterací navíc. Proto se snažíme d adaptivně měnit během výpočtu. Tzn. snažíme se rozpoznat, kde ve výpočtu dochází ke stagnaci A -normy chyby.

5.3.1 Ideální d

Nejprve vysvětlíme, jak si pro teoretické účely pro vyhodnocování kvality naší heuristiky určíme ideální hodnotu parametru d , kterou potom budeme porovnávat s námi získanou hodnotou parametru d . Řekli jsme, že chceme, aby byla splněna podmínka $\|x - x_k\|_A \gg \|x - x_{k+d}\|_A$, abychom měli dobrý dolní odhad. To si můžeme přepsat následovně

$$\frac{\|x - x_{k+d}\|_A}{\|x - x_k\|_A} \ll 1. \quad (5.2)$$

Pokud není tato podmínka splněna, nedošlo k dostatečnému poklesu A -normy chyby a je třeba volit $d > 1$. Jinak řečeno, nedošlo-li k dostatečnému poklesu A -normy chyby v $(k + d)$ krocích, jsou si hodnoty $\|x - x_k\|_A \approx \|x - x_{k+d}\|_A$ velmi blízké a jejich podíl je téměř ≈ 1 . V experimentech jsem používala podmínku

$$\frac{\|x - x_{k+d}\|_A}{\|x - x_k\|_A} < 0.8, \quad (5.3)$$

která pokud nebyla splněna, nedošlo k dostatečnému poklesu A -normy chyby a bylo třeba zvětšit parametr d .

5.3.2 Algoritmy pro adaptivní volbu d

Zkoumání chování horního odhadu založeného na Gauss-Radau kvadratuře jsme se snažili využít k rozpoznávání stagnace A -normy chyby a pomocí něj volit adaptivně parametr d .

1. Jednou z možností je využít pouze horního odhadu. Stejnou podmínku jakou kládeme na pokles A -normy chyby (5.3) při volbě ideálního parametru d , budeme chtít splnit i u horního odhadu. Tzn. pokud nedojde v d krocích k dostatečnému poklesu horního odhadu, zvětšíme parametr d . Pro každý krok k určíme hodnotu

parametru d_k tak, aby podmínka poklesu horního odhadu byla splněna. Tuto podmínku (5.2) si můžeme pomoci vztahů (4.26) zapsat následovně

$$\frac{\|x - x_{k+d}\|_{\mathbf{A}}}{\|x - x_k\|_{\mathbf{A}}} \sim \frac{\sqrt{g_{k+d}^{(\mu)}}}{\sqrt{\text{Odhad_GQ}_{(k)}^{(\mu)}}} \ll 1.$$

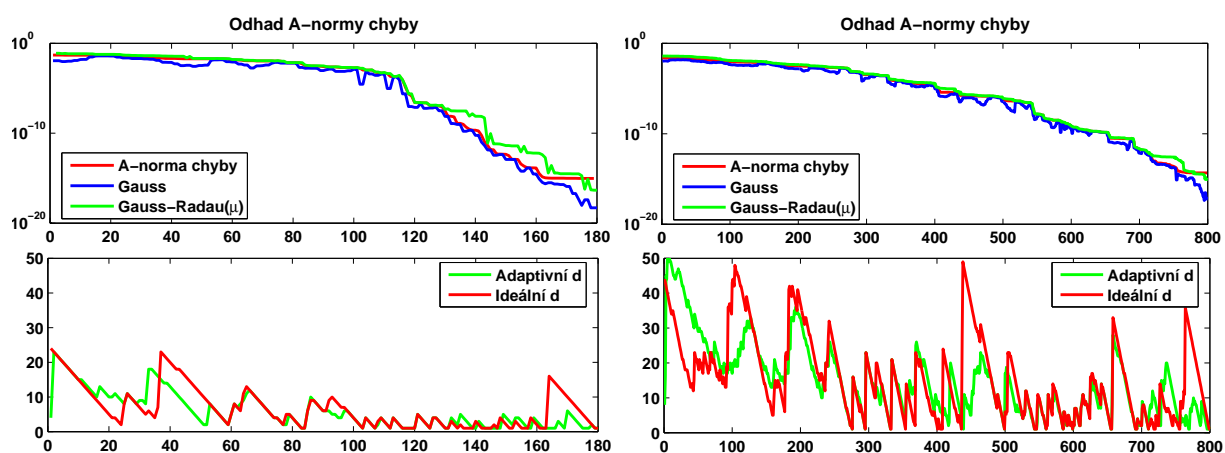
Algoritmicky tento postup můžeme interpretovat takto

Algoritmus 5: Adaptivní d _horní

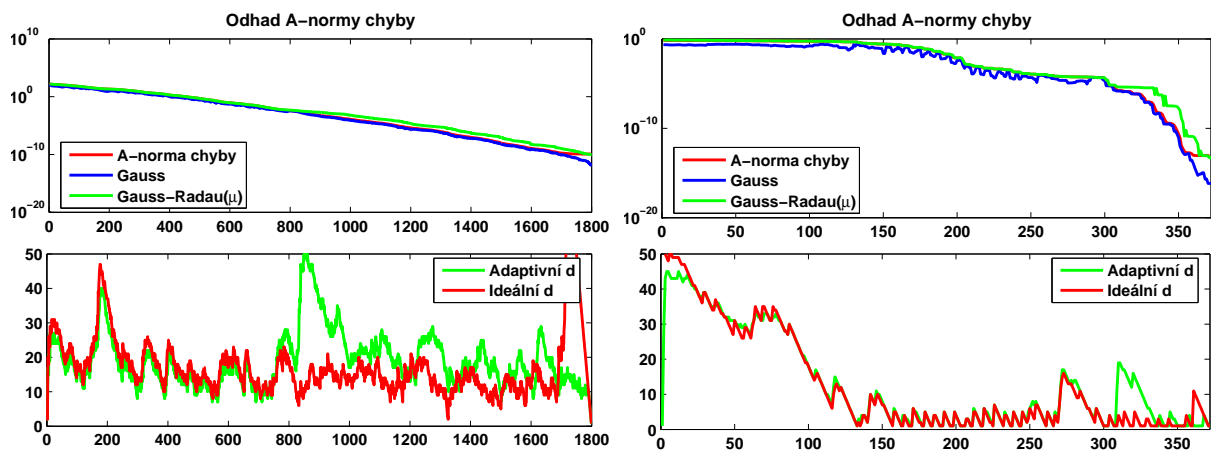
```

for  $k = 1, 2, \dots$ 
   $d_k = d$ 
  while  $\frac{g_{k+i}^{(\mu)}}{\sum_k^{k+i} g_k + g_{k+i}^{(\mu)}} > 0.8^2$ 
     $d_k = d_k + 1$ 
     $i = i + 1$ 
  end
end

```



Obrázek 5.20: Adaptivní d pomocí algoritmu 5 - matice $Bcstkt01, Bcstkt03$



Obrázek 5.21: Adaptivní d pomocí algoritmu 5 - matice $494_bus, Lund_a$

Z obrázků 5.20, 5.21. je vidět, že až do „místa nestability“ tento algoritmus ideální d dobře kopíruje. V „místě nestability“, dojde k nadhodnocení ideálního d tak, jak bychom očekávali, protože horní odhad v tomto místě klesá pomaleji než aktuální A-norma chyby, ale potom se zase téměř vrátí k hodnotě ideálního d .

2. Další možností je klást podmínku poklesu pouze na hodnoty $g_k^{(\mu)}$, které charakterizují horní odhad založený na Gauss-Radau $^{(\mu)}$ kvadratuře. Podmínka poklesu pomocí hodnot $g_k^{(\mu)}$ by se dala zapsat následujícím vztahem

$$\frac{\|x - x_{k+d}\|_{\mathbf{A}}}{\|x - x_k\|_{\mathbf{A}}} \sim \frac{\sqrt{g_{k+d}^{(\mu)}}}{\sqrt{g_k^{(\mu)}}} \ll 1,$$

což můžeme zapsat algoritmem

Algoritmus 6: Adaptivní d _podíl

for $k = 1, 2, \dots$ k -tý krok CG

$d_k = d$

while $\frac{g_{k+i}^{(\mu)}}{g_k^{(\mu)}} > 0.8^2$

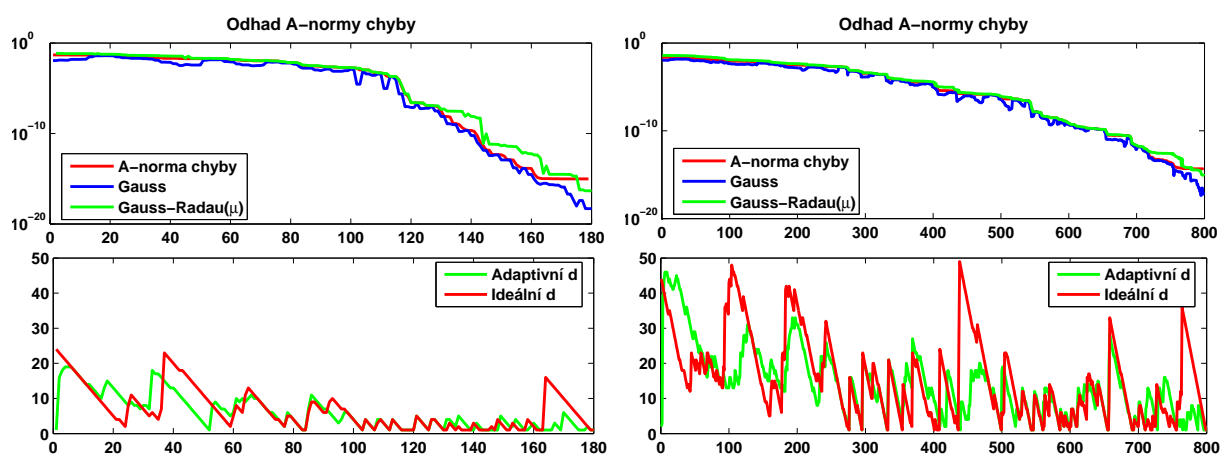
$d_k = d_k + 1$

$i = i + 1$

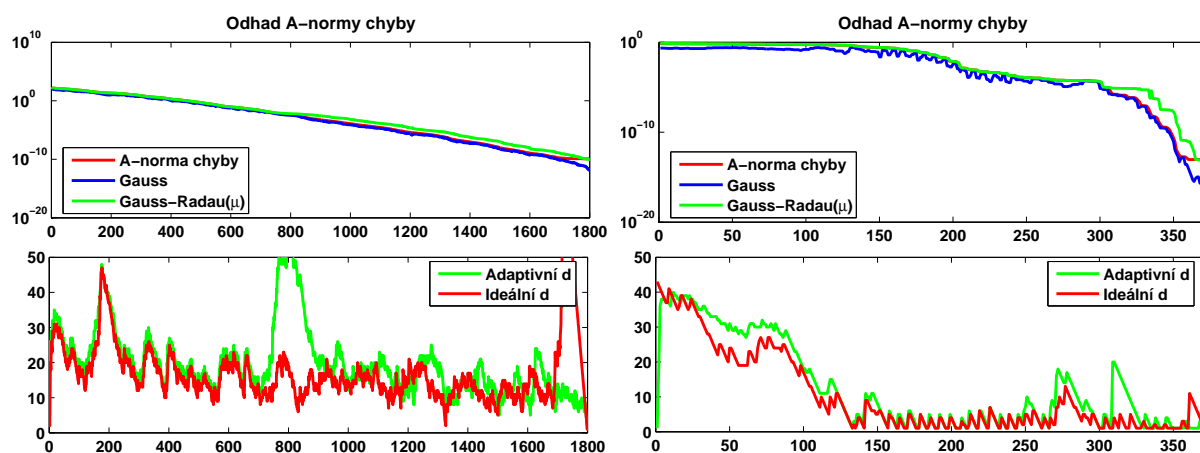
end

end

Z obrázků 5.22 a 5.23 je vidět, že takovýto postup nám až do „místa nestability“ také dobře kopíruje hodnotu ideálního parametru d , potom dochází opět k nadhodnocení, protože „místo nestability“ se ve výpočtu projeví právě v hodnotách $g_k^{(\mu)}$ a poté už adaptivní d opět dobře kopíruje ideální hodnotu.



Obrázek 5.22: Adaptivní d pomocí algoritmu 6 - matice $Bcsstk01, Bcsstk03$



Obrázek 5.23: Adaptivní d pomocí algoritmu 6 - matice $494_bus, Lund_a$

3. Poslední heuristiku, kterou jsem navrhla, je použít současně dolní a horní odhad. Ukázali jsme, že pokles $\frac{\|x - x_j\|_A}{\|x - x_k\|_A}$ můžeme odhadovat zdola. Tento pokles si omezíme i shora, což zapíšeme následovně

$$\frac{\sum_j^{j+d} g_j}{\sum_k^{k+d} g_k} \leq \frac{\|x - x_j\|_A^2}{\|x - x_k\|_A^2} \leq \frac{g_{j+1}^{(\mu)} + \sum_j^{j+d} g_j}{\sum_k^{k+d} g_k}, \quad (5.4)$$

kde $j > k$. Označíme si D dolní odhad a H horní odhad poklesu A-normy chyby mezi kroky j a k

$$D = \frac{\sum_j^{j+d} g_j}{\sum_k^{k+d} g_k}, \quad H = \frac{g_{j+1}^{(\mu)} + \sum_j^{j+d} g_j}{\sum_k^{k+d} g_k}.$$

Pokud bychom měli dobrý horní i dolní odhad, znamenalo by to, že se hodnoty D a H liší např. nejvýše o dva řády a platilo by

$$0,1 < \left| \frac{D - H}{D} \right| < 10.$$

Dosadíme-li si za D a H , můžeme tuto podmínku přepsat následovně

$$0,1 < \left| \frac{\frac{\sum_j^{j+d} g_j}{\sum_k^{k+d} g_k} - \frac{g_{j+1}^{(\mu)} + \sum_j^{j+d} g_j}{\sum_k^{k+d} g_k}}{\frac{\sum_j^{j+d} g_j}{\sum_k^{k+d} g_k}} \right| < 10$$

$$0,1 < \left| \frac{g_{j+1}^{(\mu)}}{\sum_j^{j+d} g_j} \right| < 10. \quad (5.5)$$

Podmínku 5.5 můžeme také chápat tak, že chceme, aby dolní a horní odhad byly od sebe vzdáleny v kroku j nejvýše o dva řády. To si můžeme zapsat algoritmicky

Algoritmus 7: Adaptivní d_suma

for $k = 1, 2, \dots$ k -tý krok CG

$d_k = d$

while $\left| \frac{g_{k+i}^{(\mu)}}{\sum_k^{k+i} g_k} \right| > eps$

$d_k = d_k + 1$

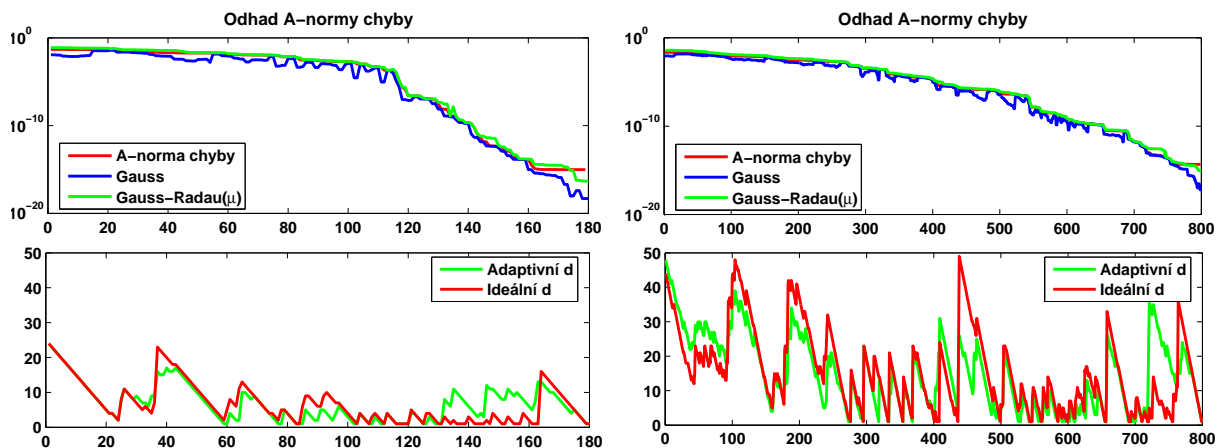
$i = i + 1$

end

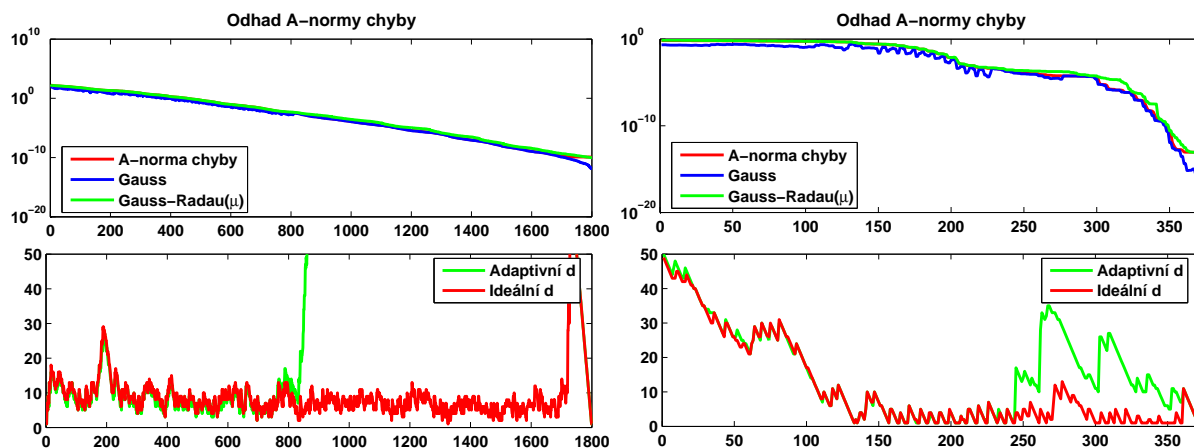
end

Konstantu eps volíme z intervalu $(0, 10)$. Já jsem ve všech experimentech používala hodnotu $eps = 5$. Na obrázcích 5.24 a 5.25 je vidět, že nám tento postup dává velmi dobrou hodnotu parametru d až do „místa nestability“. Potom dojde k nadhodnocení ideálního d a oproti dvěma předchozím algoritmům se už nezlepší. Tento jev se dá očekávat, protože algoritmus je založen na předpokladu, že

si hodnoty horního a dolního odhadu jsou velmi blízké, což od „místa nestability“ není splněno a tak tam, kde horní odhad nadhodnocuje A-normu chyby, dochází k nadhodnocování i ideálního d .



Obrázek 5.24: Adaptivní d pomocí algoritmu 7 - matice *Bcsstk01, Bcsstk03*



Obrázek 5.25: Adaptivní d pomocí algoritmu 7 - matice *494_bus, Lund_a*

Nejllepší výsledky adaptivní volby parametru d nám dával až do „místa nestability“ algoritmus 7. Adaptivní d_{suma} , protože velmi dobře kopíruje i místa, ve kterých dochází k velkému nárůstu ideálního parametru d . Ale dále oproti dvěma zbývajícím algoritmům značně nadhodnocuje ideální d tam, kde nemáme dobrý horní odhad. Zbývající dva algoritmy 5. Adaptivní $d_{\text{horní}}$ a 6. Adaptivní $d_{\text{podíl}}$ sice nekopírují tak dobře tvar ideálního d , ale protože jsou založeny na rozpoznávání stagnace A-normy chyby, za „místem nestability“ dávají opět dobré výsledky. Z těchto dvou algoritmů bych řekla, že je ještě o něco lepší algoritmus 5. Adaptivní $d_{\text{horní}}$, protože o trochu

lépe kopíruje ideální hodnotu parametru d . Z uvedených obrázků je zřejmé, že „místo nestability“ je ve všech třech algoritmech velmi problematické.

5.4 Použité matice

Matice, které jsem používala k experimentům, jsem získala z internetové stránky MATRIX MARKET³ a THE UNIVERSITY OF FLORIDA SPARSE MATRIX COLLECTION⁴
Seznam matic:

| Název | Rozměr | λ_1 (Matlab2010) |
|--------------|--------------------|--------------------------|
| Bcsstk01.mtx | 48×48 | 3417.26756 |
| Bcsstk03.mtx | 112×112 | 29410.20464 |
| Bcsstk06.mtx | 420×420 | 460.62460 |
| Nos1.mtx | 237×237 | 123.35420 |
| Nos6.mtx | 675×675 | 1.00002 |
| 494_bus.mtx | 494×494 | 0.012423 |
| 662_bus.mtx | 662×662 | 0.0054667 |
| 1138_bus.mtx | 1138×1138 | 0.0035163 |
| Mesh1e1.mtx | 48×48 | 1.74006 |
| Mesh1em6.mtx | 48×48 | 1.17985 |
| Nasa1824.mtx | 1824×1824 | 11.19057 |
| Lund_a.mtx | 147×147 | 80.03513 |
| Strakos.mat | $n \times n$ | a |

Matice Strakos.mat má vlastní čísla, která leží v intervalu $[a, b]$ a jsou soustředěna u a . Pokud zvolíme větší hodnotu parametru ρ jsou více rozptýlena viz např. [11]

$$\begin{aligned} & \text{Strakos}(n, a, b, \rho) \\ & \lambda_1 = a, \quad \lambda_n = b \\ & \lambda_i = \lambda_1 + \frac{i-1}{n-1}(\lambda_n - \lambda_1)\rho^{n-i}, \quad i = 2, \dots, n \end{aligned}$$

³<http://math.nist.gov/MatrixMarket>

⁴<http://www.cise.ufl.edu/research/sparse/matrices>.

Kapitola 6

Závěr

Ve své práci jsme se seznámila s metodou sdružených gradientů z jiného pohledu, než jak jsem se s ní setkávala doposud během studia, odvozenou pomocí minimalizace kvadratického funkcionálu, ale seznámila jsem se s ní jako s metodou, na kterou mohu pohlížet jako na Gaussovu kvadraturu. Souvislosti CG s Gaussovou kvadraturou mi pak umožnily počítat jak dolní tak i horní odhad A-normy chyby v metodě sdružených gradientů.

V experimentální části jsem nejprve uvedla příklady všech dolních i horních odhadů, které jsme si odvodili v teoretické části. Experimentálně jsem ukázala, že problém dolního odhadu založeného na Gaussově kvadratuře spočívá v tom, že pokud přesná A-norma chyby v d krocích stagnuje, dolní odhad podhodnocuje A-normu chyby. Úskalí horního odhadu založeného na Gauss-Radau kvadratuře je, že je velmi citlivý na předepsaný uzel μ . Pro některé matice malá změna vstupního parametru μ způsobila zásadní změnu celého odhadu. Pokud pro danou matici A dochází ke zpoždění konvergence metody sdružených gradientů, je horní odhad nestabilní, neboli dochází v nějakém místě k výraznému zhoršení doposud velmi přesného odhadu. „Místo nestability“ je možné rozpoznat pomocí vlastních čísel modifikované Jacobiho matice nebo pomocí prvků \tilde{d}_{k+1} LDL^T rozkladu modifikované Jacobiho matice, což dává dobré výsledky a navíc prvky \tilde{d}_{k+1} můžeme snadno získávat v průběhu CG.

Nakonec jsem se pokusila navrhnout tři způsoby, jak je možné adaptivně volit parametr d . Ve všech třech případech jsem k volbě používala horní odhad založený na Gauss-Radau kvadratuře. Algoritmy velmi dobře kopírují teoretickou hodnotu ideálního parametru d až do „místa nestability“. Protože horní odhad klesá v „místě nestability“ mnohem pomaleji a dochází k nadhodnocení A-normy chyby, v tomto místě dochází k nadhodnocování ideálního d . Otevřenou otázkou v adaptivní volbě parametru d tedy např. zůstává, jestli volit adaptivní parametr d v celém výpočtu pomocí horního odhadu a soustředit se na vylepšení odhadu v „místě nestability“ nebo se snažit detekovat „místo nestability“ a od toho místa použít jinou heuristiku volby adaptivního d založenou např. pouze na dolním odhadu.

Literatura

- [1] Arioli, M.: A stopping criterion for the conjugate gradient algorithms in a finite element method framework. *Numer. Math.*, ročník 97, č. 1, 2004: s. 1–24.
- [2] Duintjer Tebbens, J.; Hnětynková, I.; Plešinger, M.; aj.: *Analýza metod pro maticové výpočty (Základní metody)*. Charles University, Prague: Matfyzpress, 2012, ISBN 978-80-7378-201-6, 328 s.
- [3] Gautschi, W.: *Orthogonal polynomials: computation and approximation*. Numerical Mathematics and Scientific Computation, New York: Oxford University Press, 2004, x+301 s., oxford Science Publications.
- [4] Golub, G. H.; Meurant, G.: Matrices, moments and quadrature. In *Numerical analysis 1993 (Dundee, 1993)*, *Pitman Res. Notes Math. Ser.*, ročník 303, Harlow: Longman Sci. Tech., 1994, s. 105–156.
- [5] Golub, G. H.; Meurant, G.: Matrices, moments and quadrature. II. How to compute the norm of the error in iterative methods. *BIT*, ročník 37, č. 3, 1997: s. 687–705, ISSN 0006-3835, direct methods, linear algebra in optimization, iterative methods (Toulouse, 1995/1996).
- [6] Golub, G. H.; Meurant, G.: *Matrices, Moments and Quadrature With Applications*. USA: Princeton University Press, 2010, xxx+698 s.
- [7] Golub, G. H.; Strakoš, Z.: Estimates in quadratic formulas. *Numer. Algorithms*, ročník 8, č. 2-4, 1994: s. 241–268, ISSN 1017-1398.
- [8] Hestenes, M. R.; Stiefel, E.: Methods of conjugate gradients for solving linear systems. *J. Research Nat. Bur. Standards*, ročník 49, 1952: s. 409–436 (1953).
- [9] Meurant, G.: Estimates of the l_2 norm of the error in the conjugate gradient algorithm. *Numer. Algorithms*, ročník 40, č. 2, 2005: s. 157–169.
- [10] Meurant, G.; Tichý, P.: On computing quadrature-based bounds for the A -norm of the error in conjugate gradients. *submitted to Numerical Algorithms*, minor revision, 2012.

- [11] Strakoš, Z.; Tichý, P.: On error estimation in the conjugate gradient method and why it works in finite precision computations. *Electron. Trans. Numer. Anal.*, ročník 13, 2002: s. 56–80 (electronic), ISSN 1068-9613.
- [12] Strakoš, Z.; Tichý, P.: Error estimation in preconditioned conjugate gradients. *BIT*, ročník 45, č. 4, 2005: s. 789–817.

Příloha A

Ovládání programu

Pro všechny experimentální výpočty jsem sestavila v Matlabu program, který je možné najít na příloženém CD. Program se dá spustit pomocí dialogového okna zapsáním „CG“ do příkazového řádku.

Tím se otevře dialogové okno, ve kterém je možné zadávat a volit tyto položky:

- V seznamu matic si zvolíme tu, pro kterou chceme provádět výpočty.
- V seznamu experimentů si zvolíme, jaký chceme realizovat. Jedná se o následující možnosti:
 - **Odhad** - funkce spočítá a zobrazí námi zvolené odhady pro danou matici
 - **Relativní chyba** - horní odhad a relativní chyba odhadu
 - **Dvě mi** - horní odhad s přepnutím μ_2
 - **Odhad okolo mi** - počítá hodnoty horních odhadů pro volby $\mu \in (0, \lambda_3)$ v pevné iteraci k
 - **Ideální d** - vykresluje ideální hodnotu parametru d
 - **Odhad dopředu** - experiment pro počítání „odhad dopředu“
 - **Adaptivní d_horní** - adaptivní d algoritmem 5. Adaptivní d_horní
 - **Adaptivní d_podíl** - adaptivní d algoritmem 6. Adaptivní d_podíl
 - **Adaptivní d_suma** - adaptivní d algoritmem 7. Adaptivní d_suma
 - **Vlastní čísla** - vykresluje dvě nejmenší vlastní čísla Jacobiho matice
- Po zvolení konkrétní matice se v editačním políčku μ a η zobrazí hodnota nejmenšího a největšího vlastního čísla. Tyto hodnoty můžeme zvětšovat resp. zmenšovat a tím např. pozorovat, jak se mění jednotlivé odhady v závislosti na volbě těchto parametrů.
- **new** zvolíme algoritmus 3.CGQ

- **old** zvolíme algoritmus 4.CGQL
- Můžeme zvolit výpočet a vykreslení jednotlivých odhadů založených na různých kvadraturních pravidlech:
 - **Gauss** - počítá dolní odhad pomocí Gaussovy kvadratury
 - **Gauss-Radau(a)** - počítá horní odhad pomocí Gauss-Radau^(μ) kvadratury
 - **Gauss-Radau(b)** - počítá dolní odhad pomocí Gauss-Radau^(η) kvadratury
 - **Gauss-Lobatto** - počítá horní odhad pomocí Gauss-Lobatto kvadratury

Celý výpočet se po zvolení všech parametrů spustí tlačítkem **Goooo;-)** na spodní části panelu.

Pokud nechceme počítat pro stejnou matici znovu CG ale pouze „kvadraturní část“, např. pokud pouze změníme hodnotu μ , stačí použít tlačítko **Update**.