Prohlašuji, že jsem svoji diplomovou práci zpracoval samostatně s použitím uvedené literatury. Souhlasím se zapůjčováním práce.

V Praze dne 1.4.1997 ....

MATEMATICKO-FYZIKÁLNÍ FAKULTA

UNIVERZITY KARLOVY

Diplomová práce

Chování BiCG a CGS algoritmů

P e t r T i c h ý

KATEDRA NUMERICKÉ MATEMATIKY

Vedoucí diplomové práce: Doc. RNDr. Jan Zítko, CSc.

Obor: Matematika

Zaměření: Výpočtová matematika

Rád bych poděkoval vedoucímu diplomové práce Doc. J. Zítkovi za jeho cenné rady a připomínky, které mi v průběhu celé práce a při úpravě textu poskytoval.

Obsah

Úvod

Jedním ze základních úkolů numerické lineární algebry je rychlé nalezení řešení obsáhlého systému lineárních algebraických rovnic. Tyto systémy lineárních rovnic vznikají například při aplikaci metody konečných prvků nebo metody sítí na systém parciálních diferenciálních rovnic. Matice těchto soustav lineárních rovnic jsou obvykle řídké. Použití přímé metody na takovouto soustavu často ztroskotává kvůli velkým nárokům na pamět počítače. Potom jsou alternativou blokové iterační postupy nebo nějaká verze zobecněné metody sdružených gradientů (CG).

Je-li matice soustavy symetrická, pozitivně definitní, je jednou z možností, jak tuto soustavu řešit, použít metody sdružených gradientů (CGM). Je-li však matice soustavy pouze regulární (obecně nesymetrická), je nutno použít jiných metod např. metodu bikonjugovaných gradientů (BCG), která je zobecněním CGM v tom smyslu, že pro matice symetrické, pozitivně definitní generuje při vhodné volbě počátečních nastavení stejnou posloupnost vektorů x_k, konvergujících k přesnému řešení. Metoda bikonjugovaných gradientů je náplní této diplomové práce.

Uvažujme soustavu

$$Ax = b,$$

kde A regulární, obecně nesymetrická, $A \in L(\mathbb{R}^n)$, $b \in \mathbb{R}^n$, $x \in \mathbb{R}^n$.

První kapitola je věnována metodě sdružených gradientů (CGM), která je speciálním případem BCG s volbou A symetrická, pozitivně definitní a $\widetilde{r}_0 = r_0$. První podkapitola pojednává o odvození a formulaci CGM, přičemž jsou zde do podrobnosti rozebrány některé důkazy, které jsou v knihách často vynechávány. V další podkapitole jsme se pokusili podívat se na iterační proces metody CGM z geometrického hlediska. Zjistili jsme, že další aproximace řešení je vektor, jehož koncový bod dostaneme jako bod dotyku jistého elipsoidu a jisté nadroviny. Umocnění metody se provádí obvykle u metody BCG. V podkapitole třetí jsme umocnili i metodu CGM (CGMS) s nadějí, že dostaneme metodu s lepší konvergencí, než CGM. V následující podkapitole jsme provedli další, v literatuře často vynechávaný důkaz. Jde o odhad normy vektoru chyby metody CGM. Jak jsme též ukázali, lze poměrně jednoduchým způsobem odvodit z odhadu normy vektoru chyby metody CGM i odhad pro metodu CGMS. Ukáže se, že umocnění sice přineslo zlepšení odhadu normy vektoru chyby, ale zároveň vzrosla početní náročnost algoritmu. Na závěr první kapitoly jsme zformulovali algoritmy předpomíněné CGM a předpodmíněné CGMS.

Některé postupy v kapitole první jsme zobecnili v kapitole druhé, která pojednává o metodě BCG. To, že BCG je interpretací Lanczosova ortogonalizačního procesu (LOP) je všeobecně známé. Není však zřejmé, jak odvodit z LOP metodu BCG. Rozhodli jsme se, že se o toto odvození pokusíme a skutečně se nám to podařilo. Je v něm dobře vidět, jaké požadavky musíme klást, abychom mohli algoritmus BCG zformulovat. Při tomto odvození si lze také dobře uvědomit pojmy *skutečného* a *lanczosova* zhroucení. Zatímco v LOP nemůže lanczosovo zhroucení nastat, je právě nesplnění oněch požadavků nutných ke zformulování BCG příčinou jejího lanczosova zhroucení. V další podkapitole jsme zobecnili umocnění CGM a dostali jsme celkem bez problémů algoritmus umocněné BCG (CGS). Následovné předpodmínění obou metod je taktéž pouze obecnějším přepisem předpodmínění CGM a CGMS. O metodě CGS je známé, že normy reziduí neklesají obecně monotónně. Jak docílit monotonie jsme ukázali v podkapitole čtvrté a páté, kde jsou zformulovány algoritmy jednoduše zhlazené CGS (SSCGS) a dvojnásobně zhlazené CGS (DSCGS).

Třetí kapitola je věnována případům, kdy dojde ke zhroucení Lanczosova ortogonalizačního procesu. Zdroj, ze kterého jsme čerpali je článek [Brezn–94], který je však psán méně přehledným způsobem. Pokusili jsme se o srozumitelnější podání dané teorie. Dokázali jsme dosti důležitou větu 3.1, jež autor článku přechází několika slovy a která je podstatná pro zjišťování skutečného zhroucení LOP. Zároveň jsme dokázali lemmata 3.10 a 3.11, na kterých stojí celá teorie o vyhnutí se zhroucení a jež byla v článku opět jen lehce zmíněna. Základní myšlenkou, jak se vyhnou zhroucení, je sestrojovat pouze ta rezidua r_k a aproximace x_k, při jejichž výpočtu nedochází ke zhroucení. Na základě teorie ortogonálních polynomů a rekurentních vztahů, jež pro dané polynomy platí, je možné setrojovat pouze některé polynomy $\mathcal{P}_k$ ($r_k = \mathcal{P}_k(A)r_0$). Je poměrně početně náročné odvodit, jak spočíst další polynom $\mathcal{P}_{k+i}$, což je dobře vidět v podkapitole třetí. Na základě těchto poznatků si jednak ukážeme, jak se vyhnout zhroucení u metody CGS a jednak zformulujeme metodu, která je interpretací metody CGS, ale používá jiné rekurentní vztahy. Tuto metodu jsme nazvali CGS_1. Na závěr třetí kapitoly jsme se zabývali tím, kdy dochází ke skutečnému a lanczosovu zhroucení metody CGS a zformulovali jsme algoritmus CGS bez skoro-zhroucení.

Ve čtvrté kapitole jsme si definovali zobecněnou metodu sdružených gradientů (CG). Tato metoda určuje jistou třídu metod na řešení soustavy lineárních rovnic. Naším úkolem v této kapitole bylo ukázat, že metoda BCG leží v této třídě metod. Nejdříve jsme si odvodili, za jakých předpokladů generuje Lanczosova metoda stejné vektory r_k a x_k jako metoda BCG. Lanczosova metoda, jež je interpretací BCG, nám poté posloužila při dalších důkazech. Základním problémem tedy bylo, jak zvolit matici Z v definici metody CG, aby poté metoda CG generovala stejné vektory r_k a x_k jako metoda BCG. V [Weiss–95] byla tato matice volena jako $Z = W^*(D^*)^{-T}\Omega^T\Omega(D^*)^{-1}(W^*)^TA$, kde matice W^*, D^* jsou matice z konjugované Lanczosovy metody a Ω je libovolná diagonální matice. Nepodařilo se nám však dokázat, že by při této volbě generovala CG stejné vektory jako BCG. Po úvaze jsme však našli vhodnou matici $Z = WD^{-1}W^T$, kde matice W a D jsou matice z Lanczosovy metody. Druhým způsobem, jak volit matici Z se zabýváme v závěrečné podkapitole. Tato druhá volba je dobře známa z literatury, avšak důkaz, že při této volbě generuje CG vektory r_k a x_k jako BCG, je v literatuře opět vynecháván. Bylo tedy na nás, abychom tuto skutečnost dokázali.

Kapitola pátá se zabývá numerickými experimenty s metodami, které jsou popsány v této diplomové práci. Zjistili jsme, že umocněním metody CGM jsme sice dostali metodu, která konverguje rychleji k přesnému řešení (co se počtu iterací týče), avšak v časovém srovnání vyhrává jasně metoda CGM. V dalších podkapitolách jsme graficky znázornili různá porovnání metod BCG, CGS, SSCGS a DSCGS. V poslední podkapitole jsme se zabývali testy metody BSMRZS. Zjistili jsme, že tato metoda není vhodná na řešení velkých řídkých soustav a že je spíše zajímavým teoretickým pokusem.

Závěrečná šestá kapitola obsahuje popisy programů jež jsme vytvořili (BCG, CGS) a programu, který jsme upravili (BSMRZS).

Kapitola 1

Metoda sdružených gradientů

Metoda sdružených gradientů je iterační metoda na řešení soustav lineárních rovnic se symetrickou, pozitivně definitní maticí. Jak uvidíme později, je specielním případem metody obecnější, řešící soustavy lineárních rovnic s libovolnou regulární maticí, a tou je metoda bikonjugovaných gradientů.

1.1 Odvození metody sdružených gradientů

Nechť $x \in \mathbb{R}^n, b \in \mathbb{R}^n$ a $A \in L(\mathbb{R}^n)$ je symetrická, pozitivně definitní matice. Mějme soustavu lineárních rovnic

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

a funkcionál

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

Jelikož $\phi'(x) = x^T A - b^T$, $\phi''(x) = A$ a A je symetrická a pozitivně definitní, je úloha minimalizace funkcionálu (1.2) ekvivalentní úloze řešit soustavu rovnic (1.1). Na řešení soustavy (1.1) lze tedy použít minimalizační strategii aplikovanou na funkcionál $\phi(x)$.

Proces minimalizace

Definice 1.1 : Nechť x_0 je počáteční aproximace řešení, x_k je již spočtená k-tá aproximace minima funkcionálu $\phi(x)$ a nechť p_k je vhodně zvolený směr. Číslo α_k je takové číslo, pro které platí

$$\phi(x_k + \alpha_k p_k) = \min_{\alpha} \phi(x_k + \alpha p_k). \tag{1.3}$$

Další aproximaci řešení x_{k+1} definujeme jako $x_{k+1} = x_k + \alpha_k p_k$. Vektor $r_k = b - Ax_k$ budeme nazývat $k-$tým reziduem soustavy (1.1), x^* označuje přesné řešení (1.1).

Lemma 1.1 : Platí

$$\alpha_k = \frac{p_k^T r_k}{p_k^T A p_k}, \tag{1.4}$$

$$\phi(x_k + \alpha_k p_k) = \phi(x_k) - \frac{1}{2}\frac{(p_k^T r_k)^2}{p_k^T A p_k}. \tag{1.5}$$

Důkaz : Definujme

$$g(\alpha) = \phi(x_k + \alpha p_k) = \frac{1}{2}(x_k + \alpha p_k)^T A(x_k + \alpha p_k) - (x_k + \alpha p_k)^T b =$$

$$= \frac{1}{2}x_k^T A x_k + \alpha p_k^T A x_k + \frac{1}{2}\alpha^2 p_k^T A p_k - x_k^T b - \alpha p_k^T b.$$

Problém nalezení α_k je ekvivalentní problému nalezení minima kvadratické funkce $g(\alpha)$. Platí

$$\frac{\partial g}{\partial \alpha} = p_k^T A x_k - p_k^T b + \alpha p_k^T A p_k = -p_k^T r_k + \alpha p_k^T A p_k,$$

$$\frac{\partial g}{\partial \alpha} = 0 \Longleftrightarrow p_k^T r_k = \alpha_k p_k^T A p_k.$$

Vztah (1.5) se dokáže výpočtem dosazením α_k do $\phi(x)$. ■

Podmínkou pro to, aby hodnota funkcionálu $\phi(x)$ klesala, je

$$p_k^T r_k \neq 0, \tag{1.6}$$

což je vidět ze vztahu (1.5) . Právě popsanou minimalizaci funkcionálu $\phi(x)$ můžeme vyjádřit následujícím algoritmem.

Algoritmus 1.1

Input $x_0,\ A,\ b$;
for $k = 0$ **to** $\ldots$ **do**
  begin
    $r_k = b - Ax_k$;
    if $r_k = 0$ **then** $x^* = x_k$ *and exit*
    else
      begin
        Choose direction of p_k *such that* $p_k^T r_k \neq 0$;
        $\alpha_k = \frac{p_k^T r_k}{p_k^T A p_k}$;
        $x_{k+1} = x_k + \alpha_k p_k$;
      end;
  end.

Lemma 1.2 : Pro vektor x_{k+1} platí

$$x_{k+1} = x_0 + \alpha_0 p_0 + .. + \alpha_k p_k.$$

Důkaz : Plyne přímo z definice x_k. ■

Volba vektorů p_k

Problémem tedy zůstává, jak volit vektory p_k. Ideální by bylo volit p_k lineárně nezávislé s vlastností

$$\phi(x_{k+1}) = \min_{x \in x_0 + sp\{p_0,..,p_k\}} \phi(x), \tag{1.7}$$

což lze též psát jako

$$\phi(x_{k+1}) = \min_{y \in sp\{p_0,..,p_k\}} \phi(x_0 + y).$$

To by nám zajišťovalo konečnost počtu iterací. Po nejvýše n krocích by totiž x_n minimalizovalo funkcionál (1.2) přes celé $\mathbb{R}^n$, tedy by bylo řešením soustavy (1.1). Ukážeme, že vektory p_k lze volit tak, aby řešení jednodimenzionálního problému (1.3) bylo zároveň řešením k-dimenzionálního problému (1.7).

Definice 1.2 : Označme symbolem P_k matici $[p_0, p_1, .., p_{k-1}]$, $\mathcal{S}_k = sp\{p_0, \dots, p_{k-1}\}$ prostor směrových vektorů, $\mathcal{R}_k = sp\{r_0, \dots, r_{k-1}\}$ prostor reziduí a $\mathcal{K}_k$ Krylovův prostor určený maticí A a vektorem r_0, $\mathcal{K}_k = sp\{r_0, Ar_0, \dots, A^{k-1}r_0\}$.

Poznámka : $P_k \in L(\mathbb{R}^k, \mathbb{R}^n)$

Věta 1.1 : *Nechť x_k minimalizuje $\phi(x)$ na množině $x_0 + sp\{p_0, \dots, p_{k-1}\}$. Volíme-li vektor p_k tak, že splňuje podmínku*

$$p_k^T A p_i = 0 \quad \text{pro} \quad i = 0, .., k-1, \tag{1.8}$$

potom vektor $x_{k+1} = x_k + \alpha_k p_k$, kde α_k je určeno vztahem (1.4), minimalizuje funkcionál $\phi(x)$ přes $x_0 + sp\{p_0, \dots, p_k\}$.

Důkaz : Je-li $x \in x_0 + sp\{p_0, \dots, p_k\}$, můžeme x psát ve tvaru

$$x = x_0 + P_k y + \lambda p_k,$$

kde $y \in \mathbb{R}^k$ a $\lambda \in \mathbb{R}$. Dosadíme-li x do (1.2), dostaneme vyjádření

$$\phi(x) = \phi(x_0 + P_k y) + \lambda y^T P_k^T A p_k + \frac{\lambda^2}{2} p_k^T A p_k - \lambda p_k^T r_0. \tag{1.9}$$

Označme

$$g(\lambda) = \frac{\lambda^2}{2} p_k^T A p_k - \lambda p_k^T r_0.$$

Protože $\lambda y^T P_k^T A p_k = 0$, můžeme (1.9) přepsat ve tvaru

$$\phi(x) = \phi(x_0 + P_k y) + g(\lambda),$$

tedy

$$\min_{x \in x_0 + \mathcal{S}_{k+1}} \phi(x) = \min_{y,\lambda} \phi(x_0 + P_k y + \lambda p_k) = \min_y \phi(x_0 + P_k y) + \min_\lambda g(\lambda). \tag{1.10}$$

Zbývá již jen zjistit, pro jaké λ a y nabývá výraz na pravé straně minima. Vyjádříme-li si x_k jako $x_k = x_0 + P_k y_k$, můžeme psát

$$\phi(x_0 + P_k y_k) = \phi(x_k) = \min_y \phi(x_0 + P_k y).$$

Funkce $g(\lambda)$ nabývá minima pro

$$\lambda_k = \frac{p_k^T r_0}{p_k^T A p_k}.$$

Z předchozích dvou vztahů plyne, že výraz (1.10) nabývá minima pro $\lambda = \lambda_k$ a $y = y_k$. Platí tedy

$$\phi(x_k + \lambda_k p_k) = \phi(x_0 + P_k y_k + \lambda_k p_k) = \min_{y,\lambda} \phi(x_0 + P_k y + \lambda p_k).$$

Poznamenejme ještě, že

$$x_k - x_0 \in sp\{p_0, .., p_{k-1}\} \Rightarrow p_k^T A(x_k - x_0) = 0 \Rightarrow p_k^T b - p_k^T A x_k - (p_k^T b - p_k^T A x_0) = 0$$

a odtud $p_k^T r_k = p_k^T r_0$, neboli $\alpha_k = \lambda_k$.
Platí tedy

$$\phi(x_{k+1}) = \phi(x_k + \alpha_k p_k) = \min_{x \in x_0 + \mathcal{S}_{k+1}} \phi(x). \qquad \blacksquare$$

Lemma 1.3 : *Nechť vektory p_i pro $i = 0, \ldots, k-1$ splňují podmínku (1.8) . Je-li $p^T r_k = 0$ pro všechny vektory $p \in sp\{Ap_0, .., Ap_{k-1}\}^\perp$, potom $x_k = x^*$.*

Důkaz : Platí $r_k = b - Ax_k = b - Ax_0 - AP_k z$ pro nějaké $z \in \mathbb{R}^k$. Dle předpokladů lemmatu je

$$r_k \perp sp\{Ap_0, \ldots, Ap_{k-1}\}^\perp \quad \text{neboli} \quad b - Ax_0 - AP_k z \perp sp\{Ap_0, \ldots, Ap_{k-1}\}^\perp,$$

aneb

$$b - Ax_0 - AP_k z \in sp\{Ap_0, \ldots, Ap_{k-1}\} \Rightarrow b \in Ax_0 + sp\{Ap_0, \ldots, Ap_{k-1}\} \Rightarrow$$

$$\Rightarrow x^* = A^{-1} b \in x_0 + sp\{p_0, .., p_{k-1}\}. \qquad \blacksquare$$

Z předchozí úvahy plyne, že v k-tém kroku je buď $r_k = 0$ a potom máme přesné řešení x_k, nebo $r_k \neq 0$ a potom existuje vektor $p \in sp\{Ap_0, .., Ap_{k-1}\}^\perp$ takový, že $p^T r_k \neq 0$. Vektor p_k se budeme snažit volit tak, aby úbytek funkcionálu $\phi(x)$

$$\phi(x_k) - \phi(x_{k+1}) = \frac{1}{2} \frac{(p_k^T r_k)^2}{p_k^T A p_k} \qquad (1.11)$$

byl co možná největší. Z (1.11) je vidět, že úbytek nezávisí na délce vektoru p_k, ale pouze na směru. Asi nejlépe je volit p_k tak, že vektory p_k a r_k svírají co nejmenší úhel neboli $(p_k^T r_k)^2$ je maximální. Vektorem, který splňuje uvedené podmínky, leží tedy v prostoru $sp\{Ap_0, .., Ap_{k-1}\}^\perp$ a svírá s r_k nejmenší úhel, je projekce vektoru r_k do prostoru $sp\{Ap_0, .., Ap_{k-1}\}^\perp$.

Lemma 1.4 : *Definujeme-li vektor p_k jako projekci vektoru r_k do prostoru $sp\{Ap_0, \ldots, Ap_{k-1}\}^\perp$, lze vektor p_k psát ve tvaru*

$$p_k = (I - (AP_k)(AP_k)^+) r_k, \qquad (1.12)$$

kde matice $(AP_k)^+$ je Moor-Penroseova zobecněná inverzní matice.

Důkaz : $\mathbb{R}^n$ rozložíme na podprostory $sp\{Ap_0, .., Ap_{k-1}\}^\perp$ a $sp\{Ap_0, .., Ap_{k-1}\}$. Projekci do prvního z uvedených označme p_k, projekci do druhého z uvedených označme s_k. Mezi oběmi projekcemi zřejmě platí vztah $p_k + s_k = r_k$. Projekci s_k můžeme napsat jako

$$s_k = z_1 A p_0 + z_2 A p_1 + .. + z_k A p_{k-1},$$

což lze vyjádřit jako $s_k = AP_k v_k$ kde vektor $v_k = (z_1, .., z_k)^T$. Vektor p_k lze pomocí tohoto přepisu s_k vyjádřit jako $p_k = r_k - AP_k v_k$. Z definice projekce musí vektor v_k splňovat podmínku

$$\| r_k - AP_k v_k \| = \min_{v \in \mathbb{R}^k} \| r_k - AP_k v \| .$$

Takovéto v_k je řešením soustavy normálních rovnic, tedy

$$v_k = (AP_k)^+ r_k$$

a z toho

$$p_k = (I - (AP_k)(AP_k)^+) r_k. \qquad \blacksquare$$

Formulace rekurentních vztahů

V dalším předpokládejme, že $p_0 = r_0$ a vektory p_k pro $k \geq 1$ jsou voleny tak, že jsou násobkem projekce vektoru r_k do prostoru $sp\{Ap_0, .., Ap_{k-1}\}^\perp$.

Lemma 1.5 : *Nechť je dán vektor x_0. Potom platí*

$$\begin{aligned} &(1.13) \qquad & r_{k+1} &= r_k - \alpha_k A p_k, \\ &(1.14) \qquad & p_i^T r_k &= 0 \quad \text{pro} \quad i = 0 \ldots k-1. \end{aligned}$$

Důkaz : Přenásobíme-li vztah $x_{k+1} = x_k + \alpha_k p_k$ maticí $(-A)$ zleva a přičteme-li k obou stranám vektor b dostáváme $b - Ax_{k+1} = b - Ax_k - \alpha_k A p_k$ neboli $r_{k+1} = r_k - \alpha_k A p_k$.

Pro $p_i^T r_k$ $(i = 0, \ldots, k-1)$ platí

$$p_i^T r_k = p_i^T (b - Ax_k) = p_i^T (b - Ax_0 - \alpha_0 A p_0 - \ldots - \alpha_{k-1} A p_{k-1}) =$$

$$= p_i^T r_0 - \alpha_i p_i^T A p_i = p_i^T r_0 - \frac{p_i^T r_0}{p_i^T A p_i} p_i^T A p_i = 0. \qquad \blacksquare$$

Lemma 1.6 : *Nechť rezidua r_i pro $i = 0, \ldots, k$ jsou nenulová. Potom vektory p_i pro $i = 0, \ldots, k$ a vektory r_i pro $i = 0, \ldots, k$ jsou lineárně nezávislé.*

Důkaz : Platí implikace

$$\sum_{i=0}^{k} \beta_i p_i = 0 \Rightarrow \sum_{i=0}^{k} \beta_i A p_i = 0 \Rightarrow \sum_{i=0}^{k} \beta_i p_j^T A p_i = 0 \quad \text{pro} \quad j = 0, \ldots, k,$$

ale $p_j^T A p_i = 0$ pro $i \neq j \Rightarrow \beta_j p_j^T A p_j = 0$ pro $j = 0, \ldots, k$. Protože matice A je pozitivně definitní, je $\beta_j = 0$ pro $j = 0 \ldots k$. Dále platí

$$\sum_{i=0}^{k} \beta_i r_i = 0 \Rightarrow \sum_{i=0}^{k} \beta_i p_j r_i = 0 \Rightarrow \beta_j p_j^T r_j = 0 \text{ pro } \quad j = 0, \ldots, k.$$

Protože $r_j \neq 0$ je dle lemmatu 1.3 $\beta_j = 0$ pro $j = 0, \ldots, k$. $\blacksquare$

Lemma 1.7 : Nechť $r_0, \ldots, r_k$ jsou nenulové, lineárně nezávislé vektory. Potom platí

(1.15) $sp\{p_0, .., p_k\} = sp\{r_0, .., r_k\} = sp\{r_0, Ar_0, .., A^k r_0\}$ $\quad (\mathcal{S}_{k+1} = \mathcal{R}_{k+1} = \mathcal{K}_{k+1})$.

Důkaz : Vztah budeme dokazovat indukcí. Pro $k = 0$ vztah zřejmě platí. Nechť vztah (1.15) platí pro k. Dokážeme, že platí i pro $k+1$. Ze vztahu $r_{k+1} = r_k - \alpha_k A p_k$ a z toho, že $r_k, p_k \in \mathcal{K}_{k+1}$ plyne, že $r_{k+1} \in \mathcal{K}_{k+2}$. Dále ze vztahů $p_{k+1} = r_{k+1} - AP_{k+1}v_{k+1}$ a $r_{k+1} \in \mathcal{K}_{k+2}$ plyne, že $p_{k+1} \in \mathcal{K}_{k+2}$. Odtud dostáváme

$$\mathcal{S}_{k+2} \subseteq \mathcal{K}_{k+2} \quad \text{a} \quad \mathcal{R}_{k+2} \subseteq \mathcal{K}_{k+2}.$$

Z lemmatu 1.6 plyne, že $dim(\mathcal{S}_{k+2}) = dim(\mathcal{R}_{k+2}) = k+2$ a tudíž i $dim(\mathcal{K}_{k+2}) = k+2$ a odtud

$$\mathcal{S}_{k+2} = \mathcal{R}_{k+2} = \mathcal{K}_{k+2}.$$

■

Lemma 1.8 : *Rezidua r_i pro $i = 0 \dots k$ jsou vzájemně ortogonální.*

Důkaz : Podle lemmatu 1.5 a 1.7 je

$$r_k \perp sp\{p_0, .., p_{k-1}\} = sp\{r_0, .., r_{k-1}\}.$$

■

Lemma 1.9 : Vektor p_{k+1} lze psát ve tvaru

$$p_{k+1} = r_{k+1} + \beta_k p_k.$$

Důkaz : Protože úbytek funkcionálu $\phi(x)$ nezávisí na délce vektoru p_{k+1}, stačí dokázat, že $p_{k+1} \in sp\{r_{k+1}, p_k\}$. Podle vztahu (1.12) můžeme p_{k+1} psát ve tvaru $p_{k+1} = r_{k+1} - AP_{k+1}v_{k+1}$. Uvažujme vektor v_{k+1} ve tvaru $v_{k+1} = (w, \mu)$, kde $w \in \mathbb{R}^k$ a $\mu \in \mathbb{R}$. Potom platí

$$p_{k+1} = r_{k+1} - AP_k w - \mu A p_k = r_{k+1} - AP_k w + \frac{\mu}{\alpha_k}(r_{k+1} - r_k) =$$

$$= (1 + \frac{\mu}{\alpha_k}) r_{k+1} - \frac{\mu}{\alpha_k} r_k - AP_k w = (1 + \frac{\mu}{\alpha_k}) r_{k+1} + h_k,$$

kde

$$h_k = -\frac{\mu}{\alpha_k} r_k - AP_k w.$$

Protože $h_k \in sp\{r_0, .., r_k\}$, je $h_k \perp r_{k+1}$. Platí

$$\| r_{k+1} - AP_{k+1}v_{k+1} \|^2 = \Big(1 + \frac{\mu}{\alpha_k}\Big)^2 \| r_{k+1} \|^2 + \Big(\frac{\mu}{\alpha_k}\Big)^2 \| r_k - AP_k \frac{\alpha_k}{\mu} w \|^2 \geq$$

$$\geq \Big(1 + \frac{\mu}{\alpha_k}\Big)^2 \| r_{k+1} \|^2 + \Big(\frac{\mu}{\alpha_k}\Big)^2 \| r_k - AP_k v_k \|^2 .$$

Protože na levé straně je minimum přes všechny vektory $v \in \mathbb{R}^{k+1}$, místo neostré nerovnosti můžeme psát rovnost a z jednoznačnosti určení vektoru w plyne $v_k = \frac{\alpha_k}{\mu} w$. Dosadíme-li za $w = \frac{\mu}{\alpha_k} v_k$ do h_k dostaneme $h_k = -\frac{\mu}{\alpha_k} p_k$. Tedy

$$p_{k+1} = (1 + \frac{\mu}{\alpha_k}) r_{k+1} - \frac{\mu}{\alpha_k} p_k.$$

■

Lemma 1.10 : Platí rovnosti

$$\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}, \qquad \alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}.$$

Důkaz : Rovnost $p_{k+1} = r_{k+1} + \beta_k p_k$ přenásobíme $p_k^T A$ zleva

$$0 = p_k^T A r_{k+1} + \beta_k p_k^T A p_k$$

a odtud plyne, že

$$\beta_k = -\frac{p_k^T A r_{k+1}}{p_k^T A p_k}.$$

Dále přenásobením vztahu $r_{k+1} = r_k - \alpha_k A p_k$ vektory r_{k+1}^T a p_k^T dostaneme rovnosti

$$r_{k+1}^T r_{k+1} = -\alpha_k r_{k+1}^T A p_k,$$

$$r_k^T r_k = \alpha_k p_k^T A p_k.$$

Pomocí těchto vztahů plyne, že

$$\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}.$$

O α_k víme, že jej lze psát ve tvaru

$$\alpha_k = \frac{p_k^T r_k}{p_k^T A p_k}.$$

Zbývá tedy dokázat, že $p_k^T r_k = r_k^T r_k$. Uvědomíme-li si, že $p_k = r_k + \beta_{k-1} p_{k-1}$, dostáváme

$$p_k^T r_k = r_k^T r_k + \beta_{k-1} p_{k-1}^T r_k = r_k^T r_k. \qquad \blacksquare$$

Algoritmus 1.2

Input $x_0,\ A,\ b$;
$r_0 = b - Ax_0$;
$p_0 = r_0$;
for $k = 0$ **to** $n-1$ **do**
  if $r_k = 0$ **then** $x^* = x_k$ *and exit*
  else
    begin
      $\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$;
      $x_{k+1} = x_k + \alpha_k p_k$;
      $r_{k+1} = r_k - \alpha_k A p_k$;
      $\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$;
      $p_{k+1} = r_{k+1} + \beta_k p_k$;
    end;
$x^* = x_n$.

Definice 1.3 : Metodu definovanou algoritmem 1.2 nazýváme metodou sdružených gradientů (CGM).

1.2 Geometrická interpretace

V tomto odstavci se budeme zabývat tím, jak si představit iterační proces metody CGM v $\mathbb{R}^n$ z geometrického hlediska.

Definice 1.4 : Označme $\mathcal{S}$ jednotkovou kouli v $\mathbb{R}^n$.

Lemma 1.11 : *Nechť $v \in \mathbb{R}^n$, $A \in L(\mathbb{R}^n)$ je symetrická, pozitivně definitní matice. Potom platí*

(1.16)
$$\| v \|_A = \inf\{\alpha \geq 0; v \in \alpha A^{-1/2}\mathcal{S}\}.$$

Důkaz : Platí

$$\| v \|_A = \sqrt{v^T A v} = \sqrt{v^T A^{1/2^T} A^{1/2} v} = \| A^{1/2} v \|_2 = \inf\{\alpha \geq 0; A^{1/2} v \in \alpha\mathcal{S}\} =$$
$$= \inf\{\alpha \geq 0; v \in \alpha A^{-1/2}\mathcal{S}\}. \qquad \blacksquare$$

Poznámka : V dalším textu se budeme zabývat tím, jak si představit množinu $\alpha A^{-1/2}\mathcal{S}$.

Poznámka : Je-li $v_1, ..., v_n$ ortonormální báze z vlastních vektorů matice A (A je pozitivně definitní) a $\lambda_1, \dots, \lambda_n$ jsou příslušná vlastní čísla matice A, potom matice $A^{-1/2}$ má vlastní vektory $v_1, \dots, v_n$ a příslušná vlastní čísla $\frac{1}{\sqrt{\lambda_1}}, \dots, \frac{1}{\sqrt{\lambda_n}}$.

Lemma 1.12 : Platí

$$v \in A^{-1/2}\mathcal{S} \Leftrightarrow v = \sum_{k=1}^{n} \frac{c_k}{\sqrt{\lambda_k}} v_k, \quad \text{kde} \quad \sum_{k=1}^{n} c_k^2 \leq 1.$$

Důkaz : Protože

$$x \in \mathcal{S} \Leftrightarrow x = \sum_{k=1}^{n} c_k v_k, \quad \sum_{k=1}^{n} c_k^2 \leq 1,$$

platí

$$v \in A^{-1/2}\mathcal{S} \Leftrightarrow v = \sum_{k=1}^{n} c_k A^{-1/2} v_k = \sum_{k=1}^{n} \frac{c_k}{\sqrt{\lambda_k}} v_k, \quad \sum_{k=1}^{n} c_k^2 \leq 1. \qquad \blacksquare$$

Poznámka : Nechť d_1, d_2 je ortonormální báze v $\mathbb{R}^2$ (d_1, d_2 udávají směr souřadnicových os). Elipsou rozumíme množinu bodů $X = [x, y]$ takovou, že pro souřadnice x, y bodu X platí :

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1.$$

Čísla $a > 0, b > 0$ nazýváme poloosami elipsy.
V našem případě koeficienty $z_k = \frac{c_k}{\sqrt{\lambda_k}}$ splňují nerovnost

$$\sum_{k=1}^{n} \frac{z_k^2}{(\frac{1}{\sqrt{\lambda_k}})^2} \leq 1.$$

Věta 1.2 : *Vektory $v \in \alpha A^{-1/2}\mathcal{S}$ tvoří vnitřek n-rozměrného elipsoidu s poloosami $\frac{\alpha}{\sqrt{\lambda_k}}$. (Vektory $v_1, \dots, v_n$ udávají směr souřadnicových os.)*

Důkaz : Plyne z předchozí poznámky. $\blacksquare$

Lemma 1.13 : Pro vektor x_k platí

$$\| x_k - x^* \|_A = \min_{u \in x_0 + \mathcal{K}_k} \inf \{\alpha \geq 0; u \ \in x^* + \alpha A^{-1/2} \mathcal{S}\}.$$

Důkaz : Protože platí

$$\| x - x^* \|_A = (x - x^*)^T A (x - x^*) = x^T A x - 2 b^T x + (x^*)^T A x^* = 2\phi(x) + (x^*)^T A x^*$$

a

$$\phi(x_k) = \min_{u \in x_0 + \mathcal{K}_k} \phi(u),$$

platí též

$$\| x_k - x^* \|_A = \min_{u \in x_0 + \mathcal{K}_k} \| u - x^* \|_A .$$

Celkem tedy dostáváme

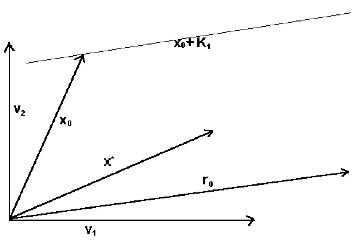
$$\| x_k - x^* \|_A = \min_{u \in x_0 + \mathcal{K}_k} \| u - x^* \|_A = \min_{u \in x_0 + \mathcal{K}_k} \inf \{\alpha \geq 0; u - x^* \in \alpha A^{-1/2} \mathcal{S}\}$$

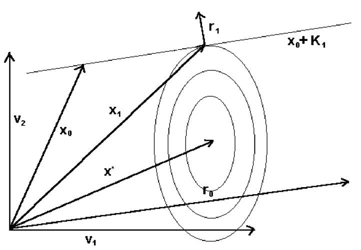
$$= \min_{u \in x_0 + \mathcal{K}_k} \inf \{\alpha \geq 0; u \ \in x^* + \alpha A^{-1/2} \mathcal{S}\}.$$

■

Poznámka : Množinu $x^* + \alpha A^{-1/2} \mathcal{S}$ si lze představit jako vnitřek elipsoidu se středem v bodě x^* a příslušnými poloosami.

Příklad : Pro [illegible] $\mathbb{R}^2$ [illegible]





Jinak řečeno, najdeme parametr α tak, aby se daná elipsa se středem v x^* dotýkala přímky $x_0 + \mathcal{K}_1$.Tímto bodem je dána další aproximace řešení, tedy vektor x_1.
Podobně si můžeme tuto situaci představit v $\mathbb{R}^3$.
V prvním kroku budeme volit α tak, aby se daný elipsoid dotýkal přímky $x_0 + \mathcal{K}_1$. Bod dotyku určuje vektor x_1.
V druhém kroku volíme α tak, aby se daný elipsoid dotýkal roviny $x_0 + \mathcal{K}_2$. Opět bod dotyku určuje vektor x_2. Ve třetím kroku již máme přesné řešení.

1.3 Umocnění metody sdružených gradientů

Jak si ukážeme v následujícím lemmatu, vektor r_k lze vyjádřit ve tvaru $r_k = \varphi_k(A)r_0$. Umocněnou metodu sdružených gradientů dostaneme tak, že budeme volit vektor $\bar{r}_k$ jako $\bar{r}_k = \varphi_k^2(A)r_0 = \varphi_k(A)r_k$. Je-li tedy $r_k = 0$, je i $\bar{r}_k = 0$. Umocnění se používá hlavně u metody BCG, kde se tímto vyhneme použití transponované matice. Pokusili jsme se použít umocnění i u metody CGM a zjistit, jaké přináší výhody a nevýhody.

Lemma 1.14 : Vektory r_k a p_k z algoritmu 1.2 lze vyjádřit ve tvaru

$$r_k = \varphi_k(A)r_0 \quad \text{a} \quad p_k = \psi_k(A)r_0, \tag{1.17}$$

kde $\varphi_k(A)$ a $\psi_k(A)$ jsou maticové polynomy stupně nejvýše k. Polynom $\varphi_k(A)$ má navíc vlastnost $\varphi_k(0) = 1$.

Důkaz : Pro k=0 je $\varphi_0(A) = I = \psi_0(A)$ a (1.17) platí.
Nechť (1.17) platí pro k. Potom

$$r_{k+1} = r_k - \alpha_k A p_k = \varphi_k(A)r_0 - \alpha_k A\psi_k(A)r_0 = \varphi_{k+1}(A)r_0$$

a platí $\varphi_{k+1}(0) = 1$. Vektor p_{k+1} lze psát jako

$$p_{k+1} = r_{k+1} + \beta_k p_k = \varphi_{k+1}(A)r_0 + \beta_k\psi_k(A)r_0 = \psi_{k+1}(A)r_0. \qquad \blacksquare$$

Z algoritmu 1.2 je vidět, že tyto polynomy je možné počítat dle následujícího algoritmu.

Algoritmus 1.3

Input $x_0,\ A,\ b;$
$\varphi_0(A) = I;\ \psi_0(A) = I;\ r_0 = b - Ax_0;$
for $k = 0$ **to** $n-1$ **do**
  begin
    $\alpha_k = \frac{r_0^T\varphi_k^2(A)r_0}{r_0^T A\psi_k^2(A)r_0};$
    $\varphi_{k+1}(A) = \varphi_k(A) - \alpha_k A\psi_k(A);$
    $\beta_k = \frac{r_0^T\varphi_{k+1}^2(A)r_0}{r_0^T\varphi_k^2(A)r_0};$
    $\psi_{k+1}(A) = \varphi_{k+1}(A) + \beta_k\psi_k(A);$
  end.

Pro přehlednost pišme v dalším φ_k místo $\varphi_k(A)$ a ψ_k místo $\psi_k(A)$.

Lemma 1.15 : Nechť

$$\varphi_{k+1} = \varphi_k - \alpha_k A\psi_k, \tag{1.18}$$
$$\psi_{k+1} = \varphi_{k+1} + \beta_k\psi_k, \tag{1.19}$$

kde α_k a β_k jsou libovolná čísla, A matice.
Potom platí :

$$\varphi_{k+1}\psi_k = \varphi_k\psi_k - \alpha_k A\psi_k^2, \tag{1.20}$$
$$\varphi_{k+1}\psi_{k+1} = \varphi_{k+1}^2 + \beta_k\varphi_{k+1}\psi_k, \tag{1.21}$$
$$\varphi_{k+1}^2 = \varphi_k^2 - \alpha_k A(\varphi_k\psi_k + \varphi_{k+1}\psi_k), \tag{1.22}$$
$$\psi_{k+1}^2 = \varphi_{k+1}\psi_{k+1} + \beta_k(\varphi_{k+1}\psi_k + \beta_k\psi_k^2). \tag{1.23}$$

Důkaz : Vztah (1.20) dostaneme přenásobením (1.18) zprava polynomem ψ_k. Vztah (1.21) získáme přenásobením (1.19) zleva polynomem φ_{k+1}. Vztah (1.22) dostaneme umocněním a rozepsáním polynomu φ^2_{k+1}

$$\begin{aligned}\varphi^2_{k+1} &= \varphi^2_k - 2\alpha_k A\ \varphi_k\psi_k + \alpha^2_k A^2\psi^2_k = \varphi^2_k - \alpha_k A(2\varphi_k\psi_k - \alpha_k A\psi^2_k) = \\ &= \varphi^2_k - \alpha_k A(\varphi_k\psi_k + \varphi_{k+1}\psi_k).\end{aligned}$$

Vztah (1.23) dostaneme umocněním a úpravou polynomu ψ^2_{k+1}

$$\psi^2_{k+1} = \varphi^2_{k+1} + 2\beta_k\varphi_{k+1}\psi_k + \beta^2_k\psi^2_k = \varphi^2_{k+1} + \beta_k\varphi_{k+1}\psi_k + \beta_k(\varphi_{k+1}\psi_k + \beta_k\psi^2_k) =$$

$$= \varphi_{k+1}\psi_{k+1} + \beta_k(\varphi_{k+1}\psi_k + \beta_k\psi^2_k).$$

■

Poznámka : Umocnění metody CGM spočívá v tom, že položíme $\tilde{r}_k = \varphi^2_k(A)r_0$. Zřejmě tedy musíme nalézt vhodný algoritmus, který by počítal polynom $\varphi^2_k(A)$.

Definice 1.5 : Definujme maticové polynomy

$$\mathcal{U}_k(A) = \varphi_k\psi_k \quad \text{a} \quad \mathcal{Q}_k(A) = \varphi_{k+1}\psi_k.$$

V dalším opět pišme $\mathcal{U}_k$, $\mathcal{Q}_k$ místo $\mathcal{U}_k(A)$, $\mathcal{Q}_k(A)$.

Využijeme-li algoritmu 1.3 a lemmatu 1.15, můžeme pro výpočet polynomu $\varphi^2_k(A)$ použít následujícího algoritmu.

Algoritmus 1.4

Input $x_0,\ A,\ b$;
$\varphi^2_0 = I;\ \psi^2_0 = I;\ r_0 = b - Ax_0;\ \mathcal{U}_0 = I$;
for $k = 0$ **to** $n-1$ **do**
  begin
    $\alpha_k = \frac{r_0^T\varphi^2_k r_0}{r_0^T A\psi^2_k r_0}$;
    $\mathcal{Q}_k = \mathcal{U}_k - \alpha_k A\ \psi^2_k$;
    $\varphi^2_{k+1} = \varphi^2_k - \alpha_k A(\mathcal{U}_k + \mathcal{Q}_k)$;
    $\beta_k = \frac{r_0^T\varphi^2_{k+1}r_0}{r_0^T\varphi^2_k r_0}$;
    $\mathcal{U}_{k+1} = \varphi^2_{k+1} + \beta_k\mathcal{Q}_k$;
    $\psi^2_{k+1} = \mathcal{U}_{k+1} + \beta_k(\mathcal{Q}_k + \beta_k\psi^2_k)$;
  end.

Definice 1.6 : Ve shodě s algoritmem 1.4 definujme

$$\begin{aligned}\bar{r}_k &= \varphi^2_k(A)r_0, \quad \bar{p}_k = \psi^2_k(A)r_0, \quad v_k = A\psi^2_k(A)r_0, \\ u_k &= \mathcal{U}_k(A)r_0, \quad q_k = \mathcal{Q}_k(A)r_0, \quad \bar{x}_k = A^{-1}(b - \bar{r}_k).\end{aligned}$$

Lemma 1.16 : Platí

$$v_k = A\bar{p}_k, \qquad \alpha_k = \frac{r_0^T\bar{r}_k}{r_0^T v_k}, \qquad q_k = u_k - \alpha_k v_k,$$

$$\bar{r}_{k+1} = \bar{r}_k - \alpha_k A(u_k + q_k), \qquad \bar{x}_{k+1} = \bar{x}_k + \alpha_k(u_k + q_k),$$

$$\beta_k = \frac{r_0^T\bar{r}_{k+1}}{r_0^T\bar{r}_k}, \qquad u_{k+1} = \bar{r}_k + \beta_k q_k, \qquad \bar{p}_{k+1} = u_{k+1} + \beta_k(q_k + \beta_k\bar{p}_k).$$

Důkaz : Platí

$$
\begin{aligned}
v_k &= A\psi_k^2(A)r_0 = A\bar{p}_k, \\
\alpha_k &= \frac{r_0^T\varphi_k^2 r_0}{r_0^T A\psi_k^2 r_0} = \frac{r_0^T\bar{r}_k}{r_0^T A\bar{p}_k} = \frac{r_0^T\bar{r}_k}{r_0^T v_k}, \\
q_k &= \varphi_{k+1}\psi_k r_0 = \varphi_k\psi_k r_0 - \alpha_k A\psi_k^2 r_0 = u_k - \alpha_k v_k, \\
\bar{r}_{k+1} &= \varphi_{k+1}^2 r_0 = \varphi_k^2 r_0 - \alpha_k A(\varphi_k\psi_k + \varphi_{k+1}\psi_k)r_0 = \bar{r}_k - \alpha_k A(u_k + q_k), \\
\bar{x}_{k+1} &= A^{-1}(b - \bar{r}_{k+1}) = A^{-1}(b - (b - A\bar{x}_k) + \alpha_k A(u_k + q_k)) = \bar{x}_k + \alpha_k(u_k + q_k), \\
\beta_k &= \frac{r_0^T\varphi_{k+1}^2 r_0}{r_0\varphi_k^2 r_0} = \frac{r_0^T\bar{r}_{k+1}}{r_0^T\bar{r}_k}, \\
u_{k+1} &= \varphi_{k+1}\psi_{k+1}r_0 = \varphi_{k+1}^2 r_0 + \beta_k\varphi_{k+1}\psi_k = \bar{r}_k + \beta_k q_k, \\
\bar{p}_{k+1} &= \psi_{k+1}^2 r_0 = \varphi_{k+1}\psi_{k+1}r_0 + \beta_k(\varphi_{k+1}\psi_k r_0 + \beta_k\psi_k^2 r_0) = u_{k+1} + \beta_k(q_k + \beta_k\bar{p}_k).
\end{aligned}
$$

■

Poznamenejme ještě, že $\bar{r}_0 = \varphi_0^2(A)r_0 = r_0$ a $\bar{x}_0 = A^{-1}(b - \bar{r}_0) = x_0$. Na základě předchozího lemmatu a algoritmu 1.4 můžeme formulovat následující algoritmus.

Algoritmus 1.5

Input $\bar{x}_0,\ A,\ b$;
$\bar{r}_0 = b - A\bar{x}_0$;
$\bar{p}_0 = u_0 = \bar{r}_0$;
for $k = 0$ **to** $n-1$ **do**
  if $\bar{r}_k = 0$ **then** $x^* = \bar{x}_k$ *and exit*
  else
    begin
      $v_k = A\bar{p}_k$;
      $\alpha_k = \frac{\bar{r}_0^T\bar{r}_k}{\bar{r}_0^T v_k}$;
      $q_k = u_k - \alpha_k v_k$;
      $\bar{r}_{k+1} = \bar{r}_k - \alpha_k A(u_k + q_k)$;
      $\bar{x}_{k+1} = \bar{x}_k + \alpha_k(u_k + q_k)$;
      $\beta_k = \frac{\bar{r}_0^T\bar{r}_{k+1}}{\bar{r}_0^T\bar{r}_k}$;
      $u_{k+1} = \bar{r}_k + \beta_k q_k$;
      $p_{k+1} = u_{k+1} + \beta_k(q_k + \beta_k\bar{p}_k)$;
    end;
$x^* = \bar{x}_n$.

Definice 1.7 : Metodu definovanou algoritmem 1.5 nazveme umocněnou metodou sdružených gradientů (CGMS).

Poznámka : Povšimněme si, že algoritmus metody CGMS používá dvakrát násobení matice-vektor, zatímco metoda CGM pouze jednou. Metoda CGMS používá i více násobení vektor-vektor a skalár-vektor než metoda CGM. Zřejmě tedy bude tato metoda zhruba dvakrát pomalejší, než metoda CGM.

1.4 Odhady norem vektoru chyby

Ukážeme si, že lze obhadnou rychlost konvergence metody CGM a CGMS v energetické normě $\| x \|_A = \sqrt{x^T A x}$ pomocí čísla podmíněnosti matice A a počáteční aproximace řešení x_0.

Odhad chyby CGM

Lemma 1.17 : *Platí*

$$x_k - x^* = q_k(A)(x_0 - x^*),$$

kde $q_k(A)$ je maticový polynom stupně nejvýše k s vlastností $q_k(0) = 1$.

Důkaz : Z algoritmu 1.2 plyne, že

$$x_k = x_{k-1} + \alpha_{k-1} p_{k-1} = x_0 + \alpha_0 p_0 + ... + \alpha_{k-1} p_{k-1} = x_0 + \sum_{i=0}^{k-1} \gamma_i A^i r_0 =$$

$$= x_0 + \sum_{i=0}^{k-1} \gamma_i (A^i b - A^{i+1} x_0) = x_0 - \sum_{i=0}^{k-1} \gamma_i A^{i+1} (x_0 - x^*).$$

Odečteme-li od obou stran x^* dostáváme

$$x_k - x^* = q_k(A)(x_0 - x^*)$$

a polynom q_k má požadovanou vlastnost. ■

Lemma 1.18 : *Platí*

$$\| x_k - x^* \|_A \leq \min_{q \in \Pi_k^{(1)}} \| q(A) \|_A \| x_0 - x^* \|_A,$$

kde $\Pi_k^{(1)}$ je prostor polynomů stupně nejvýše k, které nabývají hodnoty 1 v bodě 0.

Důkaz : Pro $\| x_k - x^* \|_A$ platí

$$\| x_k - x^* \|_A = \min_{u \in x_0 + sp\{r_0, .., A^{k-1} r_0\}} \| u - x^* \|_A = \min_{q \in \Pi_k^{(1)}} \| q(A)(x_0 - x^*) \|_A \leq$$

$$\leq \min_{q \in \Pi_k^{(1)}} (\| q(A) \|_A \| x_0 - x^* \|_A) = \min_{q \in \Pi_k^{(1)}} \| q(A) \|_A \| x_0 - x^* \|_A \, .$$ ■

Definice 1.8 : Označme $\mathcal{E} = A^{-1/2} \mathcal{S}$ elipsoid s poloosami $\frac{1}{\sqrt{\lambda_i}}$ pro $i = 1, ..., n$, kde λ_i jsou vlastní čísla matice A.

Lemma 1.19 : Platí

$$\| q(A) \|_A = \min \{\alpha \geq 0; q(A)\mathcal{E} \subset \alpha \mathcal{E}\}.$$

Důkaz : Pro $\| q(A) \|_A$ platí

$$\| q(A) \|_A = \max_{\|y\|_A \leq 1} \| q(A) y \|_A = \max_{y \in \mathcal{E}} \min \{\alpha \geq 0; q(A) y \in \alpha \mathcal{E}\} =$$

$$= \min \{\alpha \geq 0; \forall y \in \mathcal{E} : q(A) y \in \alpha \mathcal{E}\} = \min \{\alpha \geq 0; q(A)\mathcal{E} \subset \alpha \mathcal{E}\}.$$ ■

Lemma 1.20 : Platí

$$\| x_k - x^* \|_A \leq \| x_0 - x^* \|_A \min_{q \in \Pi_k^{(1)}} \max_{\lambda \in <\lambda_1, \lambda_n>} | q(\lambda) |, \tag{1.24}$$

přičemž uvažujeme, že λ_1 je nejmenší a λ_n největší vlastní číslo matice A.

Důkaz : Vezmeme-li libovolný vektor $y \in \mathcal{E}$, můžeme ho napsat ve tvaru

$$y = \sum_{i=1}^{n} c_i v_i, \qquad \sum_{i=1}^{n} \frac{c_i^2}{\frac{1}{\lambda_i}} \leq 1.$$

Platí

$$q(A)y = q(A) \sum_{i=1}^{n} c_i v_i = \sum_{i=1}^{n} c_i q(\lambda_i) v_i.$$

Pro koeficienty této lineární kombinace platí

$$\sum_{i=1}^{n} \frac{c_i^2 q^2(\lambda_i)}{\frac{1}{\lambda_i}} \leq \max_{k=1,..,n} q^2(\lambda_k) \sum_{i=1}^{n} \frac{c_i^2}{\frac{1}{\lambda_i}} \leq \max_{k=1,..,n} q^2(\lambda_k) \quad \forall y \in \mathcal{E},$$

to znamená, že $\forall y \in \mathcal{E}$ je $q(A)y$ částí elipsoidu s poloosami $\frac{\max_k |q(\lambda_k)|}{\sqrt{\lambda_i}}$, tedy

$$q(A)\mathcal{E} \subset \max_k | q(\lambda_k) | \mathcal{E}.$$

Pro $\| x_k - x^* \|_A$ platí

$$\| x_k - x^* \|_A \leq \| x_0 - x^* \|_A \min_{q \in \Pi_k^{(1)}} \| q(A) \|_A \leq \| x_0 - x^* \|_A \min_{q \in \Pi_k^{(1)}} \min \{\alpha \geq 0; q(A)\mathcal{E} \subset \alpha\mathcal{E}\}$$

$$\leq \| x_0 - x^* \|_A \min_{q \in \Pi_k^{(1)}} \max_{i=1,..,n} | q(\lambda_i) | \leq \| x_0 - x^* \|_A \min_{q \in \Pi_k^{(1)}} \max_{\lambda \in <\lambda_1, \lambda_n>} | q(\lambda) |. \qquad \blacksquare$$

Zbývá odhadnout číslo

$$\min_{q \in \Pi_k^{(1)}} \max_{\lambda \in <\lambda_1, \lambda_n>} | q(\lambda) |.$$

Polynomem stupně k, který splňuje tyto požadavky je Čebyševův polynom, normovaný tak, aby jeho hodnota byla v bodě 0 rovna jedné a přetransformovaný pomocí prostého zobrazení z intervalu $< -1, 1 >$ na interval $< \lambda_1, \lambda_n >$.

Definice 1.9 : Definujme polynom $T_k(x)$ jako

$$\begin{aligned} T_k(x) &= \cos(k \arccos(x)) \qquad \text{je} - \text{li} \quad x \in < -1, 1 >, \\ T_k(x) &= \cosh(k \operatorname{arg\,cosh}(x)) \qquad \text{pro} \quad x > 1. \end{aligned}$$

Polynom T_k nazveme Čebyševovýv polynomem stupně k. Definujme polynom $q_k(\lambda)$ jako

$$q_k(\lambda) = \frac{T_k(\frac{\lambda_1 + \lambda_n - 2\lambda}{\lambda_n - \lambda_1})}{T_k(\frac{\lambda_1 + \lambda_n}{\lambda_n - \lambda_1})}.$$

Lemma 1.21 : Pro $\| x_k - x^* \|_A$ platí odhad

$$\| x_k - x^* \|_A \leq \frac{\| x_0 - x^* \|_A}{| T_k(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1}) |}. \tag{1.25}$$

Důkaz : Uvážíme-li, že $| \frac{\lambda_1+\lambda_n-2\lambda}{\lambda_n-\lambda_1} | \leq 1$ pro $\lambda \in < \lambda_1, \lambda_n >$, dostáváme odhad

$$\min_{q\in\Pi_k^{(1)}} \max_{\lambda\in<\lambda_1,\lambda_n>} | q(\lambda) | = \max_{\lambda\in<\lambda_1,\lambda_n>} | q_k(\lambda) | \leq \frac{1}{| T_k(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1}) |}.$$

(Poslední nerovnost plyne z toho, že $| T_k(x) | \leq 1$ pro $x \in < -1, 1 >$.)
Z poslední nerovnosti a z (1.24) již plyne (1.25) . ■

Věta 1.3 : Platí

$$\| x_k - x^* \|_A \leq 2\Big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\Big)^k \| x_0 - x^* \|_A,$$

kde κ je číslo podmíněnosti matice A.

Důkaz : Protože je $\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1} > 1$, je

$$T_k(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1}) = \cosh(k \operatorname{arg}\cosh(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1})).$$

Funkce $y = \operatorname{arg}\cosh(x)$ je inverzní funkcí k funkci $x = \cosh(y)$ uvažované jen pro $y \in < 0, \infty)$; Je definovaná pro každé $x \in < 1, \infty)$. Lze ji vyjádřit ve tvaru

$$\operatorname{arg}\cosh(x) = \ln(x + \sqrt{x^2-1}).$$

Můžeme tedy psát

$$\cosh(k \operatorname{arg}\cosh(x)) = \frac{1}{2}(e^{k\ln(x+\sqrt{x^2-1})} + e^{-k\ln(x+\sqrt{x^2-1})}) =$$

$$= \frac{1}{2}((x+\sqrt{x^2-1})^k + (x+\sqrt{x^2-1})^{-k}).$$

Označíme-li si $\delta = x + \sqrt{x^2-1}$ pro $x = \frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1}$, dostáváme

$$\frac{1}{T_k(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1})} = \frac{2}{\delta^k+\delta^{-k}} = \frac{2\delta^{-k}}{1+\delta^{-2k}} \leq 2\delta^{-k}.$$

Zbývá už jen spočíst δ.

$$\delta = \frac{\lambda_n+\lambda_1}{\lambda_n-\lambda_1} + \sqrt{\Big(\frac{\lambda_n+\lambda_1}{\lambda_n-\lambda_1}\Big)^2 - 1} = \frac{\lambda_n+\lambda_1}{\lambda_n-\lambda_1} + \sqrt{\frac{4\lambda_1\lambda_n}{(\lambda_n-\lambda_1)^2}} =$$

$$= \frac{(\sqrt{\lambda_n}+\sqrt{\lambda_1})^2}{\lambda_n-\lambda_1} = \frac{\sqrt{\lambda_n}+\sqrt{\lambda_1}}{\sqrt{\lambda_n}-\sqrt{\lambda_1}} = \frac{\sqrt{\frac{\lambda_n}{\lambda_1}}+1}{\sqrt{\frac{\lambda_n}{\lambda_1}}-1} = \frac{\sqrt{\kappa}+1}{\sqrt{\kappa}-1} \geq 0.$$

Konečný odhad tedy zní

$$\| x_k - x^* \|_A \leq \frac{\| x_0 - x^* \|_A}{| T_k(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1}) |} = \frac{\| x_0 - x^* \|_A}{T_k(\frac{\lambda_1+\lambda_n}{\lambda_n-\lambda_1})} \leq 2\Big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\Big)^k \| x_0 - x^* \|_A .$$ ■

Odhad chyby CGMS

Lemma 1.22 : Pro $\| \bar{x_k} - x^* \|_A$, kde $\bar{x_k}$ je vektor generovaný metodou CGMS, platí odhad

$$\| \bar{x_k} - x^* \|_A \leq 4\Big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\Big)^{2k} \| x_0 - x^* \|_A .$$

Důkaz : Platí

$$\begin{aligned} \bar{r_k} &= \varphi_k^2(A)r_0 = \varphi_k(A)r_k, \\ Ax^* - A\bar{x_k} &= b - A\bar{x_k} = \bar{r_k} = \varphi_k^2(A)r_0 = \varphi_k^2(A)(Ax^* - Ax_0) \end{aligned}$$

aneb

$$\bar{x_k} - x^* = \varphi_k^2(A)(x_0 - x^*) \quad \text{a} \quad x_k - x^* = \varphi_k(A)(x_0 - x^*).$$

Z jednoznačnosti určení koeficientů polynomu φ_k a polynomu q_k z podkapitoly 1.3 plyne, že $q_k(A) = \varphi_k(A)$. Dostáváme tedy odhad

$$\| \bar{x_k} - x^* \|_A \leq \| \varphi_k^2(A) \|_A \| x_0 - x^* \|_A \leq 4\Big(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\Big)^{2k} \| x_0 - x^* \|_A,$$

kde κ je číslo podmíněnosti matice A. ■

Poznámka : Umocnění tedy přineslo zlepšení odhadu konvergence. Je však otázkou, zda rychlejší konvergence metody CGMS je dostatečnou kompenzací ke zvýšenému počtu operací. Až numerické experimenty ukáží, zda je tato metoda konkurencí pro metodu CGM.

1.5 Předpodmínění

Nechť C je regulární matice. Přenásobíme-li rovnici (1.1) maticí C^T zleva, dostáváme

$$C^T ACC^{-1}x = C^T b.$$

Provedeme-li nyní substituci

$$\widehat{A} = C^T AC, \qquad \widehat{x} = C^{-1}x, \qquad \widehat{b} = C^T b,$$

dostáváme systém

$$\widehat{A}\widehat{x} = \widehat{b}. \tag{1.26}$$

Mezi řešením soustavy (1.1) a soustavy (1.26) platí vztah $\widehat{x} = C^{-1}x$.

Lemma 1.23 : *Matice $\widehat{A}$ je symetrická a pozitivně definitní.*

Důkaz : Platí

$$\widehat{A}^T = C^T A^T C = C^T AC = \widehat{A}$$

Pro libovolné $x \neq 0$ je též $Cx \neq 0$. Z toho plyne, že je-li $x \neq 0$, je

$$x^T \widehat{A}x = x^T C^T ACx = (Cx)^T ACx = y^T Ay > 0.$$ ■

Je-li x_0 počáteční odhad soustavy (1.1), definujme počáteční odhad soustavy (1.26) jako $\widehat{x}_0 = C^{-1}x_0$. Na soustavu (1.26) můžeme nyní použít metodu CGM a CGMS.

Algoritmus 1.6

Input $x_0,\ \widehat{A},\ \widehat{b},\ C$;
$\widehat{x}_0 = C^{-1}x_0$;
$\widehat{r}_0 = \widehat{b} - \widehat{A}\widehat{x}_0$;
$\widehat{p}_0 = \widehat{r}_0$;
for $k = 0$ **to** $n-1$ **do**
  if $\widehat{r}_k = 0$ **then** $x^* = C\widehat{x}_k$ *and exit*
  else
    begin
      $\alpha_k = \frac{\widehat{r}_k^T\widehat{r}_k}{\widehat{p}_k^T\widehat{A}\widehat{p}_k}$;
      $\widehat{x}_{k+1} = \widehat{x}_k + \alpha_k\widehat{p}_k$;
      $\widehat{r}_{k+1} = \widehat{r}_k - \alpha_k\widehat{A}\widehat{p}_k$;
      $\beta_k = \frac{\widehat{r}_{k+1}^T\widehat{r}_{k+1}}{\widehat{r}_k^T\widehat{r}_k}$;
      $\widehat{p}_{k+1} = \widehat{r}_{k+1} + \beta_k\widehat{p}_k$;
    end;
$x^* = C\widehat{x}_n$.

Algoritmus 1.7

Input $\bar{x}_0,\ \widehat{A},\ \widehat{b},\ C$;
$\widehat{x}_0 = C^{-1}\bar{x}_0$;
$\widehat{r}_0 = \widehat{b} - \widehat{A}\widehat{x}_0$;
$\widehat{p}_0 = \widehat{r}_0 = \widehat{u}_0$;
for $k = 0$ **to** $n-1$ **do**
  if $\widehat{r}_k = 0$ **then** $x^* = C\widehat{x}_k$ *and exit*
  else
    begin
      $\widehat{v}_k = \widehat{A}\widehat{p}_k$; $\alpha_k = \frac{\widehat{r}_0^T\widehat{r}_k}{\widehat{r}_0^T\widehat{v}_k}$;
      $\widehat{q}_k = \widehat{u}_k - \alpha_k\widehat{v}_k$;
      $\widehat{r}_{k+1} = \widehat{r}_k - \alpha_k\widehat{A}(\widehat{u}_k + \widehat{q}_k)$;
      $\widehat{x}_{k+1} = \widehat{x}_k + \alpha_k(\widehat{u}_k + \widehat{q}_k)$;
      $\beta_k = \frac{\widehat{r}_0^T\widehat{r}_{k+1}}{\widehat{r}_0\widehat{r}_k}$; $\widehat{u}_{k+1} = \widehat{r}_{k+1} + \beta_k\widehat{q}_k$;
      $\widehat{p}_{k+1} = \widehat{u}_{k+1} + \beta_k(\widehat{q}_k + \beta_k\widehat{p}_k)$;
    end;
$x^* = C\widehat{x}_n$.

Definujeme-li v algoritmu 1.6

$$M = CC^T, \quad p_k = C\widehat{p}_k, \quad x_k = C\widehat{x}_k, \quad r_k = C^{-T}\widehat{r}_k,$$

a v algoritmu 1.7

$$\begin{gathered} M = CC^T, \quad \bar{p}_k = C\widehat{p}_k, \quad \bar{x}_k = C\widehat{x}_k, \quad \bar{r}_k = C^{-T}\widehat{r}_k, \\ v_k = C^{-T}\widehat{v}_k, \quad u_k = C\widehat{u}_k, \quad q_k = C\widehat{q}_k, \end{gathered}$$

dostáváme pro metodu CGM

$$\begin{aligned} \alpha_k = \frac{\widehat{r}_k^T\widehat{r}_k}{\widehat{p}_k^T\widehat{A}\widehat{p}_k} = \frac{r_k^T CC^T r_k}{p_k^T C^{-T}C^T ACC^{-1}p_k} &= \frac{r_k^T M r_k}{p_k^T A p_k}, \\ C\widehat{x}_{k+1} = C\widehat{x}_k + C\alpha_k\widehat{p}_k \Rightarrow x_{k+1} &= x_k + \alpha_k p_k, \\ C^{-T}\widehat{r}_{k+1} = C^{-T}\widehat{r}_k - \alpha_k C^{-T}C^T AC\widehat{p}_k \Rightarrow r_{k+1} &= r_k - \alpha_k A\ p_k, \\ \beta_k = \frac{\widehat{r}_{k+1}^T\widehat{r}_{k+1}}{\widehat{r}_k^T\widehat{r}_k} = \frac{r_{k+1}^T CC^T r_{k+1}}{r_k^T CC^T r_k} &= \frac{r_{k+1}^T M r_{k+1}}{r_k^T M r_k}, \\ C\widehat{p}_{k+1} = C\widehat{r}_{k+1} + \beta_k C\widehat{p}_k \Rightarrow p_{k+1} &= M r_{k+1} + \beta_k p_k, \end{aligned}$$

a pro metodu CGMS

$$\begin{aligned} C^{-T}\widehat{v}_k = C^{-T}\widehat{A}\widehat{p}_k \Rightarrow C^{-T}C^T v_k = C^{-T}C^T ACC^{-1}\bar{p}_k \Rightarrow v_k &= A\bar{p}_k, \\ \alpha_k = \frac{\widehat{r}_0^T\widehat{r}_k}{\widehat{r}_0^T\widehat{A}\widehat{p}_k} == \frac{\bar{r}_0^T CC^T\bar{r}_k}{\bar{r}_0^T CC^T ACC^{-1}\bar{p}_k} = \frac{\bar{r}_0^T M\bar{r}_k}{\bar{r}_0^T MA\bar{p}_k} &= \frac{\bar{r}_0^T M\bar{r}_k}{\bar{r}_0^T M v_k}, \\ C\widehat{q}_k = C\widehat{u}_k - \alpha_k C\widehat{v}_k \Rightarrow q_k &= u_k - \alpha_k M v_k, \\ C^{-T}\widehat{r}_{k+1} = C^{-T}\widehat{r}_k - \alpha_k C^{-T}C^T AC(C^{-1}u_k + C^{-1}q_k) \Rightarrow \bar{r}_{k+1} &= \bar{r}_k - \alpha_k A(u_k + q_k), \end{aligned}$$

$$
\begin{aligned}
C\widehat{x}_{k+1} = C\widehat{x}_k + \alpha_k(C\widehat{u}_k + C\widehat{q}_k) \Rightarrow \bar{x}_{k+1} &= \bar{x}_k + \alpha_k(u_k + q_k),\\
\beta_k = \frac{\widehat{r}_0^T \widehat{r}_{k+1}}{\widehat{r}_0^T \widehat{r}_k} = \frac{\bar{r}_0^T CC^T \bar{r}_{k+1}}{\bar{r}_0^T CC^T \bar{r}_k} &= \frac{\bar{r}_0^T M \bar{r}_{k+1}}{\bar{r}_0^T M \bar{r}_k},\\
C\widehat{u}_{k+1} = C\widehat{r}_{k+1} + \beta_k C\widehat{q}_k \Rightarrow u_{k+1} &= M\bar{r}_{k+1} + \beta_k q_k,\\
C\widehat{p}_{k+1} = C\widehat{u}_{k+1} + \beta_k(C\widehat{q}_k + \beta_k C\widehat{p}_k) \Rightarrow \bar{p}_{k+1} &= u_{k+1} + \beta_k(q_k + \beta_k \bar{p}_k).
\end{aligned}
$$

Přepsáním algoritmů 1.6 a 1.7 za použití předchozích definic dostáváme následující algoritmy.

Algoritmus 1.8

Input $x_0,\ A,\ b,\ M$;
$r_0 = b - Ax_0$;
$p_0 = Mr_0$;
for $k = 0$ **to** $n-1$ **do**
  if $r_k = 0$ **then** $x^* = x_k$ *and exit*
  else
    begin
      $\alpha_k = \frac{r_k^T M r_k}{p_k^T A p_k}$;
      $x_{k+1} = x_k + \alpha_k p_k$;
      $r_{k+1} = r_k - \alpha_k A p_k$;
      $\beta_k = \frac{r_{k+1}^T M r_{k+1}}{r_k^T M r_k}$;
      $p_{k+1} = M r_{k+1} + \beta_k p_k$
    end;
$x^* = x_n$.

Algoritmus 1.9

Input $\bar{x}_0,\ A,\ b,\ M$;
$\bar{r}_0 = b - A\bar{x}_0$;
$\bar{p}_0 = u_0 = M\bar{r}_0$;
for $k = 0$ **to** $n-1$ **do**
  if $\bar{r}_k = 0$ **then** $x^* = \bar{x}_k$ *and exit*
  else
    begin
      $v_k = A\bar{p}_k;\ \ \alpha_k = \frac{\bar{r}_0^T M \bar{r}_k}{\bar{r}_0^T M v_k}$;
      $q_k = u_k - \alpha_k M v_k$;
      $\bar{r}_{k+1} = \bar{r}_k - \alpha_k A(u_k + q_k)$;
      $\bar{x}_{k+1} = \bar{x}_k + \alpha_k(u_k + q_k)$;
      $\beta_k = \frac{\bar{r}_0^T M \bar{r}_{k+1}}{\bar{r}_0 M \bar{r}_k};\ \ u_{k+1} = M\bar{r}_{k+1} + \beta_k q_k$;
      $\bar{p}_{k+1} = u_{k+1} + \beta_k(q_k + \beta_k \bar{p}_k)$;
    end;
$x^* = \bar{x}_n$.

Poznámka : Za povšimnutí stojí, že pro rezidua a směrové vektory metody předpodmíněné metody CGM platí vztahy

$$0 = \widehat{r}_i^T \widehat{r}_j = r_i^T M r_j \quad \text{pro} \quad i \neq j$$

a

$$0 = \widehat{p}_i^T \widehat{A} \widehat{p}_j = p_i^T A p_j \quad \text{pro} \quad i \neq j.$$

Kapitola 2

Metoda bikonjugovaných gradientů

Metoda bikonjugovaných gradientů je iterační metoda na řešení soustav lineárních rovnic s libovolnou regulární maticí. Jak si ukážeme v příštím odstavci, tato metoda je založena na Lanczosově ortogonalizačním procesu.

2.1 Odvození BCG

Definice 2.1 : Předpokládejme systém lineárních rovnic v $\mathbb{R}^n$

$$Ax = b,$$

kde A je regulární matice, $b \in \mathbb{R}^n$ vektor. Proces definovaný následujícími podmínkami nazveme Lanczosův ortogonalizační proces (LOP).

1. Nechť x_0 libovolný, $\widetilde{r}_0$ nenulový vektor, $r_0 = b - Ax_0$.
2. Nechť $x_k - x_0 \in \mathcal{R}_k = sp\{r_0, Ar_0, \ldots, A^{k-1}r_0\}$ a $r_k = b - Ax_k \perp \widetilde{\mathcal{R}}_k = sp\{\widetilde{r}_0, A^T\widetilde{r}_0, \ldots, (A^T)^{k-1}\widetilde{r}_0\}$.

Definice 2.2 : Definujme pomocný vektor $\widetilde{r}_k$ pro $k \geq 1$ podmínkami

1. $\widetilde{r}_k \in \widetilde{r}_0 + sp\{A^T\widetilde{r}_0, (A^T)^2\widetilde{r}_0, \ldots, (A^T)^k\widetilde{r}_0\} = \widetilde{r}_0 + A^T \cdot \widetilde{\mathcal{R}}_k$.
2. $\widetilde{r}_k \perp \mathcal{R}_k$.

Poznámka : V dalším ukážeme, že takové vektory x_k, r_k a $\widetilde{r}_k$ za určitých podmínek existují a jsou určeny jednoznačně.

Definice 2.3 : Nechť $k \geq 1$. Potom definujme

$$H_k = \begin{pmatrix} (\widetilde{r}_0, Ar_0) & (\widetilde{r}_0, A^2r_0) & \ldots & (\widetilde{r}_0, A^kr_0) \\ (\widetilde{r}_0, A^2r_0) & (\widetilde{r}_0, A^3r_0) & \ldots & (\widetilde{r}_0, A^{k+1}r_0) \\ \vdots & & & \\ (\widetilde{r}_0, A^kr_0) & (\widetilde{r}_0, A^{k+1}r_0) & \ldots & (\widetilde{r}_0, A^{2k-1}r_0) \end{pmatrix}, \quad h^{(k)} = -\begin{pmatrix} (\widetilde{r}_0, r_0) \\ (\widetilde{r}_0, Ar_0) \\ \vdots \\ (\widetilde{r}_0, A^{k-1}r_0) \end{pmatrix},$$

$$d_k = det(H_k).$$

Lemma 2.1 : *Matici H_k lze psát ve tvaru*

$$
H_k = \begin{pmatrix} (A^T\widetilde{r}_0, r_0) & (A^T\widetilde{r}_0, Ar_0) & \dots & (A^T\widetilde{r}_0, A^{k-1}r_0) \\ ((A^T)^2\widetilde{r}_0, r_0) & ((A^T)^2\widetilde{r}_0, Ar_0) & \dots & ((A^T)^2\widetilde{r}_0, A^{k-1}r_0) \\ \vdots & & & \\ ((A^T)^k\widetilde{r}_0, r_0) & ((A^T)^k\widetilde{r}_0, Ar_0) & \dots & ((A^T)^k\widetilde{r}_0, A^{k-1}r_0) \end{pmatrix}. \tag{2.1}
$$

Lemma 2.2 : *Nechť H_k je regulární matice. Potom jsou vektory $r_0, Ar_0, \dots, A^{k-1}r_0$ i vektory $\widetilde{r}_0, A^T\widetilde{r}_0, \dots, (A^T)^{k-1}\widetilde{r}_0$ lineárně nezávislé.*

Důkaz : Nechť jsou vektory $r_0, Ar_0, \dots, A^{k-1}r_0$ lineárně závislé. Potom existuje index i $(0 \le i \le k-1)$ takový, že platí

$$
A^i r_0 = \sum_{j=0\ j\neq i}^{k-1} c_j A^j r_0, \qquad \sum_{j=0\ j\neq i}^{k-1} c_j^2 \neq 0. \tag{2.2}
$$

Uvažujeme-li matici H_k ve tvaru (2.1) je vidět, že lze rozepsat $A^i r_0$ pomocí (2.2), neboli $i+1$ sloupec je netriviální lineární kombinací ostatních sloupců, matice H_k je tedy singulární, což je spor. Podobně lze dojít ke sporu při předpokladu lineární závislosti vektorů $\widetilde{r}_0, A^T\widetilde{r}_0, \dots, (A^T)^{k-1}\widetilde{r}_0$. ■

Lemma 2.3 : *Nechť $d_k \neq 0$. Potom vektory r_k, $\widetilde{r}_k$, x_k existují a jsou určeny jednoznačně. Uvažujeme-li x_k ve tvaru*

$$
x_k = x_0 - \sum_{i=1}^{k} b_i^{(k)} A^{i-1} r_0, \tag{2.3}
$$

kde $b_i^{(k)} \in \mathbb{R}$, $i = 1, \dots, k$, můžeme r_k a $\widetilde{r}_k$ psát ve tvaru

$$
r_k = r_0 + \sum_{i=1}^{k} b_i^{(k)} A^i r_0 \quad \text{a} \quad \widetilde{r}_k = \widetilde{r}_0 + \sum_{i=1}^{k} b_i^{(k)} (A^T)^i \widetilde{r}_0. \tag{2.4}
$$

Důkaz : Protože $x_k - x_0 \in \mathcal{R}_k$, existují koeficienty $b_1^{(k)}, b_2^{(k)}, \dots, b_k^{(k)}$ takové, že

$$
x_k - x_0 = -\sum_{i=1}^{k} b_i^{(k)} A^{i-1} r_0. \tag{2.5}
$$

Z (2.5) plyne, že

$$
r_k = r_0 + \sum_{i=1}^{k} b_i^{(k)} A^i r_0. \tag{2.6}
$$

Vektor $\widetilde{r}_k$ můžeme dle definice uvažovat ve tvaru

$$
\widetilde{r}_k = \widetilde{r}_0 + \sum_{i=1}^{k} c_i^{(k)} (A^T)^i \widetilde{r}_0. \tag{2.7}
$$

Podmínky $r_k \perp \widetilde{\mathcal{R}}_k$ a $\widetilde{r}_k \perp \mathcal{R}_k$ lze psát ve tvaru

$$
\begin{aligned}
(r_k, (A^T)^i\widetilde{r}_0) &= 0, \quad i = 0, \dots, k-1, &(2.8)\\
(\widetilde{r}_k, A^i r_0) &= 0, \quad i = 0, \dots, k-1. &(2.9)
\end{aligned}
$$

Označme nyní $b^{(k)} = (b_1^{(k)}, b_2^{(k)}, \ldots, b_k^{(k)})^T$ a $c^{(k)} = (c_1^{(k)}, c_2^{(k)}, \ldots, c_k^{(k)})^T$. Dosadíme-li nyní (2.6) do (2.8) a (2.7) do (2.9) a využijeme-li toho, že $((A^T)^i \widetilde{r}_0, A^j r_0) = (\widetilde{r}_0, A^{i+j} r_0)$, dostáváme soustavy rovnic

$$h^{(k)} = H_k b^{(k)} \quad \text{a} \quad h^{(k)} = H_k c^{(k)}. \tag{2.10}$$

Protože $d_k \neq 0$, je H_k regulární. Řešení soustav (2.10) tedy existuje a je určeno jednoznačně. Odsud a z lemmatu 2.2 již plyne existence a jednoznačnost vektorů r_k, $\widetilde{r}_k$ a x_k a navíc z jednoznačnosti určení řešení soustav (2.10) máme $b^{(k)} = c^{(k)}$. ■

Poznámka : Může se tedy stát (je-li H_k singulární), že neexistuje x_k takové, aby byly splněny podmínky 1. a 2. v definici LOP. Potom dochází k zhroucení LOP. Je však možné, jak ukážeme v příští kapitole, přeskočit případy, kdy je H_k singulární, a sestrojit až x_m $(m > k)$, kdy je H_m regulární.

Lemma 2.4 : *Je-li v každém kroku $d_k \neq 0$, najde LOP řešení soustavy $Ax = b$ (A regulární) nejvýše po n-krocích.*

Důkaz : $d_n \neq 0 \Rightarrow dim(\widetilde{\mathcal{R}_n}) = n \Rightarrow \widetilde{\mathcal{R}_n} = \mathbb{R}^n \Rightarrow r_n \perp \mathbb{R}^n \Rightarrow r_n = 0.$ ■

Poznámka : Mluvíme-li v dalším o r_i resp. o $\widetilde{r}_i$, předpokládejme implicitně, že r_i resp. $\widetilde{r}_i$ existuje a $0 \leq i \leq n$.

Lemma 2.5 : Pro vektory $\widetilde{r}_i^T$ a r_j platí

$$\widetilde{r}_i^T r_j = 0 \quad \text{pro} \quad i \neq j. \tag{2.11}$$

Důkaz : Nechť $i < j$. Potom $\widetilde{r}_i \in \widetilde{\mathcal{R}}_{i+1}$ a

$$i + 1 \leq j \Rightarrow r_j \perp \widetilde{\mathcal{R}}_{i+1} \Rightarrow \widetilde{r}_i^T r_j = 0.$$

Pro $i > j$ analogicky. ■

Lemma 2.6 : *Nechť $\widetilde{r}_i^T r_i \neq 0$ pro $i = 0, \ldots, k$. Potom jsou rezidua $r_0, \ldots, r_k$ a vektory $\widetilde{r}_0, \ldots, \widetilde{r}_k$ lineárně nezávislé, neboli*

$$sp\{r_0, \ldots, r_k\} = \mathcal{R}_{k+1} \quad \text{a} \quad sp\{\widetilde{r}_0, \ldots, \widetilde{r}_k\} = \widetilde{\mathcal{R}}_{k+1}.$$

Důkaz : Přenásobíme-li rovnost

$$\sum_{i=0}^{k} c_i r_i = 0, \qquad c_i \in \mathbb{R}, \tag{2.12}$$

vektorem $\widetilde{r}_j^T$ pro $j = 0, \ldots, k$, dostáváme za použití lemmatu 2.5

$$c_j \widetilde{r}_j^T r_j = 0.$$

Protože $\widetilde{r}_j^T r_j \neq 0$, je $c_j = 0$ pro $j = 0, \ldots, k$. (2.12) je tedy splněno jen tehdy, je-li $c_j = 0$ pro $j = 0, \ldots, k$ neboli vektory $r_0, \ldots, r_k$ jsou lineárně nezávislé. Podobně by se dokazovala lineární nezávislost vektorů $\widetilde{r}_0, \ldots, \widetilde{r}_k$. ■

Lemma 2.7 : *Nechť* $d_j \neq 0$ *pro* $j = 1, \ldots, k+1$ *a nechť* $\tilde{r}_i^T r_i \neq 0$ *pro* $i = 0, \ldots, k$. *Potom existují a jsou jednoznačně určeny koeficienty* $\gamma_j^{(i)}$ *pro* $0 \leq j \leq i \leq k$ *takové, že* $\gamma_j^{(i)} \neq 0$ *a*

$$
\begin{aligned}
(2.13) \qquad r_{i+1} &= r_i + A(\gamma_i^{(i)} r_i + \gamma_{i-1}^{(i)} r_{i-1} + \ldots + \gamma_1^{(i)} r_1 + \gamma_0^{(i)} r_0), \\
(2.14) \qquad \tilde{r}_{i+1} &= \tilde{r}_i + A^T(\gamma_i^{(i)} \tilde{r}_i + \gamma_{i-1}^{(i)} \tilde{r}_{i-1} + \ldots + \gamma_1^{(i)} \tilde{r}_1 + \gamma_0^{(i)} \tilde{r}_0).
\end{aligned}
$$

Důkaz : Platí

$$
\begin{aligned}
r_{i+1} &= r_0 + b_1^{(i+1)} A\, r_0 + b_2^{(i+1)} A^2 r_0 + \ldots + b_{i+1}^{(i+1)} A^{i+1} r_0, \\
r_i &= r_0 + b_1^{(i)} A\, r_0 + b_2^{(i)} A^2 r_0 + \ldots + b_i^{(i)} A^i r_0, \\
r_{i+1} &= r_i + A((b_1^{(i+1)} - b_1^{(i)}) r_0 + \ldots + (b_i^{(i+1)} - b_i^{(i)}) A^{i-1} r_0 + b_{i+1}^{(i+1)} A^i r_0).
\end{aligned}
$$

Dle předchozího lemmatu existují a jsou jednoznačně určeny koeficienty $\gamma_0^{(i)}, \ldots, \gamma_i^{(i)}$ takové, že

$$(2.15) \qquad r_{i+1} = r_i + A(\gamma_i^{(i)} r_i + \gamma_{i-1}^{(i)} r_{i-1} + \ldots + \gamma_1^{(i)} r_1 + \gamma_0^{(i)} r_0).$$

Analogicky dostaneme (2.14) .

Zbývá dokázat nenulovost koeficientů $\gamma_j^{(i)}$.

Označme $c_{ij} = \tilde{r}_i^T A r_j$. Zřejmě je $c_{ij} = 0$ je-li $\mid i - j \mid > 1$. Pro $i = 0, \ldots, k$ máme

$$
\begin{aligned}
(2.16) \qquad r_i &= r_{i-1} + A\; (\gamma_{i-1}^{(i-1)} r_{i-1} + \ldots + \gamma_0^{(i-1)} r_0) \qquad / \tilde{r}_i^T \cdot, \\
\tilde{r}_i^T r_i &= \gamma_{i-1}^{(i-1)} \tilde{r}_i^T A r_{i-1}.
\end{aligned}
$$

Jelikož je $\tilde{r}_i^T r_i \neq 0$ je $\tilde{r}_i^T A r_{i-1} \neq 0$ neboli $c_{i\ i-1} \neq 0$. Podobně vyjádříme-li si $\tilde{r}_i$ ve tvaru (2.14) dostáváme $c_{i-1\ i} \neq 0$.

Násobíme-li rovnost

$$r_{i+1} = r_i + A\; (\gamma_i^{(i)} r_i + \ldots + \gamma_0^{(i)} r_0)$$

postupně vektory $\tilde{r}_i^T, \ldots, \tilde{r}_0^T$ dostáváme pro koeficienty $\gamma_l^{(i)}$ soustavu rovnic

$$
(2.17) \qquad -\begin{pmatrix} \tilde{r}_i^T r_i \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} c_{i\ i} & c_{i\ i-1} & & & \\ c_{i-1\ i} & c_{i-1\ i-1} & c_{i-1\ i-2} & & \\ & c_{i-2\ i-1} & c_{i-2\ i-2} & c_{i-2\ i-3} & \\ & \ddots & \ddots & \ddots & \\ & & c_{12} & c_{11} & c_{10} \\ & & & c_{01} & c_{00} \end{pmatrix} \begin{pmatrix} \gamma_i^{(i)} \\ \gamma_{i-1}^{(i)} \\ \vdots \\ \gamma_0^{(i)} \end{pmatrix}.
$$

Z existence koeficientů $\gamma_l^{(i)}$ plyne existence řešení dané soustavy. Z jednoznačnosti jejich určení pak plyne regularita matice dané soustavy (singulární $\Rightarrow$ r_k daných vlastností není určeno jednoznačně $\Rightarrow$ spor s předpoklady). Označíme-li třídiagonální matici soustavy (2.17) jako V_i, je $det(V_i) \neq 0$ pro $i = 0, \ldots, k$. Označme $V_i^{(j)}$ pro $j = 1, \ldots, i+1$ matici, která vznikne z matice V_i tak, že nahradíme j-tý sloupec matice V_i sloupcem levé strany soustavy (2.17) . Potom dle Cramerova pravidla pro $j = 0, \ldots, i$ dostáváme

$$\gamma_j^{(i)} = \frac{det(V_i^{(i+1-j)})}{det(V_i)}.$$

Dodefinujeme-li $V_{-1} = 1$, můžeme $det(V_i^{(j)})$ psát jako

$$(2.18) \qquad det(V_i^{(j)}) = (-1)^j \cdot \tilde{r}_i^T r_i \cdot \prod_{l=i-j+1}^{i-1} c_{l\ l+1} \cdot det(V_{i-j}), \quad j = 1, \ldots, i+1.$$

Z (2.18) plyne $det(V_i^{(i+1-j)}) \neq 0$ a tedy $\gamma_j^{(i)} \neq 0$ pro $j = 0, \ldots, i$. ■

Lemma 2.8 : Nechť $d_i \neq 0$ pro $i = 1, \ldots, k+1$, nechť $\tilde{r}_i^T r_i \neq 0$ pro $i = 0, \ldots, k$ a nechť $p_0 = r_0$ a $\tilde{p}_0 = \tilde{r}_0$. Potom lze vektory r_{i+1}, $\tilde{r}_{i+1}$ a x_{i+1} pro $i = 0, \ldots, k$ počítat dle vztahů

$$(2.19) \qquad r_{i+1} = r_i - \alpha_i A\, p_i,$$
$$(2.20) \qquad \tilde{r}_{i+1} = \tilde{r}_i - \alpha_i A^T \tilde{p}_i,$$
$$(2.21) \qquad p_{i+1} = r_{i+1} + \beta_i p_i,$$
$$(2.22) \qquad \tilde{p}_{i+1} = \tilde{r}_{i+1} + \beta_i \tilde{p}_i,$$
$$(2.23) \qquad x_{i+1} = x_i + \alpha_i p_i,$$

kde

$$(2.24) \qquad \alpha_i = \frac{\tilde{r}_i^T r_i}{\tilde{p}_i^T A p_i}, \quad \beta_i = \frac{\tilde{r}_{i+1}^T r_{i+1}}{\tilde{r}_i^T r_i}.$$

Důkaz : Vyjdeme-li z tvaru (2.16) (víme, že $\gamma_j^{(i)} \neq 0$ pro $j \leq i \leq k$) můžeme psát

$$\begin{aligned} r_{i+1} &= r_i + \gamma_i^{(i)} A(r_i + \frac{\gamma_{i-1}^{(i)}}{\gamma_i^{(i)}} r_{i-1} + \frac{\gamma_{i-2}^{(i)}}{\gamma_i^{(i)}} r_{i-2} + \ldots + \frac{\gamma_0^{(i)}}{\gamma_i^{(i)}} r_0) = \\ &= r_i + \gamma_i^{(i)} A(r_i + \frac{\gamma_{i-1}^{(i)}}{\gamma_i^{(i)}}(r_{i-1} + \frac{\gamma_{i-2}^{(i)}}{\gamma_{i-1}^{(i)}} r_{i-2} + \ldots + \frac{\gamma_0^{(i)}}{\gamma_{i-1}^{(i)}} r_0)) = \\ (2.25) \qquad &= r_i + \gamma_i^{(i)} A(r_i + \frac{\gamma_{i-1}^{(i)}}{\gamma_i^{(i)}}(r_{i-1} + \frac{\gamma_{i-2}^{(i)}}{\gamma_{i-1}^{(i)}}(r_{i-2} + \ldots \frac{\gamma_1^{(i)}}{\gamma_2^{(i)}}(r_1 + \frac{\gamma_0^{(i)}}{\gamma_1^{(i)}} r_0) \ldots))). \end{aligned}$$

Zřejmě lze ve tvaru (2.25) rozepsat každé reziduum pro $i = 0, \ldots, k$. Dokažme si nyní vztah

$$(2.26) \qquad \frac{\gamma_{i-1}^{(l)}}{\gamma_i^{(l)}} = \frac{\gamma_{i-1}^{(j)}}{\gamma_i^{(j)}} \quad \text{pro} \quad 1 \leq i \leq l \leq k, \quad 1 \leq i \leq j \leq k.$$

Platí

$$\frac{\gamma_{i-1}^{(l)}}{\gamma_i^{(l)}} = \frac{det(V_l^{(l+2-i)})}{det(V_l^{(l+1-i)})} = \frac{(-1)^{l+2-i} \cdot \tilde{r}_l^T \tilde{r}_l \cdot \prod_{j=i-1}^{l-1} c_{j\ j+1} det(V_{i-2})}{(-1)^{l+1-i} \cdot \tilde{r}_l^T \tilde{r}_l \cdot \prod_{j=i}^{l-1} c_{j\ j+1} det(V_{i-1})} = -c_{i-1\ i} \cdot \frac{det(V_{i-2})}{det(V_{i-1})}.$$

Vidíme, že pravá strana nezávisí na l. Tedy zřejmě též

$$\frac{\gamma_{i-1}^{(j)}}{\gamma_i^{(j)}} = -c_{i-1\ i} \cdot \frac{det(V_{i-2})}{det(V_{i-1})}$$

a (2.26) platí. Definujme nyní koeficienty α_i a β_i jako

$$\alpha_i = -\gamma_i^{(i)} \quad \text{a} \quad \beta_{i-1} = \frac{\gamma_{i-1}^{(i)}}{\gamma_i^{(i)}}.$$

Potom s využitím vztahů (2.26) a (2.25) lze pro $i = 0, \dots, k$ psát reziduum r_{i+1} ve tvaru

$$(2.27) \qquad r_{i+1} = r_i - \alpha_i A(r_i + \beta_{i-1}(r_{i-1} + \beta_{i-2}(r_{i-2} + \dots \beta_1(r_1 + \beta_0 r_0) \dots))).$$

Definujeme-li nyní

$$p_0 = r_0, \quad p_{j+1} = r_{j+1} + \beta_j p_j \quad \text{pro} \quad j = 0, \dots, i-1,$$

můžeme (2.27) přepsat ve tvaru

$$(2.28) \qquad r_{i+1} = r_i - \alpha_i A p_i \quad \text{pro} \quad i = 0, \dots, k.$$

Zbývá již jen odvodit tvary koeficientů α_i a β_i pro $i = 0, \dots, k$. Platí

$$\begin{aligned} r_{i+1} &= r_i - \alpha_i A\, p_i \qquad \Big/ \; \widetilde{p}_i^T \cdot, \\ 0 &= \widetilde{p}_i^T r_i - \alpha_i \widetilde{p}_i^T A p_i, \\ \alpha_i &= \frac{\widetilde{p}_i^T r_i}{\widetilde{p}_i^T A p_i}. \end{aligned}$$

Protože je

$$\widetilde{p}_i^T r_i = \widetilde{r}_i^T r_i + \beta_{i-1} \widetilde{p}_{i-1}^T r_i = \widetilde{r}_i^T r_i,$$

dostáváme

$$\alpha_i = \frac{\widetilde{r}_i^T r_i}{\widetilde{p}_i^T A p_i}.$$

Pokračujme dále v odvození tvaru β_i. Platí

$$\begin{aligned} p_{i+1} &= r_{i+1} + \beta_i p_i \qquad \Big/ \; \widetilde{r}_i^T \cdot, \\ \widetilde{r}_i^T p_{i+1} &= \beta_i \widetilde{r}_i^T r_i, \\ \beta_i &= \frac{\widetilde{r}_i^T p_{i+1}}{\widetilde{r}_i^T r_i}. \end{aligned}$$

Protože je

$$\widetilde{r}_{i+1}^T r_{i+1} = \widetilde{r}_{i+1}^T p_{i+1} = \widetilde{r}_i^T p_{i+1} - \alpha_i \widetilde{p}_i^T A p_{i+1} = \widetilde{r}_i^T p_{i+1} + \frac{\alpha_i}{\alpha_{i-1}} \widetilde{p}_i^T (r_{i+2} - r_{i+1}) = \widetilde{r}_i^T p_{i+1},$$

dostáváme

$$\beta_i = \frac{\widetilde{r}_{i+1}^T r_{i+1}}{\widetilde{r}_i^T r_i}.$$

Vztahy (2.20) a (2.22) by se dokazovaly analogicky. Vztah (2.23) dostáváme z

$$b - A x_{i+1} = b - A x_i - \alpha_i A\, p_i. \qquad \blacksquare$$

Poznámka : Na základě předchozího lemmatu můžeme zformulovat následující algoritmus.

Algoritmus 2.1

Input $x_0,\ A,\ b,\ \widetilde{r}_0;$
$r_0 = b - Ax_0;$
$p_0 = r_0;\ \widetilde{p}_0 = \widetilde{r}_0;$
for $k = 0$ **to** $n-1$ **do**
  if $r_k = 0$ **then** $x^* = x_k$ *and exit*
  else
    begin
      $\alpha_k = \frac{\widetilde{r}_k^T r_k}{\widetilde{p}_k^T A p_k};$
      $x_{k+1} = x_k + \alpha_k p_k;$
      $r_{k+1} = r_k - \alpha_k A p_k;\ \widetilde{r}_{k+1} = \widetilde{r}_k - \alpha_k A^T \widetilde{p}_k;$
      $\beta_k = \frac{\widetilde{r}_{k+1}^T r_{k+1}}{\widetilde{r}_k^T r_k};$
      $p_{k+1} = r_{k+1} + \beta_k p_k;\ \widetilde{p}_{k+1} = \widetilde{r}_{k+1} + \beta_k \widetilde{p}_k;$
    end;
$x^* = x_n.$

Definice 2.4 : Metodu definovanou algoritmem 2.1 nazveme metodou bikonjugovaných gradientů (BCG).

Poznámka : Povšimněme si, že je-li A symetrická a pozitivně definitní a volíme-li $\widetilde{r}_0 = r_0$, dostáváme metodu sdružených gradientů.

Ukažme si nyní další vlastnosti vektorů, generovaných metodou BCG.

Lemma 2.9 : Platí

$$r_j^T \widetilde{r}_i = \widetilde{r}_j^T r_i = 0 \quad \text{pro} \quad i \neq j, \tag{2.29}$$

$$\widetilde{p}_j^T r_i = p_j^T \widetilde{r}_i = 0 \quad \text{pro} \quad j < i, \tag{2.30}$$

$$p_j^T A^T \widetilde{p}_i = \widetilde{p}_j^T A p_i = 0 \quad \text{pro} \quad i \neq j. \tag{2.31}$$

Důkaz : Vztah (2.29) jsme dokázali již v lemmatu 2.5.
Vztah (2.30) plyne z toho, že $\widetilde{p}_j \in \widetilde{\mathcal{R}}_{j+1}$ a $r_i \perp \widetilde{\mathcal{R}}_{j+1}$ protože $j+1 \leq i$. Podobně $p_j \in \mathcal{R}_{j+1}$ a $\widetilde{r}_i \perp \mathcal{R}_{j+1}$ protože $j+1 \leq i$.
Vztah (2.31) dokážeme indukcí. Nechť $0 \leq j < n$ potom

$$\widetilde{p}_j^T A p_{j+1} = -\frac{1}{\alpha_j}(\widetilde{r}_{j+1} - \widetilde{r}_j)^T r_{j+1} + \beta_j \widetilde{p}_j^T A p_j = -\frac{\widetilde{p}_j^T A p_j}{\widetilde{r}_j^T r_j}\widetilde{r}_{j+1}^T r_{j+1} + \frac{\widetilde{r}_{j+1}^T r_{j+1}}{\widetilde{r}_j^T r_j}\widetilde{p}_j^T A p_j = 0,$$

$$p_j^T A^T \widetilde{p}_{j+1} = -\frac{1}{\alpha_j}(r_{j+1} - r_j)^T \widetilde{r}_{j+1} + \beta_j p_j^T A^T \widetilde{p}_j = -\frac{\widetilde{p}_j^T A p_j}{\widetilde{r}_j^T r_j}\widetilde{r}_{j+1}^T r_{j+1} + \frac{\widetilde{r}_{j+1}^T r_{j+1}}{\widetilde{r}_j^T r_j}\widetilde{p}_j^T A p_j = 0.$$

Nechť nyní i je takové, že $j < i < n$. Předpokládejme, že $\widetilde{p}_j^T A p_i = 0$ a $p_j^T A^T \widetilde{p}_i = 0$. Ukážeme, že i $\widetilde{p}_j^T A p_{i+1} = 0$ a $p_j^T A^T \widetilde{p}_{i+1} = 0$. Platí

$$\widetilde{p}_j^T A p_{i+1} = \widetilde{p}_j^T A r_{i+1} + \beta_i \widetilde{p}_j^T A p_i = -\frac{1}{\alpha_j}(\widetilde{r}_{j+1} - \widetilde{r}_j) r_{i+1} + \beta_i \widetilde{p}_j^T A p_i = 0,$$

$$p_j^T A^T \widetilde{p}_{i+1} = p_j^T A^T \widetilde{r}_{i+1} + \beta_i p_j^T A^T \widetilde{p}_i = -\frac{1}{\alpha_j}(r_{j+1} - r_j)\widetilde{r}_{i+1} + \beta_i p_j^T A^T \widetilde{p}_i = 0.$$

Máme dokázáno (2.31) pro $j < i$. Pro $j > i$ dostáváme

$$p_j^T A^T \widetilde{p}_i = \widetilde{p}_i^T A p_j = 0 \quad \text{a} \quad \widetilde{p}_j^T A p_i = p_i^T A^T \widetilde{p}_j = 0. \qquad \blacksquare$$

2.2 Umocnění metody BCG

Umocnění metody BCG se provede analogickým způsobem jako umocnění metody CGM. Umocnění BCG má tu dobrou vlastnost, že se vyhneme použití matice A^T. Z algoritmů metod BCG a CGS lze poté vyčíst, že umocnění BCG nepřinese zdvojnásobení operací jako u metody CGM.

Lemma 2.10 : Existují maticové polynomy $\varphi(A)$ a $\psi(A)$ stupně nejvýše k takové, že

$$r_k = \varphi_k(A) r_0, \tag{2.32}$$
$$\widetilde{r}_k = \varphi_k^T(A) \widetilde{r}_0, \tag{2.33}$$
$$p_k = \psi_k(A) r_0, \tag{2.34}$$
$$\widetilde{p}_k = \psi_k^T(A) \widetilde{r}_0. \tag{2.35}$$

Důkaz : Plyne z (2.4), (2.13) a (2.14). ■

Lemma 2.11 : Polynomy $\varphi_k(A)$ a $\psi_k(A)$ lze počítat dle následujícího algoritmu.

Algoritmus 2.2

Input $x_0,\ A,\ b,\ \widetilde{r}_0$;
$\varphi_0(A) = I;\ \psi_0(A) = I;\ r_0 = b - Ax_0;$
for $i = 0$ **to** $k-1$ **do**
  begin
    $\alpha_i = \frac{\widetilde{r}_0^T \varphi_i^2(A) r_0}{\widetilde{r}_0^T A \psi_i^2(A) r_0};$
    $\varphi_{i+1}(A) = \varphi_i(A) - \alpha_i A \psi_i(A);$
    $\beta_i = \frac{\widetilde{r}_0^T \varphi_{i+1}^2(A) r_0}{\widetilde{r}_0^T \varphi_i^2(A) r_0};$
    $\psi_{i+1}(A) = \varphi_{i+1}(A) + \beta_i \psi_i(A);$
  end.

Důkaz : Plyne z algoritmu 2.1. ■

Použijeme-li nyní lemmatu 1.15, definice 1.6 a lemmatu 1.16 můžeme formulovat následující algoritmus. (Jediný rozdíl v odvození je ve tvaru koeficientů α_k a β_k.)

Algoritmus 2.3

Input $\bar{x}_0,\ A,\ b,\ \widetilde{r}_0$;
$\bar{r}_0 = b - A\bar{x}_0;$
$\bar{p}_0 = u_0 = \bar{r}_0;$
for $k = 0$ **to** $n-1$ **do**
  if $\bar{r}_k = 0$ **then** $x^* = \bar{x}_k$ *and exit*
  else
    begin

$v_k = A\bar{p}_k;$
$\alpha_k = \frac{\widetilde{r}_0^T \bar{r}_k}{\widetilde{r}_0^T v_k};$
$q_k = u_k - \alpha_k v_k;$
$\bar{r}_{k+1} = \bar{r}_k - \alpha_k A(u_k + q_k);$
$\bar{x}_{k+1} = \bar{x}_k + \alpha_k(u_k + q_k);$
$\beta_k = \frac{\widetilde{r}_0^T \bar{r}_{k+1}}{\widetilde{r}_0^T \bar{r}_k};$
$u_{k+1} = \bar{r}_k + \beta_k q_k;$
$p_{k+1} = u_{k+1} + \beta_k(q_k + \beta_k \bar{p}_k);$
end;
$x^* = \bar{x}_n.$

Definice 2.5 : Metodu definovanou algoritmem 2.3 nazveme umocněnou metodou bikonjugovaných gradientů (CGS).

2.3 Předpodmínění BCG a CGS

Předpodmínění metod BCG a CGS se provede analogicky jako u metod CGM a CGMS. Nechť C a D jsou regulární matice. Přenásobíme-li rovnici (2.1) maticí C zleva, dostáváme

$$CADD^{-1}x = Cb.$$

Provedeme-li nyní substituci

$$\widehat{A} = CAD, \quad \widehat{x} = D^{-1}x, \quad \widehat{b} = Cb,$$

dostáváme systém

$$\widehat{A}\widehat{x} = \widehat{b}. \tag{2.36}$$

Mezi řešením soustavy (2.1) a soustavy (2.36) platí vztah $\widehat{x} = D^{-1}x$. Je-li x_0 počáteční odhad soustavy (2.1), definujme počáteční odhad soustavy (2.36) jako $\widehat{x}_0 = D^{-1}x_0$.

Na soustavu (2.36) můžeme nyní použít metodu BCG a CGS. V algoritmech obou metod se podobně jako v podkapitole 1.5 píší všude místo normálních vektorů vektory se stříškou. Proveďme nyní pouze odvození rekurentních vztahů předpodmíněných metod.

Definujme v metodě BCG aplikované na matici $\widehat{A}$

$$M = DC, \quad p_k = D\widehat{p}_k, \quad \widetilde{p}_k = C^T\widehat{\widetilde{p}}_k, \quad x_k = D\widehat{x}_k, \quad r_k = C^{-1}\widehat{r}_k, \quad \widetilde{r}_k = D^{-T}\widehat{\widetilde{r}}_k.$$

Dostáváme

$$\begin{aligned}
\alpha_k = \frac{\widehat{\widetilde{r}}_k^T \widehat{r}_k}{\widehat{\widetilde{p}}_k^T \widehat{A}\widehat{p}_k} = \frac{\widetilde{r}_k^T DCr_k}{\widetilde{p}_k^T C^{-1}CADD^{-1}p_k} &= \frac{\widetilde{r}_k^T Mr_k}{\widetilde{p}_k^T Ap_k}, \\
D\widehat{x}_{k+1} = D\widehat{x}_k + D\alpha_k\widehat{p}_k \Rightarrow x_{k+1} &= x_k + \alpha_k p_k, \\
C^{-1}\widehat{r}_{k+1} = C^{-1}\widehat{r}_k - \alpha_k C^{-1}CAD\widehat{p}_k \Rightarrow r_{k+1} &= r_k - \alpha_k A\, p_k, \\
D^{-T}\widehat{\widetilde{r}}_{k+1} = D^{-T}\widehat{\widetilde{r}}_k - \alpha_k D^{-T}D^TA^TC^T\widehat{\widetilde{p}}_k \Rightarrow \widetilde{r}_{k+1} &= \widetilde{r}_k - \alpha_k A^T\, \widetilde{p}_k, \\
\beta_k = \frac{\widehat{\widetilde{r}}_{k+1}^T \widehat{r}_{k+1}}{\widehat{\widetilde{r}}_k^T \widehat{r}_k} = \frac{\widetilde{r}_{k+1}^T DCr_{k+1}}{\widetilde{r}_k^T DCr_k} &= \frac{\widetilde{r}_{k+1}^T Mr_{k+1}}{\widetilde{r}_k^T Mr_k}, \\
D\widehat{p}_{k+1} = D\widehat{r}_{k+1} + \beta_k D\widehat{p}_k \Rightarrow p_{k+1} &= Mr_{k+1} + \beta_k p_k, \\
C^T\widehat{\widetilde{p}}_{k+1} = C^T\widehat{\widetilde{r}}_{k+1} + \beta_k C^T\widehat{\widetilde{p}}_k \Rightarrow \widetilde{p}_{k+1} &= M^T\widetilde{r}_{k+1} + \beta_k\widetilde{p}_k.
\end{aligned}$$

Definujeme-li v metodě CGS aplikovné na $\widehat{A}$

$$\begin{array}{cccc} M = DC, & \bar{p}_k = D\widehat{p}_k, & \bar{x}_k = D\widehat{x}_k, & \bar{r}_k = C^{-1}\widehat{r}_k, \\ v_k = C^{-1}\widehat{v}_k, & u_k = D\widehat{u}_k, & q_k = D\widehat{q}_k, & \widetilde{r}_0 = D^{-T}\widehat{\widetilde{r}}_0, \end{array}$$

dostáváme

$$\begin{aligned}
C^{-1}\widehat{v}_k = C^{-1}\widehat{A}\widehat{p}_k \Rightarrow C^{-1}Cv_k = C^{-1}CADD^{-1}\bar{p}_k \Rightarrow v_k &= A\bar{p}_k, \\
\alpha_k = \frac{\widehat{\widetilde{r}}_0^T\widehat{r}_k}{\widehat{\widetilde{r}}_0^T\widehat{A}\widehat{p}_k} = \frac{\widetilde{r}_0^T DC\bar{r}_k}{\widetilde{r}_0^T DCADD^{-1}\bar{p}_k} = \frac{\widetilde{r}_0^T M\bar{r}_k}{\widetilde{r}_0^T MA\bar{p}_k} &= \frac{\widetilde{r}_0^T M\bar{r}_k}{\widetilde{r}_0^T Mv_k}, \\
D\widehat{q}_k = D\widehat{u}_k - \alpha_k D\widehat{v}_k \Rightarrow q_k &= u_k - \alpha_k M v_k, \\
C^{-1}\widehat{r}_{k+1} = C^{-1}\widehat{r}_k - \alpha_k C^{-1}CAD(D^{-1}u_k + D^{-1}q_k) \Rightarrow \bar{r}_{k+1} &= \bar{r}_k - \alpha_k A(u_k + q_k), \\
D\widehat{x}_{k+1} = D\widehat{x}_k + \alpha_k(D\widehat{u}_k + D\widehat{q}_k) \Rightarrow \bar{x}_{k+1} &= \bar{x}_k + \alpha_k(u_k + q_k), \\
\beta_k = \frac{\widehat{\widetilde{r}}_0^T\widehat{r}_{k+1}}{\widehat{\widetilde{r}}_0^T\widehat{r}_k} = \frac{\widetilde{r}_0^T DC\bar{r}_{k+1}}{\widetilde{r}_0^T DC\bar{r}_k} &= \frac{\widetilde{r}_0^T M\bar{r}_{k+1}}{\widetilde{r}_0^T M\bar{r}_k}, \\
D\widehat{u}_{k+1} = D\widehat{r}_{k+1} + \beta_k D\widehat{q}_k \Rightarrow u_{k+1} &= M\bar{r}_{k+1} + \beta_k q_k, \\
D\widehat{p}_{k+1} = D\widehat{u}_{k+1} + \beta_k(D\widehat{q}_k + \beta_k D\widehat{p}_k) \Rightarrow \bar{p}_{k+1} &= u_{k+1} + \beta_k(q_k + \beta_k\bar{p}_k).
\end{aligned}$$

Algoritmus 2.4

Input $x_0,\ A,\ b,\ \widetilde{r}_0,\ M$;
$r_0 = b - Ax_0$;
$p_0 = Mr_0$; $\widetilde{p}_0 = M^T\widetilde{r}_0$;
for $k = 0$ **to** $n-1$ **do**
  if $r_k = 0$ **then** $x^* = x_k$ *and exit*
  else
    begin
      $\alpha_k = \frac{\widetilde{r}_k^T Mr_k}{\widetilde{p}_k^T Ap_k}$;
      $x_{k+1} = x_k + \alpha_k p_k$;
      $r_{k+1} = r_k - \alpha_k Ap_k$;
      $\widetilde{r}_{k+1} = \widetilde{r}_k - \alpha_k A^T\widetilde{p}_k$;
      $\beta_k = \frac{\widetilde{r}_{k+1}^T Mr_{k+1}}{\widetilde{r}_k^T Mr_k}$;
      $p_{k+1} = Mr_{k+1} + \beta_k p_k$;
      $\widetilde{p}_{k+1} = M^T\widetilde{r}_{k+1} + \beta_k\widetilde{p}_k$;
    end;
$x^* = x_n$.

Algoritmus 2.5

Input $\bar{x}_0,\ A,\ b,\ \widetilde{r}_0,\ M$;
$\bar{r}_0 = b - A\bar{x}_0$;
$\bar{p}_0 = u_0 = M\bar{r}_0$;
for $k = 0$ **to** $n-1$ **do**
  if $\bar{r}_k = 0$ **then** $x^* = \bar{x}_k$ *and exit*
  else
    begin
      $v_k = A\bar{p}_k$; $\alpha_k = \frac{\widetilde{r}_0^T M\bar{r}_k}{\widetilde{r}_0^T Mv_k}$;
      $q_k = u_k - \alpha_k Mv_k$;
      $\bar{r}_{k+1} = \bar{r}_k - \alpha_k A(u_k + q_k)$;
      $\bar{x}_{k+1} = \bar{x}_k + \alpha_k(u_k + q_k)$;
      $\beta_k = \frac{\widetilde{r}_0^T M\bar{r}_{k+1}}{\widetilde{r}_0^T M\bar{r}_k}$;
      $u_{k+1} = M\bar{r}_{k+1} + \beta_k q_k$;
      $\bar{p}_{k+1} = u_{k+1} + \beta_k(q_k + \beta_k\bar{p}_k)$;
    end;
$x^* = \bar{x}_n$.

Poznámka : Matice C a D se snažíme volit tak, aby $CAD \approx I$. Využijeme-li neúplného LU rozkladu matice A je $A \approx LU$. Definujme $M^{-1} = LU$ (Volba $C = L^{-1}$ $D = U^{-1}$). V každém kroku metody spočteme vektor $z_k = Mr_k$ ze soustavy $M^{-1}z_k = r_k$. Zřejmě je-li $A \approx LU$, je $L^{-1}AU^{-1} \approx I$ a tedy $CAD \approx I$.

Poznámka : Při předpodmínění jsme metodu BCG aplikovali na systém (2.36) . Zřejmě tedy platí $\widehat{\widetilde{r}}_i^T\widehat{r}_j = 0$ pro $i \neq j$, neboli $\widetilde{r}_i^T Mr_j = 0$ pro $i \neq j$.

Ukažme si nyní, jaké podmínky splňují vektory r_{k+1} a $\widetilde{r}_{k+1}$ generované metodou BCG s předpodmíněním.

Lemma 2.12 : Nechť r_{k+1} a $\widetilde{r}_{k+1}$ jsou vektory spočtené algoritmem 2.4. Potom platí

$$r_{k+1} \perp sp\{M^T\widetilde{r}_0, M^TA^TM^T\widetilde{r}_0, .., (M^TA^T)^kM^T\widetilde{r}_0\}, \tag{2.37}$$
$$\widetilde{r}_{k+1} \perp sp\{Mr_0, M(MA)r_0, .., M(MA)^kr_0\}. \tag{2.38}$$

Důkaz : Protože platí $\widetilde{p}_j = M^T\widetilde{r}_j + \beta_{j-1}\widetilde{p}_{j-1}$ a $\widetilde{p}_{j-1}$ můžeme též rozepsat jako lineární kombinaci

$$\widetilde{p}_{j-1} = \sum_{i=0}^{j-1} \eta_i M^T\widetilde{r}_i,$$

je

$$sp\{\widetilde{p}_0, \dots, \widetilde{p}_k\} \subseteq sp\{M^T\widetilde{r}_0, \dots, M^T\widetilde{r}_k\}.$$

Nyní

$$\widetilde{r}_1 = \widetilde{r}_0 - \alpha_0A^T\widetilde{p}_0 = \widetilde{r}_0 - \alpha_0A^TM^T\widetilde{r}_0 = \mathcal{P}_1(A^TM^T)\widetilde{r}_0,$$

přičemž platí $\mathcal{P}_1(0) = 1$. Dále pokračujme indukcí. Platí

$$\widetilde{r}_{k+1} = \widetilde{r}_k - \alpha_kA^T\widetilde{p}_k = \mathcal{P}_k(A^TM^T)\widetilde{r}_0 - \alpha_kA^T(M^T\widetilde{r}_k + \beta_{k-1}\widetilde{p}_{k-1}) =$$

$$= \mathcal{P}_k(A^TM^T)\widetilde{r}_0 - \alpha_kA^TM^T\mathcal{P}_k(A^TM^T)\widetilde{r}_0 - \alpha_k\beta_{k-1}A^TM^T\sum_{i=0}^{k-1}\eta_i\widetilde{r}_i = \mathcal{P}_{k+1}(A^TM^T)\widetilde{r}_0$$

a opět je $\mathcal{P}_{k+1}(0) = 1$. Celkem tedy máme

$$sp\{M^T\widetilde{r}_0, \dots, M^T\widetilde{r}_k\} = sp\{M^T\widetilde{r}_0, M^T\mathcal{P}_1(A^TM^T)\widetilde{r}_0, \dots, M^T\mathcal{P}_k(A^TM^T)\widetilde{r}_0\} \subseteq$$

$$\subseteq sp\{M^T\widetilde{r}_0, M^T(A^TM^T)\widetilde{r}_0, \dots, M^T(A^TM^T)^k\widetilde{r}_0\}.$$

Dokážeme-li, že místo poslední inkluze platí rovnost prostorů, bude již tvrzení věty zřejmé, protože je $r_{k+1} \perp sp\{M^T\widetilde{r}_0, \dots, M^T\widetilde{r}_k\}$. Označme si

$$\mathcal{F}_k = sp\{\mathcal{P}_0(A^TM^T)\widetilde{r}_0, \dots, \mathcal{P}_k(A^TM^T)\widetilde{r}_0\}, \quad \mathcal{S}_k = sp\{\widetilde{r}_0, (A^TM^T)\widetilde{r}_0, \dots, (A^TM^T)^k\widetilde{r}_0\},$$

$$\mathcal{P}_i(A^TM^T) = \sum_{j=1}^{i+1} \gamma_j^i (A^TM^T)^{j-1}.$$

Z konstrukce polynomu $\mathcal{P}_i(A^TM^T)$ dále plyne, že koeficient u nejvyšší mocniny lze napsat ve tvaru

$$\gamma_{i+1}^i = \prod_{j=0}^{i-1} \alpha_j \neq 0.$$

Dále platí

$$x \in \mathcal{F}_k \Rightarrow x = \sum_{i=0}^{k} \bar{x}_i\mathcal{P}_i(A^TM^T)\widetilde{r}_0 = \sum_{i=0}^{k}\sum_{j=1}^{i+1} \bar{x}_i\gamma_j^i(A^TM^T)^{j-1}\widetilde{r}_0 =$$

$$= \sum_{j=1}^{k+1}\sum_{i=j-1}^{k} \bar{x}_i\gamma_j^i(A^TM^T)^{j-1}\widetilde{r}_0 \sum_{j=1}^{k+1}\left((A^TM^T)^{j-1}\widetilde{r}_0 \sum_{i=j-1}^{k} \bar{x}_i\gamma_j^i\right).$$

Označíme-li
$$\bar{y}_{j-1} = \sum_{i=j-1}^{k} \bar{x}_i \gamma_j^i,$$
dostaneme
$$\sum_{j=1}^{k+1} \left((A^T M^T)^{j-1} \tilde{r}_0 \sum_{i=j-1}^{k} \bar{x}_i \gamma_j^i \right) = \sum_{j=0}^{k} \bar{y}_j (A^T M^T)^j \tilde{r}_0.$$
Vztah mezi $\bar{x}_i$ a $\bar{y}_j$ můžeme vyjádřit maticově jako
$$\begin{pmatrix} \gamma_1^0 & \dots & \gamma_1^k \\ \vdots & \ddots & \vdots \\ 0 & \dots & \gamma_{k+1}^k \end{pmatrix} \begin{pmatrix} \bar{x}_0 \\ \vdots \\ \bar{x}_k \end{pmatrix} = \begin{pmatrix} \bar{y}_0 \\ \vdots \\ \bar{y}_k \end{pmatrix}.$$
Výše napsaná matice je horní trojúhelníková a na diagonále má prvky $\gamma_{i+1}^i \neq 0$ pro $i = 0, .., k$.
Z regularity této matice, která určuje vztah mezi koeficienty vektoru vyjádřeného v bázi prostoru $\mathcal{F}_k$ a $\mathcal{S}_k$, plyne rovnost obou prostorů. Protože je $r_{k+1} \perp M^T \mathcal{F}_k$ platí též
$$r_{k+1} \perp M^T \mathcal{S}_k = sp\{M^T \tilde{r}_0, M^T A^T M^T \tilde{r}_0, .., (M^T A^T)^k M^T \tilde{r}_0\}.$$
Stejně by se dokazovalo, že $\tilde{r}_{k+1} \perp sp\{M r_0, M(MA) r_0, .., M(MA)^k r_0\}$. ■

2.4 Jednoduše zhlazená metoda CGS (SSCGS)

Normy vektorů $\bar{r}_k$ neklesají obecně monotónně. Zhlazení metody CGS spočívá v tom, že kromě posloupností $\bar{r}_k$, $\bar{x}_k$ generujeme navíc posloupnosti r_k, x_k, přičemž zajistíme monotónní klesání norem vektorů r_k a navíc $\| r_k \| \leq \| \bar{r}_k \|$.

Definice 2.6 : Definujme rekurentně vektory

(2.39) $$r_{k+1} = \bar{r}_{k+1} + \lambda_k (r_k - \bar{r}_{k+1}), \quad r_0 = \bar{r}_0,$$
kde
$$\lambda_k = \arg\min_{\lambda \in \mathcal{R}} \| \bar{r}_{k+1} + \lambda (r_k - \bar{r}_{k+1}) \|.$$
Lemma 2.13 : Platí

(2.40) $$\lambda_k = -\frac{\bar{r}_{k+1}^T (r_k - \bar{r}_{k+1})}{\| r_k - \bar{r}_{k+1} \|^2}.$$

Důkaz : Předpokládejme r_{k+1} ve tvaru $r_{k+1} = \bar{r}_{k+1} + \lambda (r_k - \bar{r}_{k+1})$ kde $\lambda \in \mathbb{R}$. Potom

(2.41) $$\| r_{k+1} \|^2 = \| \bar{r}_{k+1} \|^2 + 2\lambda \bar{r}_{k+1}^T (r_k - \bar{r}_{k+1}) + \lambda^2 \| r_k - \bar{r}_{k+1} \|^2$$

je kvadratická funkce proměnné λ a nabývá minima pro (2.40) . ■

Lemma 2.14 : Pro vektory r_k a r_{k+1} platí

(2.42) $$\| r_{k+1} \| \leq \| r_k \|,$$
(2.43) $$\| r_{k+1} \| \leq \| \bar{r}_{k+1} \| .$$

Důkaz : Protože je

$$\| r_{k+1} \| = \| \bar{r}_{k+1} + \lambda_k (r_k - \bar{r}_{k+1}) \| \leq \| \bar{r}_{k+1} + 1 \cdot (r_k - \bar{r}_{k+1}) \| = \| r_k \|,$$

platí (2.42) . Vztah (2.43) plyne z toho, že

$$\| r_{k+1} \| = \| \bar{r}_{k+1} + \lambda_k (r_k - \bar{r}_{k+1}) \| \leq \| \bar{r}_{k+1} + 0 \cdot (r_k - \bar{r}_{k+1}) \| = \| \bar{r}_{k+1} \| . \qquad \blacksquare$$

Důsledek : Platí implikace $\bar{r}_k = 0 \Rightarrow r_k = 0$.

Definice 2.7 : Definujme $x_k = A^{-1}(b - r_k)$.

Nyní již můžeme zformulovat z algoritmu 2.5 algoritmus nový, ve kterém navíc generujeme vektory r_k a vektory x_k.

Algoritmus 2.6

Input $\bar{x}_0,\ A,\ b,\ \widetilde{r}_0,\ M$;
$\bar{r}_0 = b - A\bar{x}_0$;
$r_0 = \bar{r}_0;\ x_0 = \bar{x}_0$;
$\bar{p}_0 = u_0 = M\bar{r}_0$;
for $k = 0$ **to** $n - 1$ **do**
  if $r_k = 0$ **then** $x^* = x_k$ *and exit*
  else
    begin
      $v_k = A\bar{p}_k;\ \ \alpha_k = \frac{\widetilde{r}_0^T M \bar{r}_k}{\widetilde{r}_0^T M v_k}$;
      $q_k = u_k - \alpha_k M v_k$;
      $\bar{r}_{k+1} = \bar{r}_k - \alpha_k A(u_k + q_k)$;
      $\bar{x}_{k+1} = \bar{x}_k + \alpha_k (u_k + q_k)$;
      $\beta_k = \frac{\widetilde{r}_0^T M \bar{r}_{k+1}}{\widetilde{r}_0^T M \bar{r}_k}$;
      $u_{k+1} = M\bar{r}_{k+1} + \beta_k q_k$;
      $\bar{p}_{k+1} = u_{k+1} + \beta_k (q_k + \beta_k \bar{p}_k)$;
      $\lambda_k = -\frac{\bar{r}_{k+1}^T (r_k - \bar{r}_{k+1})}{\| r_k - \bar{r}_{k+1} \|^2}$;
      $x_{k+1} = \bar{x}_{k+1} + \lambda_k (x_k - \bar{x}_{k+1})$;
      $r_{k+1} = \bar{r}_{k+1} + \lambda_k (r_k - \bar{r}_{k+1})$;
    end;
$x^* = x_n$.

Definice 2.8 : Metodu definovanou algoritmem 2.6 nazveme předpodmíněnou jednoduše zhlazenou metodou CGS (SSCGS).

2.5 Dvojnásobně zhlazená metoda CGS (DSCGS)

Myšlenka je podobná jako u metody SSCGS, ve které jsme minimizovali $\| \bar{r}_{k+1} + \lambda w \|$, kde w byl vhodný vektor. Nyní budeme minimalizovat $\| \bar{r}_{k+1} + \lambda w + \mu z \|$, přičemž volíme $w = r_k - \bar{r}_{k+1}$ a $z = v_k$.

Definice 2.9 : Definujme rekurentně vektory r_k jako

$$r_{k+1} = \bar{r}_{k+1} + \lambda_k(r_k - \bar{r}_{k+1}) + \mu_k v_k, \quad r_0 = \bar{r}_0, \tag{2.44}$$

kde čísla λ_k a μ_k jsou definovány vztahem

$$(\lambda_k, \mu_k)^T = \arg \min_{(\mu,\lambda)^T \in \mathcal{R}^2} \| \bar{r}_{k+1} + \lambda_k(r_k - \bar{r}_{k+1}) + \mu_k v_k \|. \tag{2.45}$$

Lemma 2.15 : Platí

$$\begin{pmatrix} \lambda_k \\ \mu_k \end{pmatrix} = -(V_k^T V_k)^{-1} V_k^T \bar{r}_{k+1}, \tag{2.46}$$

kde

$$V_k = (r_k - \bar{r}_{k+1}, v_k).$$

Důkaz : Definujme funkci $g(\lambda, \mu)$

$$g(\lambda, \mu) = \| r_{k+1} \|^2 = \| \bar{r}_{k+1} \| + \lambda_k^2 \| r_k - \bar{r}_{k+1} \|^2 + \mu_k^2 \| v_k \|^2 +$$

$$+ 2\lambda_k \bar{r}_{k+1}^T (r_k - \bar{r}_{k+1}) + 2\mu_k \bar{r}_{k+1}^T v_k + 2\lambda_k \mu_k (r_k - \bar{r}_{k+1})^T v_k.$$

$g(\lambda, \mu)$ je kvadratická konvexní funkce proměnných λ a μ. Extrém tedy spočteme z rovnic

$$\frac{\partial g}{\partial \lambda} = 0 \quad \text{a} \quad \frac{\partial g}{\partial \mu} = 0.$$

Platí

$$\begin{aligned}
\frac{\partial g}{\partial \lambda} &= 2\lambda \| r_k - \bar{r}_{k+1} \|^2 + 2\bar{r}_{k+1}^T (r_k - \bar{r}_{k+1}) + 2\mu (r_k - \bar{r}_{k+1})^T v_k, \\
\frac{\partial g}{\partial \mu} &= 2\mu \| v_k \|^2 + 2\bar{r}_{k+1}^T v_k + 2\lambda (r_k - \bar{r}_{k+1})^T v_k.
\end{aligned}$$

Vektor $(\lambda_k, \mu_k)^T$ realizující minimum spočteme ze soustavy

$$\begin{aligned}
-\bar{r}_{k+1}^T (r_k - \bar{r}_{k+1}) &= \lambda_k \| r_k - \bar{r}_{k+1} \|^2 + \mu_k (r_k - \bar{r}_{k+1})^T v_k, \\
-\bar{r}_{k+1}^T v_k &= \lambda_k (r_k - \bar{r}_{k+1})^T v_k + \mu_k \| v_k \|^2,
\end{aligned}$$

což je soustava (2.46) . ∎

Definice 2.10 : Definujme $x_k = A^{-1}(b - r_k)$.

Lemma 2.16 : Platí

$$x_{k+1} = \bar{x}_{k+1} + \lambda_k (x_k - \bar{x}_{k+1}) - \mu_k \bar{p}_k.$$

Důkaz : Rozepsáním (2.44) dostáváme

$$\begin{aligned}
b - Ax_{k+1} &= b - A\bar{x}_{k+1} + \lambda_k (b - Ax_k - b + A\bar{x}_{k+1}) + \mu_k A\bar{p}_k, \\
-Ax_{k+1} &= -A\bar{x}_{k+1} + \lambda_k (-Ax_k + A\bar{x}_{k+1}) + \mu_k A\bar{p}_k.
\end{aligned}$$

∎

Algoritmus 2.7

Input $\bar{x}_0,\ A,\ b,\ \widetilde{r}_0,\ M;$
$\bar{r}_0 = b - A\bar{x}_0;$
$r_0 = \bar{r}_0;\ x_0 = \bar{x}_0;$
$\bar{p}_0 = u_0 = M\bar{r}_0;$
for $k = 0$ **to** $n-1$ **do**
  if $r_k = 0$ **then** $x^* = x_k$ *and exit*
  else
    begin
      $v_k = A\bar{p}_k;\ \ \alpha_k = \frac{\widetilde{r}_0^T M\bar{r}_k}{\widetilde{r}_0^T M v_k};$
      $q_k = u_k - \alpha_k M v_k;$
      $\bar{r}_{k+1} = \bar{r}_k - \alpha_k A(u_k + q_k);$
      $\bar{x}_{k+1} = \bar{x}_k + \alpha_k(u_k + q_k);$
      $\beta_k = \frac{\widetilde{r}_0^T M\bar{r}_{k+1}}{\widetilde{r}_0^T M\bar{r}_k};$
      $u_{k+1} = M\bar{r}_{k+1} + \beta_k q_k;$
      $\bar{p}_{k+1} = u_{k+1} + \beta_k(q_k + \beta_k\bar{p}_k);$
      $\begin{pmatrix}\lambda_k\\ \mu_k\end{pmatrix} = -(V_k^T V_k)^{-1} V_k^T \bar{r}_{k+1};$
      $x_{k+1} = \bar{x}_{k+1} + \lambda_k(x_k - \bar{x}_{k+1}) - \mu_k\bar{p}_k;$
      $r_{k+1} = \bar{r}_{k+1} + \lambda_k(r_k - \bar{r}_{k+1}) + \mu_k v_k;$
    end;
$x^* = x_n.$

Definice 2.11 : Metodu definovanou algoritmem 2.7 nazveme předpodmíněnou dvojnásobně zhlazenou metodou CGS (DSCGS).

Kapitola 3

Zhroucení metody CGS

Počítáme-li soustavu lineárních rovnic metodou BCG či CGS, může nastat případ, kdy v algoritmu dojde při výpočtu dělení nulou. To může být zapříčiněno tím, že neexistuje vektor r_k, který by splňoval podmínky 1. a 2. v definici LOP. Když však zvýšíme dimenzi Krylovova prostoru, je možné, že nalezneme vektor r_{k+1}, který splňuje podmínky 1. a 2. v definici LOP. Principem, jak se vyhnou zhroucení metod, je nalézt algoritmus, který by nám umožnoval sestrojovat pouze existující rezidua. Přísnějším požadavkem, který v dalším budeme uplatňovat, je sestrojovat taková rezidua r_k jež existují a jsou určena jednoznačně, neboli determinant matice (2.1) je různý od nuly.

3.1 Ortogonální polynomy

Ukážeme si, že problém nalezení rezidua r_k lze převést na problém nalezení polynomu ortogonálního ke všem polynomům stupně menšího než k vzhledem k vhodně zvolenému funkcionálu.

Lemma 3.1 : *Nechť $d_k \neq 0$. Potom*

$$r_k = \mathcal{P}_k(A)r_0, \quad \textit{kde} \quad \mathcal{P}_k(A) = I + p_1^{(k)}A + \ldots + p_k^{(k)}A^k. \tag{3.1}$$

Koeficienty $p_1^{(k)}, \ldots, p_k^{(k)}$ dostaneme jako řešení soustavy (2.10).

Důkaz : Plyne z lemmatu 2.3. ∎

Definice 3.1 : Označme $\pi(\xi^i) = (A^i r_0, \widetilde{r}_0)$ lineární funkcionál na prostoru polynomů.

Lemma 3.2 : Platí

$$(r_k, (A^T)^i \widetilde{r}_0) = \pi(\xi^i \mathcal{P}_k(\xi)) \quad \text{pro} \quad i = 0, \ldots, k-1. \tag{3.2}$$

Důkaz : Vztah (3.2) dokážeme tak, že rozepíšeme polynom $\mathcal{P}_k(\xi)$ a využijeme definice funkcionálu π. Platí

$$\pi(\xi^i \mathcal{P}_k(\xi)) = \pi(\xi^i + \sum_{j=1}^{k} p_j^{(k)} \xi^{i+j}) = \pi(\xi^i) + \sum_{j=1}^{k} p_j^{(k)} \pi(\xi^{i+j}) = (A^i r_0, \widetilde{r}_0) +$$

$$+ \sum_{j=1}^{k} p_j^{(k)} (A^{i+j} r_0, \widetilde{r}_0) = (A^i (I + \sum_{j=1}^{k} p_j^{(k)} A^j) r_0, \widetilde{r}_0) = (A^i r_k, \widetilde{r}_0) = (r_k, (A^T)^i \widetilde{r}_0). \quad \blacksquare$$

Důsledek : Protože je $(r_k, (A^T)^i\tilde{r}_0) = 0$ pro $i = 0, \ldots, k-1$ platí

$$\pi(\xi^i \mathcal{P}_k(\xi)) = 0 \quad \text{pro} \quad i = 0, \ldots, k-1. \tag{3.3}$$

Lemma 3.3 : *Nechť $d_k \neq 0$. Potom je $\mathcal{P}_k$ určen podmínkami (3.3) jednoznačně až na multiplikativní konstantu. Dodefinujeme-li tedy pro polynom $\mathcal{P}_k$ navíc podmínku $\mathcal{P}_k(0) = 1$, je $\mathcal{P}_k$ určen podmínkami (3.3) jednoznačně.*

Důkaz : Plyne z předchozího lemmatu a z lemmatu 2.3. ∎

Shrnutí : Problém nalezení rezidua r_k jsme převedli na problém nalezení polynomu $\mathcal{P}_k$ ortogonálního ke všem polynomům stupně menšího než k vzhledem k funkcionálu π a splňujícího podmínku $\mathcal{P}_k(0) = 1$.

Poznámka : Koeficienty objevující se v různých rekurentních vztazích pro počítání ortogonálních polynomů $\mathcal{P}_k$ jsou dány jako podíly skalárních součinů. Pokud je skalární součin ve jmenovateli nulový, dochází ke *zhroucení* (*breakdown*) a následovnému zastavení algoritmu. To může být způsobeno neexistencí ortogonálního polynomu (*Skutečné zhroucení - true breakdown*) nebo užitím rekurentních vztahů (*Lanczosovo zhroucení - Lanczos breakdown*). Pokud je skalární součin ve jmenovateli blízký nule, potom mohou mít zaokrouhlovací chyby vážný vliv na algoritmus. Tuto situaci budeme označovat jako *skoro-zhroucení* (*near-breakdown*). Podobně jako v předchozím definujeme *skutečné skoro-zhroucení* a *Lanczosovo skoro-zhroucení*. V této kapitole se budeme zabývat tím, jak se vyhnou zhroucení u metody CGS.

Příklad : Demonstrujme pojmy uvedené v předchozí poznámce na metodě BCG. Ke zhrouceni dochází tehdy, dojde-li k dělení nulou v některém z rekurentních vztahů, což může nastat jedině při výpočtu koeficientů α_k nebo β_k. Skutečné zhroucení nastává tehdy, není-li možné sestrojit vektor r_{k+1} (a tím ani vektor x_{k+1}), dochází tedy zřejmě k dělení nulou při výpočtu koeficientu α_k. Je-li $\tilde{r}_k^T r_k = 0$, dochází ke zhroucení BCG při výpočtu koeficientu β_k, nedochází však nutně ke zhroucení LOP, protože H_{k+1} může být regulární. Zhroucení BCG při výpočtu β_k je tedy příkladem Lanczosova zhroucení.

Definice 3.2 : Označme $\mu(\xi^i) = \pi(\xi^{i+1})$ lineární funkcionál na prostoru polynomů.

Definice 3.3 : Definujme monický polynom $\mathcal{M}_k(\xi)$ stupně k podmínkou

$$\mu(\xi^i \mathcal{M}_k(\xi)) = 0 \qquad \text{pro} \quad i = 0, \ldots, k-1. \tag{3.4}$$

Poznámka : Polynom $\mathcal{M}_k(\xi)$ budeme uvažovat ve tvaru

$$\mathcal{M}_k(\xi) = \sum_{j=0}^{k-1} m_j^{(k)} \xi^j + \xi^k. \tag{3.5}$$

Tento polynom se ukáže užitečným při zjišťování singularity matice H_k.

Lemma 3.4 : *Je-li $d_k \neq 0$ potom polynom $\mathcal{M}_k(\xi)$ existuje a je určen jednoznačně.*

Důkaz : Přepsáním ortogonálních podmínek (3.4) dostáváme pro $i = 0, \ldots, k-1$

$$\begin{aligned} 0 &= \mu(\sum_{j=0}^{k-1} m_j^{(k)} \xi^{j+i} + \xi^{k+i}) = \sum_{j=0}^{k-1} m_j^{(k)} \mu(\xi^{j+i}) + \mu(\xi^{k+i}) = \\ &= \sum_{j=0}^{k-1} m_j^{(k)} (A^{j+i+1} r_0, \tilde{r}_0) + (A^{k+i+1} r_0, \tilde{r}_0). \end{aligned}$$

Zapsáno maticově

$$(3.6)\quad -\begin{pmatrix} (\tilde{r}_0, A^{k+1} r_0) \\ (\tilde{r}_0, A^{k+2} r_0) \\ \vdots \\ (\tilde{r}_0, A^{2k} r_0) \end{pmatrix} = \begin{pmatrix} (\tilde{r}_0, A r_0) & (\tilde{r}_0, A^2 r_0) & \ldots & (\tilde{r}_0, A^k r_0) \\ (\tilde{r}_0, A^2 r_0) & (\tilde{r}_0, A^3 r_0) & \ldots & (\tilde{r}_0, A^{k+1} r_0) \\ \vdots & \vdots & & \vdots \\ (\tilde{r}_0, A^k r_0) & (\tilde{r}_0, A^{k+1} r_0) & \ldots & (\tilde{r}_0, A^{2k-1} r_0) \end{pmatrix} \begin{pmatrix} m_0^{(k)} \\ m_1^{(k)} \\ \vdots \\ m_{k-1}^{(k)} \end{pmatrix}.$$

Dostali jsme opět soustavu s maticí H_k. Je-li $d_k \neq 0$, je H_k regulární a polynom $\mathcal{M}_k(\xi)$ existuje a je určen jednoznačně. ■

3.2 Zjišťování singularity matice H_k

Definice 3.4 : Symbolem n_k budeme označovat index, pro nějž platí $n_k \geq k$. Pro zjednodušení zápisu definujme číslo $h_i = (\tilde{r}_0, A^i r_0)$, i-tý sloupec matice H_{n_k} budeme značit $H_{n_k}^{(i)}$.

Poznámka : Matici H_{n_k} a vektor $H_{n_k}^{(i)}$ můžeme psát tedy ve tvaru

$$H_{n_k} = \begin{pmatrix} h_1 & h_2 & \ldots & h_{n_k} \\ h_2 & h_3 & \ldots & h_{n_k+1} \\ \vdots & \vdots & & \vdots \\ h_{n_k} & h_{n_k+1} & \ldots & h_{2n_k-1} \end{pmatrix} \quad \text{a} \quad H_{n_k}^{(i)} = \begin{pmatrix} h_i \\ h_{i+1} \\ \vdots \\ h_{n_k+i-1} \end{pmatrix}.$$

Protože je H_{n_k} symetrická, je $(H_{n_k}^{(i)})^T$ i-tý řádek matice H_{n_k}.

Lemma 3.5 : Nechť H_{n_k} je regulární matice a nechť $\mu(\xi^i \mathcal{M}_{n_k}(\xi)) = 0$ pro hodnoty $i = 0 \ldots n_k + \delta_k - 2$, kde $2 \leq \delta_k$. Potom jsou matice $H_{n_k+1}, H_{n_k+2} \ldots H_{n_k+\delta_k-1}$ singulární.

Důkaz : Rozepsáním podmínek $\mu(\mathcal{M}_{n_k} \xi^i) = 0$ pro $i = 0 \ldots n_k + j - 2$ kde $2 \leq j \leq \delta_k$ dostáváme

$$(3.7)\quad -\begin{pmatrix} h_{n_k+1} \\ h_{n_k+2} \\ \vdots \\ h_{2n_k} \\ h_{2n_k+1} \\ \vdots \\ h_{2n_k+j-1} \end{pmatrix} = m_0^{(n_k)} \begin{pmatrix} h_1 \\ h_2 \\ \vdots \\ h_{n_k} \\ h_{n_k+1} \\ \vdots \\ h_{n_k+j-1} \end{pmatrix} + m_1^{(n_k)} \begin{pmatrix} h_2 \\ h_3 \\ \vdots \\ h_{n_k+1} \\ h_{n_k+2} \\ \vdots \\ h_{n_k+j} \end{pmatrix} + \ldots + m_{n_k-1}^{(n_k)} \begin{pmatrix} h_{n_k} \\ h_{n_k+1} \\ \vdots \\ h_{2n_k-1} \\ h_{2n_k} \\ \vdots \\ h_{2n_k+j-2} \end{pmatrix}.$$

Vektor na levé straně přenásobený číslem -1 je (n_k+1)-ním sloupkem matice H_{n_k+j-1}, vektory na pravé straně jsou postupně první až n_k-tý sloupec matice H_{n_k+j-1}. Platí tedy

$$H^{(n_k+1)}_{n_k+j-1} = -\sum_{i=1}^{n_k} m^{(n_k)}_{i-1} H^{(i)}_{n_k+j-1}$$

a H_{n_k+j-1} je singulární pro všechna $j = 2, \dots, \delta_k$. ■

Lemma 3.6 : Uvažujme H_{n_k} regulární. Nechť matice $H_{n_k+1}, \dots, H_{n_k+\delta_k-1}$ jsou singulární ($\delta_k \geq 2$). Neexistuje-li řešení soustavy

$$(3.8) \qquad -\begin{pmatrix} h_{n_k+\delta_k} \\ h_{n_k+\delta_k+1} \\ \vdots \\ h_{2n_k+\delta_k-1} \\ h_{2n_k+\delta_k} \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & \dots & h_{n_k+\delta_k-1} \\ h_2 & h_3 & \dots & h_{n_k+\delta_k} \\ \vdots & \vdots & & \vdots \\ h_{n_k} & h_{n_k+1} & \dots & h_{2n_k+\delta_k-2} \\ h_{n_k+1} & h_{n_k+2} & \dots & h_{2n_k+\delta_k-1} \end{pmatrix} \begin{pmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_{n_k+\delta_k-1} \end{pmatrix},$$

potom matice $H_{n_k+\delta_k}$ je regulární.

Důkaz : Protože H_{n_k} je regulární, je prvních n_k-řádků matice $H_{n_k+\delta_k}$ lineárně nezávislých. Indukcí dokážeme, že všechny řádky matice $H_{n_k+\delta_k}$ jsou lineárně nezávislé. Nechť tedy n_k+j řádků je lineárně nezávislých, ale n_k+j+1 řádků je již lineárně závislých ($0 \leq j \leq \delta_k - 1$). Tedy existují koeficienty c_i pro $i = 1, \dots, n_k+j$ takové, že

$$\sum_{l=1}^{n_k+j} c_l^2 \neq 0 \quad \text{a} \quad (H^{(n_k+j+1)}_{n_k+\delta_k})^T = \sum_{i=1}^{n_k+j} c_i (H^{(i)}_{n_k+\delta_k})^T.$$

Tedy (n_k+j+1)-ní řádek matice $H_{n_k+\delta_k}$ je lineární kombinací prvních n_k+j řádků s koeficienty c_i. V (n_k+j+1)-ním řádku se nacházejí h_i s indexy i od n_k+j+1 do $2n_k+\delta_k+j$. Protože je $0 \leq j \leq \delta_k - 1$, dostáváme nerovnost

$$n_k + j + 1 \leq n_k + \delta_k < 2n_k + \delta_k \leq 2n_k + \delta_k + j,$$

tedy v (n_k+j+1)-ním řádku nalezneme celý vektor levé strany (přenásobený -1) soustavy (3.8) . Znázorněme si tuto situaci na prvních (n_k+j+1)-řádcích matice $H_{n_k+\delta_k}$

$$(3.9) \qquad \left(\begin{array}{cc|ccc|cc} h_1 & \dots & h_{\delta_k-j} & \dots & h_{n_k+\delta_k-j} & \dots & h_{n_k+\delta_k} \\ \vdots & & \vdots & & \vdots & & \vdots \\ h_{n_k+j} & \dots & h_{n_k+\delta_k-1} & \dots & h_{2n_k+\delta_k-1} & \dots & h_{2n_k+\delta_k+j-1} \\ \hline h_{n_k+j+1} & \dots & h_{n_k+\delta_k} & \dots & h_{2n_k+\delta_k} & \dots & h_{2n_k+\delta_k+j} \end{array}\right).$$

Vektor levé strany (3.8) je tedy zároveň též lineární kombinací příslušných vektorů v prvních n_k+j řádcích matice $H_{n_k+\delta_k}$ s koeficienty $-c_1, \dots, -c_{n_k+j}$. Vektor $-(0, \dots, 0, c_1, \dots, c_{n_k+j})^T$ o $n_k+\delta_k-1$ složkách je tedy řešením soustavy (3.8) což je spor. ■

Důsledek : Ze singularity matice $H_{n_k+\delta_k}$ plyne tedy existence řešení soustavy (3.8) .

Lemma 3.7 : Je-li H_{n_k} regulární a jsou-li $H_{n_k+1}, \dots, H_{n_k+\delta_k-1}$ singulární, potom je $\mu(\xi^i \mathcal{M}_{n_k}) = 0$ pro $i = 0, \dots, n_k+\delta_k-2$ a $\delta_k \geq 2$.

Důkaz : Nejdříve indukcí dokážeme, že n_k+1 řádek matice H_{n_k+i} pro $i = 1, \dots, \delta_k - 1$ je netriviální lineární kombinací prvních n_k řádků.

- Nechť $i = 1$.
  Protože H_{n_k} je regulární, prvních n_k-řádků H_{n_k+1} je lineárně nezávislých, nutně je tedy poslední řádek netriviální lineární kombinací prvních n_k-řádků. Zároveň je vidět, že koeficienty této lineární kombinace jsou jednoznačně určeny jako řešení soustavy (3.6), tedy
  $$(H_{n_k+1}^{(n_k+1)})^T = -\sum_{j=1}^{n_k} m_{j-1}^{(n_k)} (H_{n_k+1}^{(j)})^T \Rightarrow -H_{n_k+1}^{(n_k+1)} = \sum_{j=1}^{n_k} m_{j-1}^{(n_k)} H_{n_k+1}^{(j)}.$$
- Nechť $1 \le i \le \delta_k - 2$ libovolné
  Nechť tedy (n_k+1)-ní řádek matice H_{n_k+i} je lineární kombinací prvních n_k řádků a nechť H_{n_k+i+1} je singulární. Z předchozího lemmatu plyne, že soustava
  $$(3.10) \qquad -\begin{pmatrix} h_{n_k+i+1} \\ \vdots \\ h_{2n_k+i+1} \end{pmatrix} = \begin{pmatrix} h_1 & \dots & h_{n_k+i} \\ \vdots & & \vdots \\ h_{n_k+1} & \dots & h_{2n_k+i} \end{pmatrix} \begin{pmatrix} \rho_1 \\ \vdots \\ \rho_{n_k+i} \end{pmatrix}$$
  má řešení. Z toho, že hodnost matice soustavy (3.10) je n_k a z toho, že soustava má řešení plyne, že matice rozšířená o (-1) násobek vektoru levé strany má hodnost n_k, což znamená, že (n_k+1)-ní řádek matice H_{n_k+i+1} je lineární kombinací prvních n_k řádků.

Pro $i = \delta_k - 1$ máme, že poslední řádek matice

$$(3.11) \qquad \left(\begin{array}{|cccc|ccc} \hline h_1 & h_2 & \dots & h_{n_k} & h_{n_k+1} & \dots & h_{n_k+\delta_k-1} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ h_{n_k} & h_{n_k+1} & \dots & h_{2n_k-1} & h_{2n_k} & \dots & h_{2n_k+\delta_k-2} \\ \hline h_{n_k+1} & h_{n_k+2} & \dots & h_{2n_k} & h_{2n_k+1} & \dots & h_{2n_k+\delta_k-1} \\ \hline \end{array}\right)$$

je lineární kombinací prvních n_k řádků. Koeficienty této lineární kombinace jsou jednoznačně určeny z prvních n_k sloupců jako řešení soustavy (3.6). Platí tedy (3.7) pro $j = \delta_k$ neboli $\mu(\mathcal{M}_{n_k}\xi^j) = 0$ pro $j = 0, \dots, n_k + \delta_k - 2$. ■

Věta 3.1 : Platí

$$\mu(\xi^i \mathcal{M}_{n_k}) = 0 \quad \text{pro} \quad i = 0, \dots, n_k + \delta_k - 2 \Leftrightarrow H_{n_k+1}, \dots, H_{n_k+\delta_k-1} \quad \text{jsou singulární.}$$

Důkaz : Plyne z předchozích lemmat. ■

3.3 Regulární polynomy

Definice 3.5 : Polynomy $\mathcal{M}_{n_k}$ a $\mathcal{P}_{n_k}$ nazveme regulárními polynomy, je-li $d_{n_k} \neq 0$.

Poznámka : Chceme-li se tedy vyhnout skutečnému zhroucení, stačí když budeme sestrojovat pouze regulární polynomy a ostatní budeme přeskakovat. Chceme-li se navíc vyhnout Lanczosovu zhroucení v nějaké interpretaci LOP, budeme z regulárních polynomů sestrojovat pouze ty, při kterých nedochází k Lanczosovu zhroucení.

Označení : Pišme v dalším $\mathcal{P}_k$ a $\mathcal{M}_k$ místo $\mathcal{P}_{n_k}$ a $\mathcal{M}_{n_k}$. Vyjímku tvoří lemma 3.11, kde jsme ponechali původní označení.

Definice 3.6 : Uvažujme polynomy $\mathcal{P}_k$ a $\mathcal{M}_k$ ve tvaru

$$\mathcal{P}_k = 1 + \sum_{j=1}^{n_k} p_j^{(k)} \xi^j \quad \text{a} \quad \mathcal{M}_k = \sum_{j=0}^{n_k-1} m_j^{(k)} \xi^j + \xi^{n_k}. \tag{3.12}$$

Lemma 3.8 : *Podmínky $\pi(\xi^{n_k+i}\mathcal{P}_{k+1}) = 0$ pro $i = 0, \ldots, \delta_k - 1$ je možné psát ve tvaru $\pi(\xi^i \mathcal{M}_k \mathcal{P}_{k+1}) = 0$ pro $i = 0, \ldots, \delta_k - 1$.*

Důkaz : Připomeňme že platí

$$\pi(\xi^i \mathcal{P}_{k+1}) = 0 \quad \text{pro} \quad i = 0, \ldots, n_k - 1.$$

- Nechť platí $\pi(\xi^i \mathcal{P}_{k+1}) = 0$ pro $i = n_k, \ldots, n_k + \delta_k - 1$. Polynom $\mathcal{P}_{k+1}$ je kolmý na všechny polynomy stupně nejvýše $n_k + \delta_k - 1$. Protože $\xi^i \mathcal{M}_k$ je pro $i = 0, \ldots, \delta_k - 1$ polynom stupně nejvýše $n_k + \delta_k - 1$, je $\pi(\xi^i \mathcal{M}_k \mathcal{P}_{k+1}) = 0$ pro $i = 0, \ldots, \delta_k - 1$.
- Nechť platí $\pi(\xi^i \mathcal{M}_k \mathcal{P}_{k+1}) = 0$ pro $i = 0, \ldots, \delta_k - 1$. Indukcí dokážeme, že platí $\pi(\xi^{n_k+i} \mathcal{P}_{k+1}) = 0$ pro $i = 0, \ldots, \delta_k - 1$.

  1. Pro $i = 0$ máme

  $$0 = \pi(\mathcal{M}_k \mathcal{P}_{k+1}) = \pi((\sum_{j=0}^{n_k-1} m_j^{(n_k)} \xi^j + \xi^{n_k}) \mathcal{P}_{k+1}) = \pi(\xi^{n_k} \mathcal{P}_{k+1}).$$

  2. Nechť nyní platí $\pi(\xi^{n_k+j} \mathcal{P}_{k+1}) = 0$ pro $j = 0, \ldots, i - 1$ $(0 < i < \delta_k)$. Potom

  $$0 = \pi(\xi^i \mathcal{M}_k \mathcal{P}_{k+1}) = \sum_{j=0}^{n_k-1} m_j^{(n_k)} \pi(\xi^{i+j} \mathcal{P}_{k+1}) + \pi(\xi^{n_k+i} \mathcal{P}_{k+1}) = \pi(\xi^{n_k+i} \mathcal{P}_{k+1}),$$

  tedy $\pi(\xi^{n_k+i} \mathcal{P}_{k+1}) = 0$. ■

Lemma 3.9 : *Podmínky $\mu(\xi^{n_k+i} \mathcal{M}_{k+1}) = 0$ pro $i = 0, \ldots, \delta_k - 1$ je možné psát ve tvaru $\mu(\xi^i \mathcal{M}_k \mathcal{M}_{k+1}) = 0$ pro $i = 0, \ldots, \delta_k - 1$.*

Důkaz : Analogicky dle předchozího lemmatu. ■

Definice 3.7 : Definujme čísla π_i a μ_i vztahy

$$\pi_i = \pi(\xi^i \mathcal{P}_k \mathcal{M}_k) \quad \text{a} \quad \mu_i = \mu(\xi^i \mathcal{M}_k^2).$$

Lemma 3.10 : *Nechť je splněno $\mu_0 = 0, \ldots, \mu_{\delta_k-2} = 0$, $\mu_{\delta_k-1} \neq 0$ a nechť existuje t_k takové $(0 \leq t_k \leq \delta_k - 1)$, že $\pi_i = 0$ pro $i < t_k$ a $\pi_{t_k} \neq 0$. Potom lze polynomy $\mathcal{P}_{k+1}$ a $\mathcal{M}_{k+1}$ počítat podle vztahů*

$$\mathcal{P}_{k+1} = \mathcal{P}_k - \xi \mathcal{W}_k \mathcal{M}_k, \tag{3.13}$$

$$\mathcal{M}_{k+1} = \mathcal{Q}_k \mathcal{M}_k + C_k \xi^{t_k} \mathcal{P}_k, \tag{3.14}$$

kde $\mathcal{W}_k$ je polynom stupně nejvýše $\delta_k - 1$, $\mathcal{Q}_k$ je monický polynom stupně δ_k a C_k je konstanta.

Důkaz : Uvažujme polynom $\mathcal{Z}_{k+1}$ ve tvaru

$$\mathcal{Z}_{k+1} = \mathcal{P}_k - \xi \mathcal{W}_k \mathcal{M}_k. \tag{3.15}$$

Polynom $\mathcal{Z}_{k+1}$ je stupně nejvýše $n_k + \delta_k$ a platí pro něj $\mathcal{Z}_{k+1}(0) = 1$. Určíme-li navíc koeficienty polynomu $\mathcal{W}_k$ tak, aby byl $\mathcal{Z}_{k+1}$ kolmý vzhledem k funkcionálu π na všechny polynomy stupně menšího než $n_k+\delta_k$, bude nutně platit $\mathcal{Z}_{k+1} = \mathcal{P}_{k+1}$ (z jednoznačnosti určení polynomu $\mathcal{P}_{k+1}$). Uvažujme polynom $\mathcal{W}_k$ ve tvaru

$$\mathcal{W}_k(\xi) = \sum_{j=0}^{\delta_k - 1} w_j \xi^j.$$

Přenásobíme-li (3.15) ξ^i $(i = 0, \dots, n_k + \delta_k - 1)$ a aplikujeme-li na rovnici funkcionál π, dostáváme

$$0 = \pi(\xi^i \mathcal{P}_k) - \sum_{j=0}^{\delta_k - 1} w_j \mu(\xi^{i+j} \mathcal{M}_k). \tag{3.16}$$

Definujme čísla $\widetilde{\pi}_i = \pi(\xi^{n_k+i}\mathcal{P}_k)$ a $\widetilde{\mu}_i = \mu(\xi^{n_k+i}\mathcal{M}_k)$. Ukážeme si, že $\widetilde{\pi}_i = \pi_i$ pro $i = 0, \dots, t_k$ a $\widetilde{\mu}_i = \mu_i$ pro $i = 0, \dots, \delta_k - 1$.

- Pro $i = 0$ máme $\widetilde{\mu}_0 = \mu(\xi^{n_k}\mathcal{M}_k) = \mu(\mathcal{M}_k^2) = \mu_0$.
- Nechť platí $\widetilde{\mu}_i = \mu_i$ pro $i = 0, \dots, j$ $(j < \delta_k - 1)$. Ukážeme, že tvrzení platí i pro $i = j + 1$.

$$\mu_{j+1} = \mu(\xi^{j+1}(\xi^{n_k} + \sum_{l=0}^{n_k-1} m_l^{(n_k)} \xi^l)\mathcal{M}_k) = \widetilde{\mu}_{j+1} + \sum_{l=0}^{n_k-1} m_l^{(n_k)} \mu(\xi^{l+j+1}\mathcal{M}_k) = \widetilde{\mu}_{j+1},$$

přičemž při poslední rovnosti jsme použili toho, že polynom $\mathcal{M}_k$ je kolmý na všechny polynomy stupně menšího než n_k, indukčního předpokladu a platnosti předpokladu $\mu_0 = 0, \dots, \mu_{\delta_k - 2} = 0$. Podobně by se ukázalo, že $\widetilde{\pi}_i = \pi_i$ pro $i = 0, \dots, t_k$.

Rovnost (3.16) je tedy zřejmě splněna triviálně pro $i = 0, \dots, n_k - 1$. Pro hodnoty $i = n_k, \dots, n_k + \delta_k - 1$ lze (3.16) přepsat ve tvaru

$$0 = \widetilde{\pi}_l - \sum_{j=0}^{\delta_k - 1} w_j \widetilde{\mu}_{j+l} \quad \text{pro} \quad l = 0, \dots, \delta_k - 1. \tag{3.17}$$

Označíme-li $x = \delta_k - 1$ a využijeme-li toho, že $\widetilde{\mu}_x = \mu_x$, můžeme (3.17) psát ve tvaru

$$\begin{pmatrix} \widetilde{\pi}_x \\ \vdots \\ \vdots \\ \widetilde{\pi}_0 \end{pmatrix} = \begin{pmatrix} \mu_x & \widetilde{\mu}_{x+1} & \dots & \widetilde{\mu}_{2x} \\ 0 & \mu_x & \dots & \widetilde{\mu}_{2x-1} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \mu_x \end{pmatrix} \begin{pmatrix} w_0 \\ \vdots \\ \vdots \\ w_x \end{pmatrix}. \tag{3.18}$$

Protože je matice soustavy (3.18) regulární, existuje a je určen jednoznačně polynom $\mathcal{W}_k$ takový, že polynom $\mathcal{Z}_{k+1}$ je kolmý na všechny polynomy stupně menšího než $n_k+\delta_k$ a tudíž platí (3.13) .

Podobně budeme dokazovat druhý vztah. Uvažujme polynom $\mathcal{X}_{k+1}$ ve tvaru

$$\mathcal{X}_{k+1} = \mathcal{Q}_k \mathcal{M}_k + C_k \xi^{t_k} \mathcal{P}_k \tag{3.19}$$

a polynom $\mathcal{Q}_k$ ve tvaru

$$\mathcal{Q}_k(\xi) = \sum_{j=0}^{\delta_k-1} q_j \xi^j + \xi^{\delta_k}.$$

Polynom $\mathcal{X}_{k+1}$ je monický polynom stupně $n_k + \delta_k$. Určíme-li navíc koeficienty polynomu $\mathcal{Q}_k$ a konstantu C_k tak, aby byl $\mathcal{X}_{k+1}$ kolmý vzhledem k funkcionálu μ na všechny polynomy stupně menšího než $n_k + \delta_k$, bude nutně platit $\mathcal{X}_{k+1} = \mathcal{M}_{k+1}$ (z jednoznačnosti určení polynomu $\mathcal{M}_{k+1}$). Přenásobíme-li (3.19) ξ^i $(i = 0, \ldots, n_k + \delta_k - 1)$ a aplikujeme-li na rovnici funkcionál μ, dostáváme

$$0 = \mu(\xi^{i+\delta_k}\mathcal{M}_k) + \sum_{j=0}^{\delta_k-1} q_j \mu(\xi^{i+j}\mathcal{M}_k) + C_k \pi(\xi^{i+t_k+1}\mathcal{P}_k). \tag{3.20}$$

Rovnost (3.20) je splněna triviálně pro $i = 0, \ldots, n_k - 2$. Pro $i = n_k - 1$ dostáváme rovnost

$$-\mu_{\delta_k-1} = C_k \pi_{t_k}. \tag{3.21}$$

Protože je $\pi_{t_k} \neq 0$, můžeme z (3.21) spočíst C_k. Pro $i = n_k, \ldots, n_k + \delta_k - 1$ můžeme (3.20) psát jako

$$-\widetilde{\mu}_{l+\delta_k} = \sum_{j=0}^{\delta_k-1} q_j \widetilde{\mu}_{l+j} + C_k \widetilde{\pi}_{l+t_k+1} \tag{3.22}$$

pro $l = 0, \ldots, \delta_k - 1$. Označíme-li $x = \delta_k - 1$, můžeme (3.22) zapsat maticově jako

$$- \begin{pmatrix} \widetilde{\mu}_{2x+1} + C_k \widetilde{\pi}_{t_k+\delta_k} \\ \vdots \\ \widetilde{\mu}_{x+2} + C_k \widetilde{\pi}_{t_k+2} \\ \widetilde{\mu}_{x+1} + C_k \widetilde{\pi}_{t_k+1} \end{pmatrix} = \begin{pmatrix} \mu_x & \widetilde{\mu}_{x+1} & \ldots & \widetilde{\mu}_{2x} \\ 0 & \mu_x & \ldots & \widetilde{\mu}_{2x-1} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & \mu_x \end{pmatrix} \begin{pmatrix} q_0 \\ \vdots \\ \vdots \\ q_x \end{pmatrix}. \tag{3.23}$$

Z regularity matice soustavy (3.23) a z toho, že $\pi_{t_k} \neq 0$ plyne, že existují a jsou jednoznačně určeny konstanta C_k a monický polynom $\mathcal{Q}_k$ takové, že polynom $\mathcal{X}_{k+1}$ je kolmý na všechny polynomy stupně nižšího než $n_k + \delta_k$ a tudíž $\mathcal{X}_{k+1} = \mathcal{M}_{k+1}$. ■

Lemma 3.11 : *Nechť $d_{n_k} \neq 0$ a $d_{n_{k+1}} \neq 0$ pro $n_k < n_{k+1}$ a nechť všechny regulární polynomy $\mathcal{P}_i$ a $\mathcal{M}_i$ $(n_k < i \leq n_{k+1})$ je možné sestrojit pomocí regulárních polynomů $\mathcal{P}_j$ a $\mathcal{M}_j$ $(j < i)$ dle lemmatu 3.10 ($\mathcal{P}_j$ a $\mathcal{P}_i$ jsou po sobě následující regulární polynomy). Označme $\delta_k = n_{k+1} - n_k$. Potom platí*

$$\begin{aligned} \mathcal{P}_{n_{k+1}}(\xi) &= \mathcal{P}_{n_k}(\xi) - \xi \mathcal{W}_{n_k}(\xi)\mathcal{M}_{n_k}(\xi) - \xi \mathcal{V}_{n_k}(\xi)\mathcal{P}_{n_k}(\xi), & (3.24)\\ \mathcal{M}_{n_{k+1}}(\xi) &= \mathcal{Q}_{n_k}(\xi)\mathcal{M}_{n_k}(\xi) + \mathcal{T}_{n_k}(\xi)\mathcal{P}_{n_k}(\xi), & (3.25) \end{aligned}$$

kde $\mathcal{W}_{n_k}(\xi)$ je polynom stupně nejvýše $\delta_k - 1$, $\mathcal{Q}_{n_k}(\xi)$ je monický polynom stupně δ_k, $\mathcal{V}_{n_k}(\xi)$ polynom stupně nejvýše $\delta_k - 2$, $\mathcal{T}_{n_k}(\xi)$ polynom stupně nejvýše $\delta_k - 1$. Je-li $\delta_k = 1$ je $\mathcal{V}_k \equiv 0$.

Důkaz : Pišme v dalším $\mathcal{P}_{s_1}$ místo $\mathcal{P}_{s_1}(\xi)$ a podobně s ostatními polynomy. Označme $\mathcal{P}_{s_1}, \ldots, \mathcal{P}_{s_z}$ posloupnost po sobě následujících regulárních polynomů, kde $s_1 = n_k$ a $s_z = n_{k+1}$ a označme dále $\Delta_j = s_{j+1} - s_j$ pro $j = 1, \ldots, z - 1$.

Je-li $z = 2$, jsou $\mathcal{P}_{n_k}$ a $\mathcal{P}_{n_{k+1}}$ po sobě následující regulární polynomy a tudíž dle lemmatu 3.10 platí vztahy (3.24) a (3.25) a $\mathcal{V}_k \equiv 0$. Nechť nyní $z > 2$. Polynom $\mathcal{P}_{s_z}$ lze podle lemmatu 3.10 rozepsat jako

$$\mathcal{P}_{s_z} = \mathcal{P}_{s_{z-1}} - \xi \mathcal{W}_{s_{z-1}} \mathcal{M}_{s_{z-1}}.$$

Polynomy $\mathcal{P}_{s_{z-1}}$ a $\mathcal{M}_{s_{z-1}}$ lze opět dle lemmatu 3.10 rozepsat pomocí polynomů $\mathcal{P}_{s_{z-2}}$ a $\mathcal{M}_{s_{z-2}}$ a dostáváme

$$(3.26)\quad \mathcal{P}_{s_z} = (\mathcal{P}_{s_{z-2}} - \xi \mathcal{W}_{s_{z-2}} \mathcal{M}_{s_{z-2}}) - \xi \mathcal{W}_{s_{z-1}} (\mathcal{Q}_{s_{z-2}} \mathcal{M}_{s_{z-2}} + C_{s_{z-2}} \xi^{t_{s_{z-2}}} \mathcal{P}_{s_{z-2}}).$$

Takto můžeme aplikovat lemma 3.10, dokud nedostaneme vyjádření polynomu $\mathcal{P}_{s_z}$ pomocí polynomů $\mathcal{P}_{s_1}$ a $\mathcal{M}_{s_1}$. Z (3.26) je vidět, že polynom s nejvyšším stupněm, který bude násobit polynom $\mathcal{M}_{s_1}$ je polynom

$$\xi \mathcal{W}_{s_{z-1}} \cdot \mathcal{Q}_{s_{z-2}} \cdot \mathcal{Q}_{s_{z-3}} \cdot \ldots \cdot \mathcal{Q}_{s_1},$$

který má stupeň nejvýše $\sum_{i=1}^{z-1} \Delta_i = \delta_k$. Zároveň každý polynom, který násobí $\mathcal{M}_{s_1}$ je dělitelný ξ. Z (3.26) je vidět, že polynom s nejvyšším stupněm, který bude násobit polynom $\mathcal{P}_{s_1}$ je polynom

$$\xi \mathcal{W}_{s_{z-1}} \cdot \mathcal{Q}_{s_{z-2}} \cdot \mathcal{Q}_{s_{z-3}} \cdot \ldots \cdot \mathcal{Q}_{s_2} \cdot \xi^{t_{s_1}},$$

který má stupeň nejvýše $\sum_{i=1}^{z-1} \Delta_i - 1 = \delta_k - 1$. Zároveň každý polynom, který násobí $\mathcal{P}_{s_1}$ (pomineme-li samotný polynom $\mathcal{P}_{s_1}$) je dělitelný ξ. Zřejmě tedy platí

$$\mathcal{P}_{s_z} = \mathcal{P}_{s_1} - \xi \mathcal{W}_{s_1} \mathcal{M}_{s_1} - \xi \mathcal{V}_{s_1} \mathcal{P}_{s_1},$$

kde $\mathcal{W}_{s_1}$ je polynom stupně nejvýše $\delta_k - 1$ a $\mathcal{V}_{s_1}$ je polynom stupně nejvýše $\delta_k - 2$. Podobně by se dokazoval vztah (3.25) . ■

Poznámka : Lemma 3.11 nám umožní setrojit regulární polynomy $\mathcal{P}_{n_{k+1}}$ a $\mathcal{M}_{n_{k+1}}$ z regulárních polynomů $\mathcal{P}_{n_k}$ a $\mathcal{M}_{n_k}$, pokud známe délku skoku δ_k. Koeficienty polynomů $\mathcal{W}_{n_k}(\xi)$, $\mathcal{V}_{n_k}(\xi)$, $\mathcal{Q}_{n_k}(\xi)$, $\mathcal{T}_{n_k}(\xi)$ získáme jako řešení soustav rovnic, které vyplynou z ortogonálních podmínek, jež mají splňovat polynomy $\mathcal{P}_{n_{k+1}}$ a $\mathcal{M}_{n_{k+1}}$. Pišme navíc v dalším pro zjednodušení zápisu $\mathcal{M}_k$, $\mathcal{P}_k$, $\mathcal{W}_k$, $\mathcal{V}_k$, $\mathcal{Q}_k$, $\mathcal{T}_k$ místo $\mathcal{M}_{n_k}(\xi)$, $\mathcal{P}_{n_k}(\xi)$, $\mathcal{W}_{n_k}(\xi)$, $\mathcal{V}_{n_k}(\xi)$, $\mathcal{Q}_{n_k}(\xi)$, $\mathcal{T}_{n_k}(\xi)$ a $\mathcal{M}_{k+1}, \mathcal{P}_{k+1}$ místo $\mathcal{M}_{n_{k+1}}(\xi), \mathcal{P}_{n_{k+1}}(\xi)$.

Poznámka : Dále předpokládejme, že $\mathcal{P}_k$, $\mathcal{M}_k$, $\mathcal{P}_{k+1}$, $\mathcal{M}_{k+1}$ jsou regulární polynomy a platí $n_k < n_{k+1}$, $\delta_k = n_{k+1} - n_k$. (Předpokládejme, že δ_k je dáno. O volbě δ_k budeme hovořit později.)

Poznámka : Vztahy (3.24) a (3.25) můžeme psát ve zjednodušeném zápisu jako

$$\begin{aligned} \mathcal{P}_{k+1} &= \mathcal{P}_k - \xi \mathcal{W}_k \mathcal{M}_k - \xi \mathcal{V}_k \mathcal{P}_k, \\ \mathcal{M}_{k+1} &= \mathcal{Q}_k \mathcal{M}_k + \mathcal{T}_k \mathcal{P}_k. \end{aligned}$$

Poznámka : Přenásobíme-li (3.24) ξ^i $(i = 0, \ldots, n_k + \delta_k - 1)$ a aplikujeme-li na rovnici funkcionál π, dostáváme rovnost

$$\pi(\xi^i \mathcal{P}_k) = \mu(\xi^i \mathcal{W}_k \mathcal{M}_k) + \pi(\xi^{i+1} \mathcal{V}_k \mathcal{P}_k).$$

Nyní je nutno rozlišit dva případy a sice $n_k \geq \delta_k - 1$ a $n_k < \delta_k - 1$. V prvním případě je totiž předchozí rovnost splněna triviálně pro $i = 0, \ldots, n_k - \delta_k$ a pro hodnoty $i = n_k - \delta_k + 1, \ldots, n_k + \delta_k - 1$ dostáváme $2\delta_k - 1$ rovnic pro $2\delta_k - 1$ neznámých (koeficienty polynomů $\mathcal{W}_k$ a $\mathcal{V}_k$).

V druhém případě dostáváme pouze $n_k + \delta_k$ rovnic pro $2\delta_k - 1$ neznámých a je $n_k + \delta_k < 2\delta_k - 1$. Volme tedy v tomto případě polynom $\mathcal{V}_k$ stupně nejvýše $n_k - 1$. Potom dostaneme $n_k + \delta_k$ rovnic pro $n_k + \delta_k$ neznámých.

Podobně se zachováme ve vztahu (3.25), kde při $n_k < \delta_k$ volíme polynom $\mathcal{T}_k$ stupně nejvýše $n_k - 1$.

Poznámka : Vyjdeme-li z ortogonálních podmínek, které má splňovat polynom $\mathcal{P}_{k+1}$, dostaneme soustavu rovnic pro koeficienty polynomů $\mathcal{W}_k$ a $\mathcal{V}_k$. Jsou-li splněny předpoklady lemmatu 3.11, víme že takové polynomy existují, tedy existuje řešení dané soustavy. Nepodařilo se nám však nic dokázat o jednoznačnosti určení polynomů $\mathcal{W}_k$ a $\mathcal{V}_k$ a tudíž nevíme nic o regularitě soustavy pro koeficienty polynomů $\mathcal{W}_k$ a $\mathcal{V}_k$.

3.4 Konstrukce regulárních polynomů $\mathcal{P}_{k+1}$ a $\mathcal{M}_{k+1}$

V tomto odstavci se budeme snažit zkonstruovat polynomy $\mathcal{P}_{k+1}$ a $\mathcal{M}_{k+1}$. Z podmínek ortogonality a s využitím lemmatu 3.11 odvodíme soustavy rovnic pro koeficienty polynomů $\mathcal{W}_{n_k}$, $\mathcal{V}_{n_k}$, $\mathcal{Q}_{n_k}$ a $\mathcal{T}_{n_k}$. Jakmile budeme znát tyto polynomy, můžeme dle lemmatu 3.11 zkonstruovat polynomy $\mathcal{P}_{k+1}$ a $\mathcal{M}_{k+1}$.

3.4.1 Konstrukce $\mathcal{P}_{k+1}$ pro $n_k \geq \delta_k - 1$

Přenásobíme-li rovnost (3.24) ξ^i a aplikujeme-li na obě strany funkcionál π, dostáváme

$$\pi(\xi^i \mathcal{P}_{k+1}) = \pi(\xi^i \mathcal{P}_k) - \mu(\xi^i \mathcal{W}_k \mathcal{M}_k) - \pi(\xi^{i+1} \mathcal{V}_k \mathcal{P}_k). \tag{3.27}$$

Polynomy $\mathcal{W}_k$ a $\mathcal{V}_k$ uvažujme ve tvaru

$$\mathcal{W}_k(\xi) = \sum_{j=0}^{\delta_k - 1} w_j \xi^j \quad \text{a} \quad \mathcal{V}_k(\xi) = \sum_{j=0}^{\delta_k - 2} v_j \xi^j. \tag{3.28}$$

Protože je $\pi(\xi^i \mathcal{P}_{k+1}) = 0$ pro $i = 0, \ldots n_k + \delta_k - 1$, můžeme (3.27) pro tato i psát ve tvaru

$$\pi(\xi^i \mathcal{P}_k) = \mu(\xi^i \mathcal{W}_k \mathcal{M}_k) + \pi(\xi^{i+1} \mathcal{V}_k \mathcal{P}_k). \tag{3.29}$$

Je-li $0 \leq i < n_k - \delta_k + 1$, je (3.29) splněno triviálně (všechny výrazy jsou nulové). Pro $i = n_k - \delta_k + 1, \ldots, n_k + \delta_k - 1$ dostáváme pro koeficienty polynomů $\mathcal{W}_k$ a $\mathcal{V}_k$ soustavu $2\delta_k - 1$ rovnic o $2\delta_k - 1$ neznámých. Rozdělme rovnice (3.29) na dvě skupiny.

- Pro $i = n_k - \delta_k + 1, \ldots, n_k - 1$.
- Pro $i = n_k, \ldots, n_k + \delta_k - 1$.

Případ $i = n_k - \delta_k + 1, \ldots, n_k - 1$.

Pro tato i můžeme (3.29) psát ve tvaru

$$0 = \mu(\xi^i \mathcal{W}_k \mathcal{M}_k) + \pi(\xi^{i+1} \mathcal{V}_k \mathcal{P}_k). \tag{3.30}$$

Označení : Označme

$$w^{(j)} = \begin{pmatrix} w_0 \\ \vdots \\ w_{j-1} \end{pmatrix}, \quad \widetilde{w}^{(j)} = \begin{pmatrix} \widetilde{w}_0 \\ \vdots \\ \widetilde{w}_{j-1} \end{pmatrix}, v^{(j)} = \begin{pmatrix} v_0 \\ \vdots \\ v_{j-1} \end{pmatrix}, \quad \widetilde{v}^{(j)} = \begin{pmatrix} \widetilde{v}_0 \\ \vdots \\ \widetilde{v}_{j-1} \end{pmatrix},$$

$$A_j^{(k)} = \begin{pmatrix} 1 & m_{n_k-1}^{(k)} & m_{n_k-2}^{(k)} & \ldots & m_{n_k-j+1}^{(k)} \\ 0 & 1 & m_{n_k-1}^{(k)} & \ldots & m_{n_k-j+2}^{(k)} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & \ddots & m_{n_k-1}^{(k)} \\ 0 & \ldots & 0 & 0 & 1 \end{pmatrix},$$

přičemž čísla $\widetilde{w}_0, \ldots, \widetilde{w}_{j-1}$ a $\widetilde{v}_0, \ldots, \widetilde{v}_{j-1}$ budou určeny později.

Lemma 3.12 : *Platí*

$$\mathcal{W}_k \xi^{n_k-\delta_k+l} = \Gamma_l + \sum_{j=0}^{l-1} \widetilde{w}_{\delta_k-l+j} \xi^j \mathcal{M}_k \quad \text{pro} \quad l = 1, \ldots, \delta_k, \tag{3.31}$$

kde $\Gamma_l(\xi)$ je polynom stupně nejvýše $n_k - 1$ a koeficienty $\widetilde{w}_0, \ldots, \widetilde{w}_{\delta_k-1}$ jsou řešením soustavy

$$w^{(\delta_k)} = A_{\delta_k}^{(k)} \widetilde{w}^{(\delta_k)}. \tag{3.32}$$

Důkaz : Tvrzení dokážeme postupně pro $l = 1, \ldots, \delta_k$.

- $l = 1$: $\mathcal{W}_k \xi^{n_k-\delta_k+1}$ je polynom stupně nejvýše n_k, lze ho tedy napsat ve tvaru

  $$\mathcal{W}_k \xi^{n_k-\delta_k+1} = \Gamma_1(\xi) + \widetilde{w}_{\delta_k-1} \mathcal{M}_k, \tag{3.33}$$

  kde $st(\Gamma_1) < n_k$. (Je-li $st(\mathcal{W}_k \xi^{n_k-\delta_k+1}) < n_k$, potom je $\Gamma_1 = \mathcal{W}_k \xi^{n_k-\delta_k+1}$ a $\widetilde{w}_{\delta_k-1} = 0$. Jinak vydělíme $\mathcal{W}_k \xi^{n_k-\delta_k+1}$ polynomem $\mathcal{M}_k$ a dostáváme rovnost $\frac{\mathcal{W}_k \xi^{n_k-\delta_k+1}}{\mathcal{M}_k} = \widetilde{w}_{\delta_k-1} + \frac{\Gamma_1}{\mathcal{M}_k}$, kde $st(\Gamma_1) < n_k$.)
- $l = 2$: Přenásobíme-li (3.33) ξ, dostáváme $\mathcal{W}_k \xi^{n_k-\delta_k+2} = \xi \Gamma_1 + \widetilde{w}_{\delta_k-1} \xi \mathcal{M}_k$. Polynom $\xi \Gamma_1$ je polynom stupně nejvýše n_k, je možné ho tedy psát ve tvaru $\xi \Gamma_1 = \Gamma_2 + \widetilde{w}_{\delta_k-2} \mathcal{M}_k$ kde $st(\Gamma_2) < n_k$. Celkem tedy máme

  $$\mathcal{W}_k \xi^{n_k-\delta_k+2} = \Gamma_2 + \widetilde{w}_{\delta_k-1} \xi \mathcal{M}_k + \widetilde{w}_{\delta_k-2} \mathcal{M}_k.$$

- Indukcí dostáváme (3.31) .

Pro $l = \delta_k$ máme rovnici

$$\mathcal{W}_k \xi^{n_k} = \Gamma_{\delta_k} + \widetilde{w}_{\delta_k-1} \xi^{\delta_k-1} \mathcal{M}_k + \widetilde{w}_{\delta_k-2} \xi^{\delta_k-2} \mathcal{M}_k + \ldots + \widetilde{w}_0 \mathcal{M}_k. \tag{3.34}$$

Rozepíšeme-li si $\mathcal{W}_k$ ve tvaru (3.28) a $\mathcal{M}_k$ ve tvaru (3.25) , dostáváme srovnáním koeficientů u stejných mocnin ξ (3.32) . ■

Lemma 3.13 : *Platí*

$$(3.35)\qquad \mathcal{V}_k\xi^{n_k-\delta_k+l+1} = \Theta_l(\xi) + \sum_{j=0}^{l-1} \widetilde{v}_{\delta_k-l+j-1}\xi^j\mathcal{M}_k \quad \text{pro} \quad l = 1,\dots,\delta_k-1.$$

$\Theta_l(\xi)$ je polynom stupně nejvýše n_k-1 a koeficienty $\widetilde{v}_0,\dots,\widetilde{v}_{\delta_k-2}$ jsou řešením soustavy

$$(3.36)\qquad v^{(\delta_k-1)} = A^{(k)}_{\delta_k-1}\widetilde{v}^{(\delta_k-1)}.$$

Důkaz : Je analogií důkazu předchozího lemmatu. ■

Lemma 3.14 : *Platí*

$$\begin{pmatrix} 0\\ 0\\ \vdots\\ 0 \end{pmatrix} = \begin{pmatrix} 0 & \mu_0 & \mu_1 & \dots & \mu_{\delta_k-2}\\ 0 & 0 & \mu_0 & \dots & \mu_{\delta_k-3}\\ \vdots & \vdots & \ddots & \ddots & \vdots\\ 0 & 0 & \dots & 0 & \mu_0 \end{pmatrix} \begin{pmatrix} \widetilde{w}_0\\ \widetilde{w}_1\\ \vdots\\ \widetilde{w}_{\delta_k-1} \end{pmatrix} + \begin{pmatrix} \pi_0 & \pi_1 & \dots & \pi_{\delta_k-2}\\ 0 & \pi_0 & \dots & \pi_{\delta_k-3}\\ \vdots & \ddots & \ddots & \vdots\\ 0 & \dots & 0 & \pi_0 \end{pmatrix} \begin{pmatrix} \widetilde{v}_0\\ \widetilde{v}_1\\ \vdots\\ \widetilde{v}_{\delta_k-2} \end{pmatrix}.$$

Důkaz : Rovnice (3.30) můžeme psát ve tvaru

$$(3.37)\qquad 0 = \mu(\mathcal{W}_k\xi^{n_k-\delta_k+l}\mathcal{M}_k) + \pi(\mathcal{V}_k\xi^{n_k-\delta_k+l+1}\mathcal{P}_k) \quad \text{pro} \quad l = 1,\dots,\delta_k-1.$$

Dosadíme-li do (3.37) za $\mathcal{W}_k\xi^{n_k-\delta_k+l}$ a za $\mathcal{V}_k\xi^{n_k-\delta_k+l+1}$ (3.31) a (3.35) , dostáváme soustavu rovnic

$$(3.38)\qquad 0 = \mu((\Gamma_l + \sum_{j=0}^{l-1}\widetilde{w}_{\delta_k-l+j}\xi^j\mathcal{M}_k)\mathcal{M}_k) + \pi((\Theta_l + \sum_{j=0}^{l-1}\widetilde{v}_{\delta_k-l+j-1}\xi^j\mathcal{M}_k)\mathcal{P}_k).$$

Využijeme-li toho, že $\mu(\Gamma_l\mathcal{M}_k) = 0$ a $\pi(\Theta_l\mathcal{P}_k) = 0$, můžeme (3.38) psát ve tvaru

$$0 = \sum_{j=0}^{l-1}\widetilde{w}_{\delta_k-l+j}\mu(\xi^j\mathcal{M}_k^2) + \sum_{j=0}^{l-1}\widetilde{v}_{\delta_k-l+j-1}\pi(\xi^j\mathcal{M}_k\mathcal{P}_k) = \sum_{j=0}^{l-1}\widetilde{w}_{\delta_k-l+j}\mu_j + \sum_{j=0}^{l-1}\widetilde{v}_{\delta_k-l+j-1}\pi_j.$$

Rozepíšeme-li si tyto rovnice pro $l = \delta_k-1,\dots,1$ dostáváme tvrzení lemmatu. ■

Případ $i = n_k,\dots,n_k+\delta_k-1$.

Lemma 3.15 : Platí

(3.39)

$$\begin{pmatrix} \pi_0\\ \vdots\\ \pi_{\delta_k-1} \end{pmatrix} = \begin{pmatrix} \mu_0 & \dots & \mu_{\delta_k-1}\\ \vdots & & \vdots\\ \mu_{\delta_k-1} & \dots & \mu_{2\delta_k-2} \end{pmatrix} A_{\delta_k}\widetilde{w}^{(\delta_k)} + \begin{pmatrix} \pi_1 & \dots & \pi_{\delta_k-1}\\ \vdots & & \vdots\\ \pi_{\delta_k} & \dots & \pi_{2\delta_k-2} \end{pmatrix} A_{\delta_k-1}\widetilde{v}^{(\delta_k-1)}.$$

Důkaz : Dle lemmatu 3.8 můžeme podmínky $\pi(\xi^i\mathcal{P}_{k+1}) = 0$ pro $i = n_k,\dots,n_k+\delta_k-1$ psát ve tvaru

$$(3.40)\qquad \pi(\xi^l\mathcal{M}_k\mathcal{P}_{k+1}) = 0 \quad \text{pro} \quad l = 0,\dots,\delta_k-1.$$

Dosadíme-li nyní do (3.40) $\mathcal{P}_{k+1}$ ve tvaru (3.24) , dostáváme pro $l = 0,\ldots,\delta_k - 1$ soustavu rovnic

$$\pi(\xi^l \mathcal{M}_k \mathcal{P}_k) = \mu(\xi^l \mathcal{W}_k \mathcal{M}_k^2) + \pi(\xi^{l+1} \mathcal{V}_k \mathcal{M}_k \mathcal{P}_k). \tag{3.41}$$

Dosadíme-li nyní do (3.41) polynomy $\mathcal{W}_k$ a $\mathcal{V}_k$ ve tvaru (3.28) , můžeme (3.41) psát ve tvaru

$$\pi(\xi^l \mathcal{M}_k \mathcal{P}_k) = \sum_{j=0}^{\delta_k - 1} w_j \mu(\xi^{j+l} \mathcal{M}_k^2) + \sum_{j=0}^{\delta_k - 2} v_j \pi(\xi^{j+l+1} \mathcal{M}_k \mathcal{P}_k),$$

neboli

$$\pi_l = \sum_{j=0}^{\delta_k - 1} w_j \mu_{j+l} + \sum_{j=0}^{\delta_k - 2} v_j \pi_{j+l+1}. \tag{3.42}$$

Uvážíme-li, že vektory $w^{(\delta_k)}$ a $v^{(\delta_k - 1)}$ lze psát ve tvaru (3.32) a (3.36) , můžeme (3.42) zapsat v maticovém tvaru (3.39) . ■

Definice 3.8 : Definujme čísla g_{ij} a f_{ij} následujícími vztahy

$$g_{ij} = \sum_{l=0}^{j-1} \mu_{i+l} m_{n_k - j + l}^{(n_k)} + \mu_{i+j} \quad \text{a} \quad f_{ij} = \sum_{l=0}^{j-1} \pi_{i+l+1} m_{n_k - j + l}^{(n_k)} + \pi_{i+j+1}. \tag{3.43}$$

Označíme-li kvůli přehlednějšímu zápisu

$$x = \delta_k - 1 \quad \text{a} \quad y = \delta_k - 2,$$

dostáváme pro koeficienty $\widetilde{w}_0, \ldots, \widetilde{w}_x, \widetilde{v}_0, \ldots \widetilde{v}_y$ soustavu rovnic

$$\begin{pmatrix} \pi_0 \\ \pi_1 \\ \vdots \\ \pi_x \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \left(\begin{array}{cccc|ccc} g_{00} & g_{01} & \cdots & g_{0x} & f_{00} & \cdots & f_{0y} \\ g_{10} & g_{11} & \cdots & g_{1x} & f_{10} & \cdots & f_{1y} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ g_{x0} & g_{x1} & \cdots & g_{xx} & f_{x0} & \cdots & f_{xy} \\ \hline 0 & \mu_0 & \cdots & \mu_y & \pi_0 & \cdots & \pi_y \\ \vdots & \ddots & \ddots & \vdots & 0 & \ddots & \vdots \\ 0 & \ldots & 0 & \mu_0 & 0 & 0 & \pi_0 \end{array}\right) \begin{pmatrix} \widetilde{w}_0 \\ \widetilde{w}_1 \\ \vdots \\ \widetilde{w}_x \\ \widetilde{v}_0 \\ \vdots \\ \widetilde{v}_y \end{pmatrix}. \tag{3.44}$$

Poznámka : Dostali jsme tedy soustavu rovnic pro koeficienty $\widetilde{w}_0, \ldots, \widetilde{w}_x$ a $\widetilde{v}_0, \ldots, \widetilde{v}_y$. Koeficienty polynomů $\mathcal{W}_k$ a $\mathcal{V}_k$ dostaneme dle lemmatu 3.12 a 3.13 z $w^{(x)} = A_x^{(k)} \widetilde{w}^{(x)}$ a z $v^{(y)} = A_y^{(k)} \widetilde{v}^{(y)}$. Ostatní případy probereme trochu rychleji.

3.4.2 Konstrukce $\mathcal{P}_{k+1}$ pro $n_k < \delta_k - 1$

V tomto případě lze volit $\mathcal{V}_k$ tak, že

$$\mathcal{V}_k = \sum_{j=0}^{n_k - 1} v_j \xi^j. \tag{3.45}$$

Rovnice $\pi(\xi^i \mathcal{P}_{k+1}) = 0$ opět rozdělíme na 2 případy

- $\pi(\xi^i \mathcal{P}_{k+1}) = 0$ pro $i = 0, \ldots, n_k - 1$,

- $\pi(\xi^i \mathcal{M}_k \mathcal{P}_{k+1}) = 0$ pro $i = 0, \ldots, \delta_k - 1$,

neboli

- $0 = \mu(\xi^i \mathcal{W}_k \mathcal{M}_k) + \pi(\xi^{i+1} \mathcal{V}_k \mathcal{P}_k)$ pro $i = 0, \ldots, n_k - 1$,
- $\pi(\xi^i \mathcal{M}_k \mathcal{P}_k) = \mu(\xi^i \mathcal{W}_k \mathcal{M}_k^2) + \pi(\xi^{i+1} \mathcal{V}_k \mathcal{M}_k \mathcal{P}_k)$ pro $i = 0, \ldots, \delta_k - 1$.

Nyní již bez důkazu uveďme lemmata o rozepsání polynomů $\xi^i \mathcal{W}_k$ a $\xi^{i+1} \mathcal{V}_k$.

Lemma 3.16 : Platí

$$\xi^i \mathcal{W}_k = \Gamma_i(\xi) + \sum_{j=n_k-i}^{\delta_k-1} \widetilde{w}_j \xi^{j-n_k+i} \mathcal{M}_k \quad \text{pro} \quad i = 0, \ldots, n_k, \tag{3.46}$$

kde $st(\Gamma_i) < n_k$. Dodefinujeme-li $m_i^{(k)} = 0$ pro $i < 0$, spočteme koeficienty $\widetilde{w}_j$ jako řešení soustavy

$$w^{(\delta_k)} = A_{\delta_k}^{(k)} \widetilde{w}^{(\delta_k)}.$$

Poznámka : Porovnáme-li v (3.46) pro $i = n_k$ koeficienty u příslušných mocnin, dostaneme soustavu rovnic udávající vztah mezi $\widetilde{w}^{(\delta_k)}$ a $w^{(\delta_k)}$. Matice této soustavy je řádu δ_k a má tvar

$$\begin{pmatrix} 1 & m_{n_k-1}^{(k)} & \ldots & m_0^{(k)} & 0 & \ldots & 0 \\ 0 & 1 & m_{n_k-1}^{(k)} & \ldots & m_0^{(k)} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & m_0^{(k)} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & m_{n_k-1}^{(k)} \\ 0 & \ldots & \ldots & \ldots & \ldots & 0 & 1 \end{pmatrix}.$$

Abychom nemuseli tuto matici označovat zvláštním písmenem, definujeme $m_i^{(k)} = 0$ pro $i < 0$ a potom můžeme tuto matici označovat jako $A_{\delta_k}^{(k)}$.

Lemma 3.17 : Platí

$$\xi^{i+1} \mathcal{V}_k = \Theta_i(\xi) + \sum_{j=0}^{i} \widetilde{v}_{n_k-i+j-1} \xi^j \mathcal{M}_k \quad \text{pro} \quad i = 0, \ldots, n_k - 1, \tag{3.47}$$

kde $st(\Theta_i) < n_k$. Koeficienty $\widetilde{v}_j$ spočteme jako řešení soustavy

$$v^{(n_k)} = A_{n_k}^{(k)} \widetilde{v}^{(n_k)}.$$

Dosadíme-li nyní za $\xi^i \mathcal{W}_k$ a $\xi^{i+1} \mathcal{V}_k$ do první skupiny rovnic a upravujeme-li současně druhou skupinu, dostáváme

- $0 = \mu(\sum_{j=n_k-i}^{\delta_k-1} \widetilde{w}_j \xi^{j-n_k+i} \mathcal{M}_k^2) + \pi(\sum_{j=0}^{i} \widetilde{v}_{n_k-i+j-1} \xi^j \mathcal{M}_k \mathcal{P}_k)$ pro $i = 0, \ldots, n_k - 1$,
- $\pi_i = \sum_{j=0}^{\delta_k-1} w_j \mu_{i+j} + \sum_{j=0}^{n_k-1} v_j \pi_{j+i+1}$ pro $i = 0, \ldots, \delta_k - 1$.

Dodefinujeme-li $m_i^{(k)} = 0$ pro $i < 0$ a

$$x = \delta_k - 1, \quad y = n_k - 1,$$

dostaneme postupnými úpravami pro koeficienty $\widetilde{w}_0, \dots, \widetilde{w}_x, \widetilde{v}_0, \dots, \widetilde{v}_y$ soustavu

$$\text{(3.48)} \qquad \begin{pmatrix} \pi_0 \\ \pi_1 \\ \vdots \\ \pi_x \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \left(\begin{array}{cccccc|ccc} g_{00} & g_{01} & \dots & \dots & \dots & g_{0x} & f_{00} & \dots & f_{0y} \\ g_{10} & g_{11} & \dots & \dots & \dots & g_{1x} & f_{10} & \dots & f_{1y} \\ \vdots & \vdots & & & & \vdots & \vdots & & \vdots \\ g_{x0} & g_{x1} & \dots & \dots & \dots & g_{xx} & f_{x0} & \dots & f_{xy} \\ \hline 0 & \mu_0 & \dots & \dots & \dots & \mu_{x-1} & \pi_0 & \dots & \pi_y \\ \vdots & \ddots & \ddots & & & \vdots & 0 & \ddots & \vdots \\ 0 & \dots & 0 & \mu_0 & \dots & \mu_{x-y-1} & 0 & 0 & \pi_0 \end{array}\right) \begin{pmatrix} \widetilde{w}_0 \\ \widetilde{w}_1 \\ \vdots \\ \widetilde{w}_x \\ \widetilde{v}_0 \\ \vdots \\ \widetilde{v}_y \end{pmatrix}.$$

Poznámka : Koeficienty polynomů $\mathcal{W}_k$ a $\mathcal{V}_k$ dostaneme podle lemmatu 3.16 a 3.17 jako $w^{(x)} = A_x^{(k)} \widetilde{w}^{(x)}$ a $v^{(y)} = A_y^{(k)} \widetilde{v}^{(y)}$.

3.4.3 Konstrukce $\mathcal{M}_{k+1}$ pro $n_k \geq \delta_k$

Budeme postupovat podobně jako při konstrukci polynomu $\mathcal{P}_{k+1}$ a ortogonálních podmínek, jež má splňovat polynom $\mathcal{M}_{k+1}$, využijeme ke spočtení koeficientů polynomů $\mathcal{Q}_k$ a $\mathcal{T}_k$. Přenásobíme-li rovnost (3.25) ξ^i a aplikujeme-li na obě strany funkcionál μ, dostáváme

$$\text{(3.49)} \qquad \mu(\xi^i \mathcal{M}_{k+1}) = \mu(\xi^i \mathcal{Q}_k \mathcal{M}_k) + \pi(\xi^{i+1} \mathcal{T}_k \mathcal{P}_k).$$

Pro $i = 0, \dots, n_k + \delta_k - 1$ pak můžeme (3.49) psát ve tvaru

$$\text{(3.50)} \qquad 0 = \mu(\xi^i \mathcal{Q}_k \mathcal{M}_k) + \pi(\xi^{i+1} \mathcal{T}_k \mathcal{P}_k).$$

Polynomy $\mathcal{Q}_k$ a $\mathcal{T}_k$ uvažujme ve tvaru

$$\text{(3.51)} \qquad \mathcal{Q}_k(\xi) = \sum_{j=0}^{\delta_k - 1} q_j \xi^j + \xi^{\delta_k} \quad \text{a} \quad \mathcal{T}_k(\xi) = \sum_{j=0}^{\delta_k - 1} t_j \xi^j.$$

Je-li $0 \leq i \leq n_k - \delta_k - 1$, je rovnost (3.50) splněna triviálně.

Pro $i = n_k - \delta_k, \dots, n_k + \delta_k - 1$ dostáváme $2\delta_k$ rovnic pro $2\delta_k$ neznámých.

Případ $i = n_k - \delta_k, \dots, n_k - 1$.

Označení : Označme

$$q^{(j)} = \begin{pmatrix} q_0 \\ \vdots \\ q_{j-1} \end{pmatrix}, \quad \widetilde{q}^{(j)} = \begin{pmatrix} \widetilde{q}_0 \\ \vdots \\ \widetilde{q}_{j-1} \end{pmatrix}, \quad t^{(j)} = \begin{pmatrix} t_0 \\ \vdots \\ t_{j-1} \end{pmatrix}, \quad \widetilde{t}^{(j)} = \begin{pmatrix} \widetilde{t}_0 \\ \vdots \\ \widetilde{t}_{j-1} \end{pmatrix},$$

přičemž čísla $\widetilde{q}_0, \dots, \widetilde{q}_{j-1}$ a $\widetilde{t}_0, \dots, \widetilde{t}_{j-1}$ budou určeny později.

Lemma 3.18 : *Platí*

$$\text{(3.52)} \qquad \mathcal{Q}_k \xi^{n_k - \delta_k + l} = \Phi_l + \sum_{j=0}^{l-1} \widetilde{q}_{\delta_k - l + j} \xi^j \mathcal{M}_k + \xi^l \mathcal{M}_k \quad \textit{pro} \quad l = 0, \dots, \delta_k.$$

$\Phi_l(\xi)$ je polynom stupně nejvýše $n_k - 1$ a koeficienty $\widetilde{q}_0, \ldots, \widetilde{q}_{\delta_k - 1}$ jsou řešením soustavy

$$q^{(\delta_k)} = A^{(k)}_{\delta_k} \widetilde{q}^{(\delta_k)} + \begin{pmatrix} m^{(k)}_{n_k - \delta_k} \\ \vdots \\ m^{(k)}_{n_k - 1} \end{pmatrix}. \tag{3.53}$$

Důkaz : Je obdobou dukazu lemmatu 3.12. ■

Lemma 3.19 : *Platí*

$$\mathcal{T}_k \xi^{n_k - \delta_k + l + 1} = \Upsilon_l + \sum_{j=0}^{l} \widetilde{t}_{\delta_k - l + j - 1} \xi^j \mathcal{M}_k \quad pro \quad l = 0, \ldots, \delta_k - 1. \tag{3.54}$$

Υ_l je polynom stupně nejvýše $\delta_k - 1$ a koeficienty $\widetilde{t}_0, \ldots, \widetilde{t}_{\delta_k - 1}$ jsou řešením soustavy

$$t^{(\delta_k)} = A^{(k)}_{\delta_k} \widetilde{t}^{(\delta_k)}. \tag{3.55}$$

Důkaz : Je analogií důkazu lemmatu 3.12. ■

Lemma 3.20 : Platí

$$-\begin{pmatrix} \mu_{\delta_k - 1} \\ \mu_{\delta_k - 2} \\ \vdots \\ \mu_0 \end{pmatrix} = \begin{pmatrix} 0 & \mu_0 & \ldots & \mu_{\delta_k - 2} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \ddots & \mu_0 \\ 0 & 0 & \ldots & 0 \end{pmatrix} \begin{pmatrix} \widetilde{q}_0 \\ \widetilde{q}_1 \\ \vdots \\ \widetilde{q}_{\delta_k - 1} \end{pmatrix} + \begin{pmatrix} \pi_0 & \pi_1 & \ldots & \pi_{\delta_k - 1} \\ 0 & \pi_0 & \ldots & \pi_{\delta_k - 2} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & \pi_0 \end{pmatrix} \begin{pmatrix} \widetilde{t}_0 \\ \widetilde{t}_1 \\ \vdots \\ \widetilde{t}_{\delta_k - 1} \end{pmatrix}. \tag{3.56}$$

Důkaz : Rovnice (3.50) můžeme psát ve tvaru

$$0 = \mu(\mathcal{Q}_k \xi^{n_k - \delta_k + l} \mathcal{M}_k) + \pi(\mathcal{T}_k \xi^{n_k - \delta_k + l + 1} \mathcal{P}_k) \quad \text{pro} \quad l = 0, \ldots, \delta_k - 1. \tag{3.57}$$

Dosadíme-li do (3.57) za $\mathcal{Q}_k \xi^{n_k - \delta_k + l}$ a za $\mathcal{T}_k \xi^{n_k - \delta_k + l + 1}$ (3.52) a (3.54) , dostáváme analogicky dle důkazu lemmatu 3.14 soustavu (3.56) . ■

Případ $i = n_k, \ldots, n_k + \delta_k - 1$.

Lemma 3.21 : Platí

$$-\begin{pmatrix} g_{0,\delta_k} \\ g_{1,\delta_k} \\ \vdots \\ g_{x,\delta_k} \end{pmatrix} = \begin{pmatrix} g_{00} & \ldots & g_{0x} \\ \vdots & \ddots & \vdots \\ g_{x0} & \ldots & g_{xx} \end{pmatrix} \widetilde{q}^{(\delta_k)} + \begin{pmatrix} f_{00} & \ldots & f_{0x} \\ \vdots & \ddots & \vdots \\ f_{x0} & \ldots & f_{xx} \end{pmatrix} \widetilde{t}^{(\delta_k)}. \tag{3.58}$$

Důkaz : Dle lemmatu 3.9 můžeme podmínky $\mu(\xi^i \mathcal{M}_{k+1}) = 0$ pro $i = n_k, \ldots, n_k + \delta_k - 1$ psát ve tvaru

$$\mu(\xi^l \mathcal{M}_k \mathcal{M}_{k+1}) = 0 \quad \text{pro} \quad l = 0, \ldots, \delta_k - 1. \tag{3.59}$$

Dosadíme-li nyní do (3.59) $\mathcal{M}_{k+1}$ ve tvaru (3.25) , dostáváme pro $l = 0, \ldots, \delta_k - 1$ soustavu rovnic

$$0 = \mu(\xi^l \mathcal{Q}_k \mathcal{M}_k^2) + \pi(\xi^{l+1} \mathcal{T}_k \mathcal{M}_k \mathcal{P}_k). \tag{3.60}$$

Dosadíme-li nyní do (3.60) polynomy $\mathcal{Q}_k$ a $\mathcal{T}_k$ ve tvaru (3.51) , můžeme (3.60) psát ve tvaru

$$-\mu(\xi^{\delta_k+l}\mathcal{M}_k^2) = \sum_{j=0}^{\delta_k-1} q_j\mu(\xi^{j+l}\mathcal{M}_k^2) + \sum_{j=0}^{\delta_k-1} t_j\pi(\xi^{j+l+1}\mathcal{M}_k\mathcal{P}_k). \tag{3.61}$$

Definujme

$$x = \delta_k - 1 \quad \text{a} \quad y = \delta_k - 2.$$

Uvážíme-li, že vektory $q^{(\delta_k)}$ a $t^{(\delta_k)}$ lze psát ve tvaru (3.53) a (3.55) , můžeme (3.61) zapsat v maticovém tvaru

$$\begin{aligned}
-\begin{pmatrix} \mu_{x+1} \\ \vdots \\ \mu_{2x+1} \end{pmatrix} &= \begin{pmatrix} \mu_0 & \dots & \mu_x \\ \vdots & & \vdots \\ \mu_x & \dots & \mu_{2x} \end{pmatrix} \left[\begin{pmatrix} m_{n_k-\delta_k}^{(n_k)} \\ \vdots \\ m_{n_k-1}^{(n_k)} \end{pmatrix} + A_{\delta_k}^{(k)}\widetilde{q}^{(\delta_k)} \right] + \\
&+ \begin{pmatrix} \pi_1 & \dots & \pi_{x+1} \\ \vdots & & \vdots \\ \pi_{x+1} & \dots & \pi_{2x+1} \end{pmatrix} A_{\delta_k}^{(k)}\widetilde{t}^{(\delta_k)} = \\
&= \begin{pmatrix} g_{0,\delta_k} - \mu_{x+1} \\ g_{1,\delta_k} - \mu_{x+2} \\ \vdots \\ g_{x,\delta_k} - \mu_{2x+1} \end{pmatrix} + \begin{pmatrix} g_{00} & \dots & g_{0x} \\ \vdots & \ddots & \vdots \\ g_{x0} & \dots & g_{xx} \end{pmatrix} \widetilde{q}^{(\delta_k)} + \begin{pmatrix} f_{00} & \dots & f_{0x} \\ \vdots & \ddots & \vdots \\ f_{x0} & \dots & f_{xx} \end{pmatrix} \widetilde{t}^{(\delta_k)}.
\end{aligned}$$

Odečtením vektoru na pravé straně od vektoru na levé straně dostáváme (3.58) . ■

Celkem tedy dostaneme pro koeficienty soustavu

$$-\begin{pmatrix} g_{0\delta_k} \\ g_{1\delta_k} \\ \vdots \\ g_{x\delta_k} \\ \mu_x \\ \vdots \\ \vdots \\ \mu_0 \end{pmatrix} = \left(\begin{array}{cccc|cccc} g_{00} & g_{01} & \dots & g_{0x} & f_{00} & f_{01} & \dots & f_{0x} \\ g_{10} & g_{11} & \dots & g_{1x} & f_{10} & f_{11} & \dots & f_{1x} \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ g_{x0} & g_{x1} & \dots & g_{xx} & f_{x0} & f_{x1} & \dots & f_{xx} \\ \hline 0 & \mu_0 & \dots & \mu_y & \pi_0 & \pi_1 & \dots & \pi_y \\ \vdots & \ddots & \ddots & \vdots & 0 & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \mu_0 & \vdots & \ddots & \ddots & \pi_1 \\ 0 & \dots & 0 & 0 & 0 & \dots & 0 & \pi_0 \end{array}\right) \begin{pmatrix} \widetilde{q}_0 \\ \widetilde{q}_1 \\ \vdots \\ \widetilde{q}_x \\ \widetilde{t}_0 \\ \widetilde{t}_1 \\ \vdots \\ \widetilde{t}_x \end{pmatrix}. \tag{3.62}$$

Povšimeme si, že z poslední rovnice lze spočíst $\widetilde{t}_x = -\mu_0/\pi_0$ (O nenulovosti π_0 viz. příští kapitola o volbě δ_k). Dosadíme-li takto spočtené $\widetilde{t}_x$ do prvních $2\delta_k - 1$ rovnic, dostáváme pro zbývající koeficienty soustavu rovnic

$$-\begin{pmatrix} g_{0\delta_k} + f_{0x}\widetilde{t}_x \\ g_{1\delta_k} + f_{1x}\widetilde{t}_x \\ \vdots \\ g_{x\delta_k} + f_{xx}\widetilde{t}_x \\ \mu_x + \pi_y\widetilde{t}_x \\ \vdots \\ \mu_0 + \pi_1\widetilde{t}_x \end{pmatrix} = \left(\begin{array}{cccc|ccc} g_{00} & g_{01} & \dots & g_{0x} & f_{00} & \dots & f_{0y} \\ g_{10} & g_{11} & \dots & g_{1x} & f_{10} & \dots & f_{1y} \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ g_{x0} & g_{x1} & \dots & g_{xx} & f_{x0} & \dots & f_{xy} \\ \hline 0 & \mu_0 & \dots & \mu_y & \pi_0 & \dots & \pi_y \\ \vdots & \ddots & \ddots & \vdots & 0 & \ddots & \vdots \\ 0 & \dots & 0 & \mu_0 & 0 & 0 & \pi_0 \end{array}\right) \begin{pmatrix} \widetilde{q}_0 \\ \widetilde{q}_1 \\ \vdots \\ \widetilde{q}_x \\ \widetilde{t}_0 \\ \vdots \\ \widetilde{t}_y \end{pmatrix}. \tag{3.63}$$

Poznámka : Koeficienty polynomů $\mathcal{Q}_k$ a $\mathcal{T}_k$ dostaneme podle lemmatu 3.18 a 3.19 ze soustav (3.53) a (3.55).

3.4.4 Konstrukce $\mathcal{M}_{k+1}$ pro $n_k < \delta_k$

V tomto případě lze volit $\mathcal{T}_k$ tak, že

$$\mathcal{T}_k = \sum_{j=0}^{n_k-1} t_j \xi^j. \tag{3.64}$$

Uplatníme-li předchozí postup pouze s jinou volbou polynomu $\mathcal{T}_k$ a definujeme-li

$$x = \delta_k - 1, \quad y = n_k - 1, \quad m_i^{(k)} = 0 \quad \text{pro} \quad i < 0,$$

dostáváme soustavu

$$-\begin{pmatrix} g_{0\delta_k} \\ g_{1\delta_k} \\ \vdots \\ g_{x\delta_k} \\ \mu_x \\ \vdots \\ \mu_{x-y} \end{pmatrix} = \left(\begin{array}{cccccc|ccc} g_{00} & g_{01} & \dots & \dots & \dots & g_{0x} & f_{00} & \dots & f_{0y} \\ g_{10} & g_{11} & \dots & \dots & \dots & g_{1x} & f_{10} & \dots & f_{1y} \\ \vdots & \vdots & & & & \vdots & \vdots & & \vdots \\ g_{x0} & g_{x1} & \dots & \dots & \dots & g_{xx} & f_{x0} & \dots & f_{xy} \\ \hline 0 & \mu_0 & \dots & \dots & \dots & \mu_{x-1} & \pi_0 & \dots & \pi_y \\ \vdots & \ddots & \ddots & & & \vdots & 0 & \ddots & \vdots \\ 0 & \dots & 0 & \mu_0 & \dots & \mu_{x-y-1} & 0 & 0 & \pi_0 \end{array}\right) \begin{pmatrix} \widetilde{q}_0 \\ \widetilde{q}_1 \\ \vdots \\ \widetilde{q}_x \\ \widetilde{t}_0 \\ \vdots \\ \widetilde{t}_y \end{pmatrix}. \tag{3.65}$$

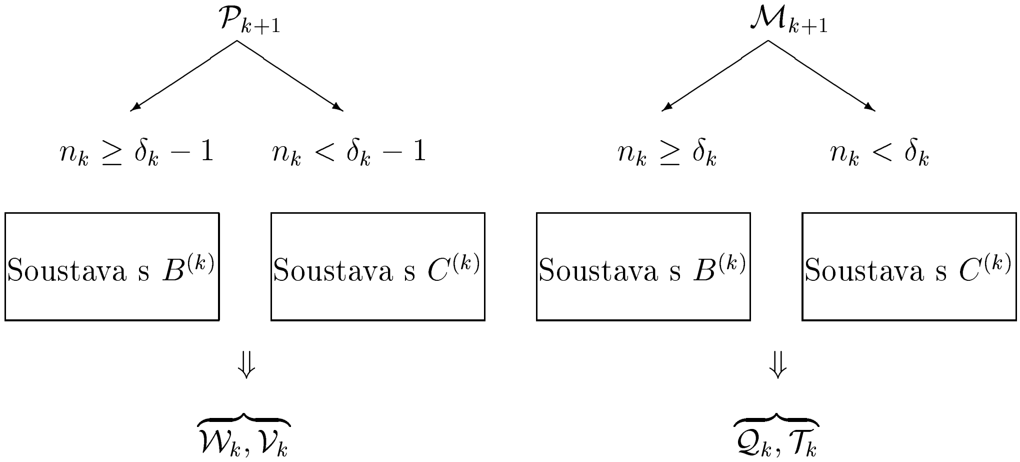
Poznámka : Vztahy 3.53 a 3.55 zůstávají v platnosti, pouze dodefinujeme $m_i^{(k)} = 0$ pro $i < 0$.

Poznámka : Povšiměme si, že matice soustavy (3.44) je stejná jako matice soustavy (3.63) a matice soustavy (3.48) je stejná jako matice soustavy (3.65) .

Označení : Označme matici soustavy (3.44) $B^{(k)}$ a matici soustavy (3.48) $C^{(k)}$.

Následujícím obrázkem shrneme schématicky počítání polynomů $\mathcal{P}_{k+1}$ a $\mathcal{M}_{k+1}$.

Shrnutí



3.5 Metoda CGS s regulárními polynomy

Zkusme nyní přizpůsobit metodu CGS tak, aby počítala pouze s regulárními polynomy, tedy tato metoda bude vypočítávat pouze některé vektory x_k a r_k. Předpokládejme, že je v každém kroku dána délka skoku δ_k. V metodě CGS je definováno reziduum jako

$$r_k = \mathcal{P}_k^2(A)r_0.$$

Poznámka : Index k nemusí nutně znamenat k-té reziduum. Použijme index k jako v předchozí kapitole k označení nějakého indexu n_k a index $k+1$ k označení nějakého indexu n_{k+1}.

Pro výpočet rezidua r_{k+1} použijeme vztahu pro polynom $\mathcal{P}_{k+1}^2$. Dostáváme

$$\mathcal{P}_{k+1}^2 = (1-\xi\mathcal{V}_k)^2\mathcal{P}_k^2 - 2(1-\xi\mathcal{V}_k)\xi\mathcal{W}_k\mathcal{P}_k\mathcal{M}_k + \xi^2\mathcal{W}_k^2\mathcal{M}_k^2. \tag{3.66}$$

Pro použití této rekurence pro výpočet rezidua r_{k+1} je nezbytné umět počítat rekurentně polynomy $\mathcal{M}_{k+1}^2$ a $\mathcal{P}_{k+1}\mathcal{M}_{k+1}$. Platí

$$\begin{aligned}
\mathcal{M}_{k+1}^2 &= \mathcal{Q}_k^2\mathcal{M}_k^2 + 2\mathcal{Q}_k\mathcal{T}_k\mathcal{P}_k\mathcal{M}_k + \mathcal{T}_k{}^2\mathcal{P}_k^2, && (3.67)\\
\mathcal{M}_{k+1}\mathcal{P}_{k+1} &= (\mathcal{Q}_k - \xi\mathcal{Q}_k\mathcal{V}_k - \xi\mathcal{T}_k\mathcal{W}_k)\mathcal{P}_k\mathcal{M}_k - \xi\mathcal{Q}_k\mathcal{W}_k\mathcal{M}_k^2 + \mathcal{T}_k(1-\xi\mathcal{V}_k)\mathcal{P}_k^2. && (3.68)
\end{aligned}$$

Definice 3.9 : Definujme vektory r_k, z_k, s_k následujícími vztahy

$$\begin{aligned}
r_k &= \mathcal{P}_k^2(A)r_0, && (3.69)\\
z_k &= \mathcal{M}_k^2(A)r_0, && (3.70)\\
s_k &= \mathcal{P}_k(A)\mathcal{M}_k(A)r_0. && (3.71)
\end{aligned}$$

Vyjádříme-li vektory r_k, z_k, s_k pomocí vztahů (3.66) , (3.67) a (3.68) dostáváme

$$\begin{aligned}
r_{k+1} &= (I - A\mathcal{V}_k)^2 r_k - 2(I - A\mathcal{V}_k)A\mathcal{W}_k s_k + A^2\mathcal{W}_k^2 z_k, && (3.72)\\
z_{k+1} &= \mathcal{Q}_k{}^2 z_k + 2\mathcal{Q}_k\mathcal{T}_k s_k + \mathcal{T}_k^2 r_k, && (3.73)\\
s_{k+1} &= (\mathcal{Q}_k - A\mathcal{Q}_k\mathcal{V}_k - A\mathcal{T}_k\mathcal{W}_k)s_k - A\mathcal{Q}_k\mathcal{W}_k z_k + \mathcal{T}_k(I - A\mathcal{V}_k)r_k. && (3.74)
\end{aligned}$$

Poznámka : Ve vztazích (3.72) - (3.74) je psáno $\mathcal{V}_k$ místo $\mathcal{V}_k(A)$. Podobně s $\mathcal{W}_k$, $\mathcal{Q}_k$, $\mathcal{T}_k$.

Lemma 3.22 : Pro vektor x_{k+1} platí

$$x_{k+1} = x_k - (A\mathcal{V}_k - 2I)\mathcal{V}_k r_k + 2(I - A\mathcal{V}_k)\mathcal{W}_k s_k - A\ \mathcal{W}_k{}^2 z_k. \tag{3.75}$$

Důkaz : Platí

$$\begin{aligned}
r_k - r_{k+1} &= b - Ax_k - b + Ax_{k+1} = A(x_{k+1} - x_k), && (3.76)\\
r_k - r_{k+1} &= r_k - (I - A\mathcal{V}_k)^2 r_k - 2(I - A\mathcal{V}_k)A\mathcal{W}_k s_k + A^2\mathcal{W}_k^2 z_k = &&\\
&= r_k - r_k - (A^2\mathcal{V}_k^2 - 2A\mathcal{V}_k)r_k - 2(I - A\mathcal{V}_k)A\mathcal{W}_k s_k + A^2\mathcal{W}_k^2 z_k. && (3.77)
\end{aligned}$$

Využijeme-li (3.76) a přenásobíme-li (3.77) maticí A^{-1} zleva dostáváme (3.75) . ■

Nyní již můžeme definovat schéma algoritmu upravené metody CGS.

Algoritmus 3.1

Input $x_0,\ A,\ b,\ \tilde{r}_0$;
$r_0 = b - Ax_0$;
$s_0 = r_0;\ z_0 = r_0$;
for $k = 0$ **to** $n-1$ **do**
  if $r_k = 0$ **then** $x^* = x_k$ *and exit*
  else
    begin

$$\boxed{?\ \text{Určení délky skoku } \delta_k;}$$

$$\boxed{\text{Výpočet } \mathcal{W}_k,\ \mathcal{V}_k,\ \mathcal{Q}_k,\ \mathcal{T}_k;}$$

$$x_{k+1} = x_k - (A\mathcal{V}_k - 2I)\mathcal{V}_k r_k + 2(I - A\mathcal{V}_k)\mathcal{W}_k s_k - A\ {\mathcal{W}_k}^2 z_k;$$
$$r_{k+1} = (I - A\mathcal{V}_k)^2 r_k - 2(I - A\mathcal{V}_k)A\mathcal{W}_k s_k + A^2\mathcal{W}_k^2 z_k;$$
$$z_{k+1} = {\mathcal{Q}_k}^2 z_k + 2\mathcal{Q}_k\mathcal{T}_k s_k + \mathcal{T}_k^2 r_k;$$
$$s_{k+1} = (\mathcal{Q}_k - A\mathcal{Q}_k\mathcal{V}_k - A\mathcal{T}_k\mathcal{W}_k)s_k - A\mathcal{Q}_k\mathcal{W}_k z_k + \mathcal{T}_k(I - A\mathcal{V}_k)r_k;$$

    end;
$x^* = x_n$.

Lemma 3.23 : Čísla π_i a μ_i lze psát ve tvaru

$$\pi_i = (\tilde{r}_0, A^i s_k), \tag{3.78}$$
$$\mu_i = (\tilde{r}_0, A^{i+1} z_k). \tag{3.79}$$

Důkaz : Využijeme-li definicí čísel π_i a μ_i, dostáváme

$$\begin{aligned}\pi_i &= \pi(\xi^i\mathcal{P}_k\mathcal{M}_k) = (\tilde{r}_0, A^i\mathcal{P}_k(A)\mathcal{M}_k(A)r_0) = (\tilde{r}_0, A^i s_k),\\ \mu_i &= \mu(\xi^i\mathcal{M}_k^2) = \pi(\xi^{i+1}\mathcal{M}_k^2) = (\tilde{r}_0, A^{i+1}\mathcal{M}_k^2(A)r_0) = (\tilde{r}_0, A^{i+1} z_k).\end{aligned}$$

■

3.6 Nové rekurentní vztahy pro metodu CGS

Definujeme-li $\delta_k = 1$ pro každé k, dostáváme na základě rekurentních vztahů (3.72) až (3.75) novou interpretaci CGS.

3.6.1 Odvození rekurentních vztahů

Lemma 3.24 : Je-li $k = 0$ je $\mathcal{T}_0(\xi) \equiv 0$, $\mathcal{V}_0(\xi) \equiv 0$, $\mathcal{W}_0(\xi) = w_0$, $\mathcal{Q}_0(\xi) = \xi + q_0$ a platí

$$w_0 = \frac{\pi_0}{\mu_0}, \quad q_0 = -\frac{\mu_1}{\mu_0}, \tag{3.80}$$

$$s_1 = q_0 s_0 + As_0 - q_0 w_0 A z_0 - w_0 A^2 z_0, \tag{3.81}$$
$$r_1 = r_0 - 2w_0 A s_0 + w_0^2 A^2 z_0, \tag{3.82}$$
$$x_1 = x_0 + 2w_0 s_0 - w_0^2 A z_0, \tag{3.83}$$
$$z_1 = q_0^2 z_0 + 2q_0 A z_0 + A^2 z_0, \tag{3.84}$$

kde $s_0 = z_0 = r_0$.

Důkaz : Je-li $k = 0$, můžeme dle lemmatu 3.11 uvažovat polynomy $\mathcal{V}_0, \mathcal{T}_0, \mathcal{W}_0, \mathcal{Q}_0$ ve tvaru

$$\mathcal{V}_0 \equiv 0, \quad \mathcal{T}_0 \equiv 0, \quad \mathcal{W}_0 = w_0, \quad \mathcal{Q}_0 = q_0 + \xi.$$

Pro koeficienty $\widetilde{w}_0$ a $\widetilde{q}_0$ dostáváme z (3.44) a z (3.65) rovnice

$$\pi_0 = g_{00}\widetilde{w}_0 \quad \text{a} \quad -g_{01} = g_{00}\widetilde{q}_0.$$

Definujeme-li $m_{-1}^{(0)} = 0$ dostáváme z (3.43) $g_{00} = \mu_0$, $g_{01} = \mu_0 m_{-1}^{(0)} + \mu_1 = \mu_1$, z (3.15) máme $w_0 = \widetilde{w}_0$ a z (3.36) $q_0 = \widetilde{q}_0$. Celkem je tedy

$$w_0 = \frac{\pi_0}{\mu_0}, \quad q_0 = -\frac{g_{01}}{g_{00}} = -\frac{\mu_1}{\mu_0}.$$

Vztahy (3.81) - (3.84) dokážeme dosazením za polynomy $\mathcal{V}_0, \mathcal{T}_0, \mathcal{W}_0, \mathcal{Q}_0$ do vztahů (3.72) - (3.75) . Dokažme si např. (3.81) . Platí

$$\begin{aligned} s_1 &= (\mathcal{Q}_0 - A\mathcal{Q}_0\mathcal{V}_0 - A\mathcal{T}_0\mathcal{W}_0)s_0 - A\mathcal{Q}_0\mathcal{W}_0 z_0 + \mathcal{T}_0(I - A\mathcal{V}_0)r_0 = \mathcal{Q}_0 s_0 - A\mathcal{Q}_0\mathcal{W}_0 z_0 \\ &= (A + q_0 I)s_0 - A(A + q_0 I)(w_0 I)z_0 = q_0 s_0 + A\ s_0 - q_0 w_0 z_0 - w_0 A^2 z_0. \end{aligned}$$

To, že $s_0 = z_0 = r_0$ dokážeme přímo z definice s_0 a z_0 uvědomíme-li si, že $\mathcal{P}_0(A) = I$ a $\mathcal{M}_0(A) = I$. ■

Lemma 3.25 : Je-li $k \neq 0$ je $\mathcal{V}_k(\xi) \equiv 0$, $\mathcal{T}_k(\xi) = t_0$, $\mathcal{W}_k(\xi) = w_0$, $\mathcal{Q}_k(\xi) = \xi + q_0$ a platí

(3.85) $$w_0 = \frac{\pi_0}{\mu_0}, \quad q_0 = \frac{\pi_1}{\pi_0} - \frac{\mu_1}{\mu_0}, \quad t_0 = -\frac{\mu_0}{\pi_0},$$

$$\begin{aligned} &(3.86) & s_{k+1} &= q_0 s_k + 2As_k - q_0 w_0 A z_k - w_0 A^2 z_k + t_0 r_k, \\ &(3.87) & r_{k+1} &= r_k - 2w_0 A s_k + w_0^2 A^2 z_k, \\ &(3.88) & x_{k+1} &= x_k + 2w_0 s_k - w_0^2 A z_k, \\ &(3.89) & z_{k+1} &= q_0^2 z_k + 2q_0 A z_k + A^2 z_k + 2q_0 t_0 s_k + 2t_0 A s_k + t_0^2 r_k. \end{aligned}$$

Důkaz : Je-li $k \neq 0$, můžeme dle lemmatu 3.11 uvažovat polynomy $\mathcal{V}_k, \mathcal{T}_k, \mathcal{W}_k, \mathcal{Q}_k$ ve tvaru

$$\mathcal{V}_k \equiv 0, \quad \mathcal{T}_k = t_0, \quad \mathcal{W}_k = w_0, \quad \mathcal{Q}_k = q_0 + \xi.$$

Pro koeficienty $\widetilde{w}_0$, $\widetilde{q}_0$ a $\widetilde{t}_0$ dostáváme z (3.44), (3.63) a z (3.62) rovnice

(3.90) $$\pi_0 = g_{00}\widetilde{w}_0, \quad -g_{01} - f_{00}\widetilde{t}_0 = g_{00}\widetilde{q}_0, \quad \widetilde{t}_0 = -\frac{\mu_0}{\pi_0}.$$

Zároveň máme z (3.32), (3.53) a z (3.55)

$$w_0 = \widetilde{w}_0, \quad q_0 = \widetilde{q}_0 + m_{k-1}^{(k)}, \quad t_0 = \widetilde{t}_0.$$

Z (3.43) plyne, že

$$g_{00} = \mu_0, \quad g_{01} = \mu_1 + \mu_0 m_{k-1}^{(k)}, \quad f_{00} = \pi_1.$$

Z (3.90) tedy dostáváme

$$w_0 = \frac{\pi_0}{g_{00}} = \frac{\pi_0}{\mu_0}, \quad t_0 = -\frac{\mu_0}{\pi_0},$$

$$q_0 = -\frac{g_{01} + f_{00}\widetilde{t}_0}{g_{00}} + m_{k-1}^{(k)} = -\frac{\mu_1 + \mu_0 m_{k-1}^{(k)} - \pi_1 \frac{\mu_0}{\pi_0}}{\mu_0} + m_{k-1}^{(k)} = \frac{\pi_1}{\pi_0} - \frac{\mu_1}{\mu_0}.$$

■

3.6.2 Algoritmus CGS s novými rekurentními vztahy

Nyní již můžeme zformulovat algoritmus metody CGS s novými rekurentními vztahy. Nazvěme tuto metodu CGS_1.

Algoritmus 3.2

Input $x_0,\ A,\ b,\ \widetilde{r}_0,\ \varepsilon$;
$r_0 = b - Ax_0$;
if $\| r_0 \| < \varepsilon$ **then** $x^* = x_0$ *and exit* ;
$z_0 = s_0 = r_0$;
$\pi_0 = (\widetilde{r}_0, s_0)$; $\mu_0 = (\widetilde{r}_0, Az_0)$; $\mu_1 = (\widetilde{r}_0, A^2 z_0)$;
$w_0 = \frac{\pi_0}{\mu_0}$; $q_0 = -\frac{\mu_1}{\mu_0}$;
$s_1 = q_0 s_0 + As_0 - q_0 w_0 Az_0 - w_0 A^2 z_0$;
$r_1 = r_0 - 2w_0 As_0 + w_0^2 A^2 z_0$;
$x_1 = x_0 + 2w_0 s_0 - w_0^2 Az_0$;
$z_1 = q_0^2 z_0 + 2q_0 Az_0 + A^2 z_0$;
for $k = 1$ **to** $n-1$ **do**
  if $\| r_k \| < \varepsilon$ **then** $x^* = x_k$ *and exit*
  else
    begin
      $\pi_0 = (\widetilde{r}_0, s_k)$; $\pi_1 = (\widetilde{r}_0, As_k)$; $\mu_0 = (\widetilde{r}_0, Az_k)$; $\mu_1 = (\widetilde{r}_0, A^2 z_k)$;
      $w_0 = \frac{\pi_0}{\mu_0}$; $q_0 = \frac{\pi_1}{\pi_0} - \frac{\mu_1}{\mu_0}$; $t_0 = -\frac{\mu_0}{\pi_0}$;
      $s_{k+1} = q_0 s_k + 2As_k - q_0 w_0 Az_k - w_0 A^2 z_k + t_0 r_k$;
      $r_{k+1} = r_k - 2w_0 As_k + w_0^2 A^2 z_k$;
      $x_{k+1} = x_k + 2w_0 s_k - w_0^2 Az_k$;
      $z_{k+1} = q_0^2 z_k + 2q_0 Az_k + A^2 z_k + 2q_0 t_0 s_k + 2t_0 As_k + t_0^2 r_k$;
    end;
$x^* = x_n$.

Poznámka : Porovnáváme-li metodu CGS s metodou CGS_1 dle počtu operací zjistíme, že CGS používá dvakrát násobení matice-vektor, zatímco CGS_1 třikrát. Metoda CGS_1 prohrává i co do počtu násobení skalár-vektor,vektor-vektor(skalární součin) a co do počtu sčítání vektorů. Výhodou metody CGS_1 je to, že počítání vektorů s_{k+1}, r_{k+1}, x_{k+1} a z_{k+1} je na sobě nezávislé. Zřejmě je tedy tato metoda vhodná pro počítač s více procesory, protože již zmíněné vektory je možné počítat paralelně.

3.7 Určení délky skoku

Nyní již umíme konstruovat regulární polynomy, máme-li dánu délku skoku δ_k. V tomto odstavci se budeme zabývat otázkou, jak volit δ_k, aby polynom $\mathcal{P}_{n_k+\delta_k}$ byl regulární a aby nedošlo ke zhroucení našeho algoritmu.

3.7.1 Skutečné zhroucení

Připomeňme, že skutečné zhroucení nastává tehdy, je-li matice H_i singulární. Jak plyne z odstavce 3.2, na zjištění singularity matic H_{n_k+i} $(i \geq 1)$ stačí testovat hodnoty

$$\mu(\mathcal{M}_{n_k}\xi^{n_k+i-1}) = \mu(\mathcal{M}_{n_k}^2\xi^{i-1}) = \mu_{i-1}.$$

Při praktických výpočtech dochází k tomu, že matice nemusí být singulární, ale je téměř nemožné sestrojit regulární polynomy kvůli zaokrouhlovacím chybám, které mají značný vliv na algoritmus. To nastává v tom případě, jsou-li matice H_{n_k+i} téměř singulární neboli $\mu_{i-1} \approx 0$. Je-li $\mu_{i-1} \approx 0$ říkáme, že dochází k takzvanému skutečnému skoro zhroucení LOP. CGS bez skutečného skoro zhroucení získáme tak, že volíme délku skoku podle pravidla

- $\mu(\mathcal{M}_{n_k}\xi^i) < \varepsilon_1$ pro $i = 0, \dots, n_k + \delta_k - 2$
- $\mu(\mathcal{M}_{n_k}\xi^{n_k+\delta_k-1}) \geq \varepsilon_1$

kde ε_1 je vhodně zvolené číslo.

3.7.2 Lanczosovo zhroucení

Připomeňme, že Lanczosovo zhroucení je způsobeno volbou rekurentních vztahů. V našem případě je to tehdy, je-li $|\mu_{\delta_k-1}| > \varepsilon_1$, tedy regulární polynomy $\mathcal{P}_{k+1}$ a $\mathcal{M}_{k+1}$ existují, ale není možné je naším algoritmem spočítat, protože je $B^{(k)}$ nebo $C^{(k)}$ singulární. (Matice soustavy na počítání koeficientů $\tilde{w}_i^{(k)}$, $\tilde{v}_i^{(k)}$, $\tilde{q}_i^{(k)}$, $\tilde{t}_i^{(k)}$.)

Předpokládejme nyní, že platí $\mu_0 = 0, \dots, \mu_{\delta_k-2} = 0$, $|\mu_{\delta_k-1}| > \varepsilon_1$. Označme $x = \delta_k - 1$ a $y = \delta_k - 2$. Potom má např. matice $B^{(k)}$ tvar

$$\left(\begin{array}{cccc|ccc}
0 & \dots & 0 & \mu_x & f_{00} & \dots & f_{0y} \\
\vdots & 0 & \mu_x & g_{1x} & f_{10} & \dots & f_{1y} \\
0 & \mu_x & g_{x1} & \vdots & \vdots & & \vdots \\
\mu_x & g_{x1} & \dots & g_{xx} & f_{x0} & \dots & f_{xy} \\
\hline
0 & 0 & \dots & 0 & \pi_0 & \dots & \pi_y \\
\vdots & \ddots & \ddots & \vdots & 0 & \ddots & \vdots \\
0 & \dots & 0 & 0 & 0 & 0 & \pi_0
\end{array}\right). \tag{3.91}$$

Vidíme, že je-li $\pi_0 = 0$, je matice (3.91) singulární a dochází k Lanczosovu zhroucení naší metody. Podobně je-li $\mu_0 \approx 0, \dots, \mu_{\delta_k-2} \approx 0$, $|\mu_{\delta_k-1}| > \varepsilon_1$ a $\pi_0 \approx 0$, může dojít k Lanczosovu skoro-zhroucení naší metody.

Poznámka : $\pi_0 \approx 0$ nastane též tehdy, je-li $r_{k+1} \approx 0$, neboli $x_{k+1} \approx x^*$.

Definice 3.10 : Definujme číslo $\sigma_k = \mu(\mathcal{P}_k\mathcal{M}_k) \quad (\sigma_k = \pi_0)$.

Poznámka : Krok δ_k budeme volit tak, aby

1. Polynomy $\mathcal{P}_{k+1}$ a $\mathcal{M}_{k+1}$ byly regulární.
2. Výpočet $\widetilde{w}^{(k)}$, $\widetilde{v}^{(k)}$, $\widetilde{q}^{(k)}$, $\widetilde{t}^{(k)}$ byl stabilní (při gaussově eliminaci je $|pivot| > \varepsilon_2$).
3. $|\sigma_{k+1}| > \varepsilon_1$ (snažíme se zabránit Lanczosovu skoro-zhroucení v příštím kroku metody). Zároveň je-li $|\sigma_{k+1}| \leq \varepsilon_1$, testujeme $\| r_{k+1} \|$. Pokud je $\| r_{k+1} \| < \varepsilon_3$, položíme $x^* = x_{k+1}$ a ukončíme algoritmus.

Lemma 3.26 : Číslo σ_k můžeme vyjádřit ve tvaru

$$\sigma_k = (\widetilde{r}_0, s_k). \tag{3.92}$$

Důkaz : Platí

$$\sigma_k = \pi(\mathcal{P}_k\mathcal{M}_k) = (\widetilde{r}_0, \mathcal{P}_k(A)\mathcal{M}_k(A)r_0) = (\widetilde{r}_0, s_k). \qquad \blacksquare$$

Poznámka : Protože počítání polynomů $\mathcal{V}_k, \mathcal{W}_k, \mathcal{Q}_k, \mathcal{T}_k$ je paměťově náročné, musí být délka skoku omezena konstantou MJ. Dojde-li k tomu, že délka skoku je větší než tato konstanta, nastavíme proměnnou Err (Indikace chyby) na hodnotu 1 a ukončíme algoritmus.

Algoritmus 3.3

```
Input x_0, A, b, r~_0, ε_1, ε_2, ε_3;
r_0 = b − Ax_0;
s_0 = r_0; z_0 = r_0; Err = 0; MJ = Const;
for k = 0,1,2,... until convergence do
  if || r_k ||< ε_3 then x* = x_k and exit
  else
    begin
      δ_k = 1;
      Jump = true;
      while |μ_{δ_k −1}| < ε_1 do
        if δ_k < MJ then
          δ_k = δ_k + 1
        else  Err = 1 and exit;
      while Jump = true do
        begin
          Jump = false;
          Compute → W_k, V_k, Q_k, T_k;
          if nearly singular system (ε_2) then
            begin
              δ_k = δ_k + 1;
              Jump = true;
            end
          else
            begin
              s_{k+1} = (Q_k − AQ_kV_k − AT_kW_k)s_k − AQ_kW_kz_k + T_k(I − AV_k)r_k;
              r_{k+1} = (I − AV_k)^2 r_k − 2(I − AV_k)AW_ks_k + A^2 W_k^2 z_k;
              σ_{k+1} = (r~_0, s_{k+1});
              if |σ_{k+1}| < ε_1 then
                if || r_{k+1} ||≥ ε_3 then
                  begin
                    δ_k = δ_k + 1;
                    Jump = true;
                  end
            end
        end
      x_{k+1} = x_k − (AV_k − 2I)V_kr_k + 2(I − AV_k)W_ks_k − A W_k^2 z_k;
      z_{k+1} = Q_k^2 z_k + 2Q_kT_ks_k + T_k^2 r_k;
    end.
```

Definice 3.11 : Metodu definovanou algoritmem 3.3 nazveme metodou BSMRZS.

Poznámka : Porovnáme-li počet operací metody BSMRZS s metodami BCG a CGS vidíme, že se s nimi metoda BSMRZS nebude moci měřit co do rychlosti výpočtu řešení. Metoda BSMRZS je však zajímavým pokusem, jak zestabilnit a zpřesnit výpočet.

Kapitola 4

Zobecněná metoda sdružených gradientů (CG)

Naším cílem v této kapitole je definovat jistou třídu metod a ukázat, že metoda BCG náleží to této třídy metod.

4.1 Lanczosova metoda

V roce 1950 uvedl Lanczos metodu na postupnou transformaci danné matice na tří-diagonální tvar. Účelem bylo postupné počítání vlastních čísel vhodnou procedurou. Tento algoritmus byl přeformulován Lanczosem samotným a později též Fletcherem pro řešení systému lineárních rovnic. Aplikujeme-li nesymetrický Lanczosův proces na Krylovovy prostory generované maticí A a vektorem r_0 a maticí A^T a vektorem $\widetilde{r}_0$, potom algoritmus vytváří dvě vektorové posloupnosti w_k a v_k, přičemž

$$sp\{v_0, \ldots, v_k\} = sp\{r_0, Ar_0, \ldots, A^k r_0\}, \tag{4.1}$$

$$sp\{w_0, \ldots, w_k\} = sp\{\widetilde{r}_0, A^T\widetilde{r}_0, \ldots, (A^T)^k\widetilde{r}_0\} \tag{4.2}$$

a

$$w_i^T v_j = \begin{cases} 0 & \text{pro } i \neq j, \\ d_i & \text{pro } i = j, \end{cases} \tag{4.3}$$

pro $i, j = 0, \ldots, k$. Pokud startujeme s $v_0 = r_0/\eta_0$, $w_0 = \widetilde{r}_0/\mu_0$, mohou být vektory počítány pomocí tří-krokových rekurencí pro $k \geq 2$ jako

$$v_k = \frac{1}{\eta_k}(Av_{k-1} - \alpha_k v_{k-1} - \beta_k v_{k-2}), \tag{4.4}$$

$$w_k = \frac{1}{\mu_k}(A^T w_{k-1} - \alpha_k w_{k-1} - \gamma_k w_{k-2}) \tag{4.5}$$

a pro $k = 1$ použijeme stejné formule pouze bez výrazů s β_1 a γ_1. Koeficienty jsou určeny z (4.3) jako

$$\alpha_k = \frac{w_{k-1}^T A v_{k-1}}{w_{k-1}^T v_{k-1}}, \quad \beta_k = \frac{w_{k-2}^T A v_{k-1}}{w_{k-2}^T v_{k-2}}, \quad \gamma_k = \frac{v_{k-2}^T A^T w_{k-1}}{v_{k-2}^T w_{k-2}}. \tag{4.6}$$

η_k a μ_k jsou vhodně zvolené koeficienty. (Obvykle normalizují vektory nebo interpretují BCG.)

Definujme dále matice V a W jako

$$V = (v_0, \ldots, v_{n-1}) \in L(\mathbb{R}^n), \quad W = (w_0, \ldots, w_{n-1}) \in L(\mathbb{R}^n) \tag{4.7}$$

a matice H a $\widetilde{H}$ jako

$$H = \begin{pmatrix} \alpha_1 & \beta_2 & 0 & \ldots & 0 \\ \eta_1 & \alpha_2 & \beta_3 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \eta_{n-2} & \alpha_{n-1} & \beta_n \\ 0 & \ldots & 0 & \eta_{n-1} & \alpha_n \end{pmatrix}, \quad \widetilde{H} = \begin{pmatrix} \alpha_1 & \gamma_2 & 0 & \ldots & 0 \\ \mu_1 & \alpha_2 & \gamma_3 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \mu_{n-2} & \alpha_{n-1} & \gamma_n \\ 0 & \ldots & 0 & \mu_{n-1} & \alpha_n \end{pmatrix}. \tag{4.8}$$

Ze vztahů (4.4) a (4.5) a z toho, že $v_n = 0$ a $w_n = 0$ plyne, že platí

$$AV = VH, \tag{4.9}$$

$$A^T W = W\widetilde{H} \tag{4.10}$$

a z (4.3) máme

$$W^T V = D, \tag{4.11}$$

kde $D = diag(d_0, \ldots, d_{n-1})$.

4.2 Lanczosova metoda jako interpretace BCG

Označme r_k^{BCG} a $\tilde{r}_k^{BCG}$ reziduové vektory generované metodou BCG s počáteční volbou $r_0^{BCG} = r_0$ a $\tilde{r}_0^{BCG} = \tilde{r}_0$. Pro metodu BCG jakožto interpretaci Lanczosova ortogonalizačního procesu platí

$$\begin{aligned} r_k^{BCG} &\in r_0 + A \cdot sp\{r_0, Ar_0, \ldots, A^{k-1}r_0\}, \\ \tilde{r}_k^{BCG} &\in \tilde{r}_0 + A^T \cdot sp\{\tilde{r}_0, A^T\tilde{r}_0, \ldots, (A^T)^{k-1}\tilde{r}_0\}. \end{aligned} \tag{4.12}$$

Předpokládejme, že v_{k-1} leží ve stejném prostoru jako r_{k-1}^{BCG} a v_{k-2} leží ve stejném prostoru jako r_{k-2}^{BCG}. Z (4.12) plyne, že chceme-li, aby v_k ležel ve stejném prostoru jako r_k^{BCG}, musí být součet koeficientů v (4.4) u v_{k-1} a v_{k-2} roven jedné (součet koeficientů u vektoru r_0 musí být roven jedné). Dodefinujeme-li $\beta_1 = \gamma_1 = 0$, můžeme tuto podmínku pro $k \geq 1$ psát ve tvaru

$$\frac{-\alpha_k - \beta_k}{\eta_k} = 1 \Rightarrow \eta_k = -\alpha_k - \beta_k. \tag{4.13}$$

Podobnou úvahou pro w_k dospějeme k podmínce

$$\frac{-\alpha_k - \gamma_k}{\mu_k} = 1 \Rightarrow \mu_k = -\alpha_k - \gamma_k. \tag{4.14}$$

Definujeme-li navíc $\mu_0 = \eta_0 = 1$, bude algoritmus Lanczosovy metody generovat takové vektory v_k a w_k, pro které platí

- $v_0 = r_0$, $w_0 = \tilde{r}_0$,

- $v_k \in r_0 + A \ \cdot sp\{r_0, Ar_0, A^{k-1}r_0\}$,
- $w_k \in \widetilde{r}_0 + A^T \cdot sp\{\widetilde{r}_0, A^T\widetilde{r}_0, (A^T)^{k-1}\widetilde{r}_0\}$,
- $w_i^T v_j = 0$ pro $i \neq j$.

Jak jsme si ukázali v kapitole 2, vektory v_k a w_k jsou těmito podmínkami (viz. definice 2.1 a 2.2) určeny jednoznačně a platí tedy $v_k = r_k^{BCG}$ a $w_k = \widetilde{r}_k^{BCG}$. Volíme-li tedy koeficienty μ_k a η_k pro $k \geq 1$ podle (4.13) a (4.14) a $\mu_0 = \eta_0 = 1$, generuje Lanczosova metoda stejné vektory jako metoda BCG. Co dále platí při této volbě koeficientů říká následující lemma.

Lemma 4.1 : Nechť jsou koeficienty μ_k a η_k v Lanczosově metodě voleny podle (4.13) a (4.14) pro $k \geq 1$ a nechť $\mu_0 = \eta_0 = 1$. Potom pro $2 \leq k \ \leq n$ platí

$$\beta_k = \gamma_k. \tag{4.15}$$

Důkaz : Indukcí.

- Pro $k = 2$ dostáváme

$$\begin{aligned} v_1 &= -\frac{1}{\alpha_1}Av_0 + v_0, \quad w_1 = -\frac{1}{\alpha_1}A^T w_0 + w_0, \\ w_0^T A v_1 &= w_0^T A(-\frac{1}{\alpha_1}Av_0 + v_0) = -\frac{1}{\alpha_1}w_0^T A^2 v_0 + w_0^T A v_0 = \\ &= -\frac{1}{\alpha_1}v_0^T (A^T)^2 w_0 + v_0^T A^T w_0 = v_0^T A^T(-\frac{1}{\alpha_1}A^T w_0 + w_0) = v_0^T A^T w_1, \end{aligned}$$

  neboli $\beta_2 = \gamma_2$.

- Nechť tvrzení platí pro $i = 2, \ldots, k-1$. Dokážeme, že platí i pro k. Platí

$$w_{k-2}^T A v_{k-1} = -\frac{1}{\alpha_{k-1} + \beta_{k-1}}(w_{k-2}^T A(Av_{k-2} - \alpha_{k-1}v_{k-2} - \beta_{k-1}v_{k-3})) =$$

$$= -\frac{1}{\alpha_{k-1} + \beta_{k-1}}(w_{k-2}^T A^2 v_{k-2} - \alpha_{k-1}w_{k-2}^T A v_{k-2} - \beta_{k-1}w_{k-2}^T A v_{k-3}).$$

  Využijeme-li nyní indukčního předpokladu a sice $\beta_{k-1} = \gamma_{k-1}$, dostáváme $w_{k-3}^T A v_{k-2} = v_{k-3}^T A^T w_{k-2}$ neboli $v_{k-2}^T A^T w_{k-3} = w_{k-2}^T A v_{k-3}$, můžeme $w_{k-2}^T A v_{k-1}$ psát ve tvaru

$$w_{k-2}^T A v_{k-1} = -\frac{1}{\alpha_{k-1} + \beta_{k-1}}(v_{k-2}^T {A^T}^2 w_{k-2} - \alpha_{k-1}v_{k-2}^T A^T w_{k-2} - \beta_{k-1}v_{k-2}^T A^T w_{k-3}).$$

  Protože je dle indukčního předpokladu $\beta_{k-1} = \gamma_{k-1}$, můžeme psát

$$v_{k-2}^T A^T w_{k-1} = -\frac{1}{\alpha_{k-1} + \beta_{k-1}}(v_{k-2}^T A^T(A^T w_{k-2} - \alpha_{k-1}w_{k-2} - \beta_{k-1}w_{k-3})) =$$

$$= -\frac{1}{\alpha_{k-1} + \beta_{k-1}}(v_{k-2}^T {A^T}^2 w_{k-2} - \alpha_{k-1}v_{k-2}^T A^T w_{k-2} - \beta_{k-1}v_{k-2}^T A^T w_{k-3})$$

  a odtud $v_{k-2}^T A^T w_{k-1} = w_{k-2}^T A v_{k-1}$ neboli $\beta_k = \gamma_k$.

$\blacksquare$

Nyní můžeme zformulovat algoritmus BCG, tentokrát v Lanczosově podobě.

Algoritmus 4.1

Input $x_0,\ A,\ b,\ \widetilde{r}_0,\ \varepsilon_0$;
$r_0 = b - Ax_0$;
$\alpha_1 = \frac{\widetilde{r}_0^T A r_0}{\widetilde{r}_0^T r_0}$;
$r_1 = -\frac{1}{\alpha_1} A r_0 + r_0$; $\widetilde{r}_1 = -\frac{1}{\alpha_1} A^T \widetilde{r}_0 + \widetilde{r}_0$; $x_1 = \frac{1}{\alpha_1} r_0 + x_0$;
for $k = 2$ **to** n **do**
  if $\| r_{k-1} \| < \varepsilon_0$ **then** $x^* = x_{k-1}$ *and exit*
  else
    begin
      $\alpha_k = \frac{\widetilde{r}_{k-1}^T A r_{k-1}}{\widetilde{r}_{k-1}^T r_{k-1}}$;
      $\beta_k = \frac{\widetilde{r}_{k-2}^T A r_{k-1}}{\widetilde{r}_{k-2}^T r_{k-2}}$;
      $r_k = -\frac{1}{\alpha_k + \beta_k}(A r_{k-1} - \alpha_k r_{k-1} - \beta_k r_{k-2})$;
      $\widetilde{r}_k = -\frac{1}{\alpha_k + \beta_k}(A^T \widetilde{r}_{k-1} - \alpha_k \widetilde{r}_{k-1} - \beta_k \widetilde{r}_{k-2})$;
      $x_k = \frac{1}{\alpha_k + \beta_k}(r_{k-1} + \alpha_k x_{k-1} + \beta_k x_{k-2})$;
    end;
$x^* = x_n$.

4.3 Zobecněná metoda sdružených gradientů

V tomto odstavci definujeme metodu, která určuje jistou třídu metod na řešení systému lineárních rovnic. V dalším ukážeme, že metoda BCG leží v této třídě metod.

Definice 4.1 : Definujme k-tou aproximaci řešení soustavy (2.1) x_k následujícími podmínkami

1. $x_k \in x_0 + \mathcal{K}_k(A, r_0) = x_0 + sp\{r_0, Ar_0, \ldots, A^{k-1} r_0\}$,
2. $r_k = b - Ax_k \Rightarrow r_k \in r_0 + A\ \mathcal{K}_k(A, r_0)$,
3. $r_k^T Z r_{k-i} = 0$ pro $i = 1, \ldots, \sigma_k$,

kde Z je libovolná regulární matice, σ_k číslo. Metodu definovanou podmínkami 1-3 nazveme zobecněnou metodou sdružených gradientů (CG).

Definice 4.2 : Metodu CG nazveme

- *Přesnou* je-li $\sigma_k = k$,
- *Znovustartovací* je-li $\sigma_k = (k-1)\ mod\ \sigma_{res} + 1$, kde σ_{res} je pevně zvolené číslo,
- *Zkracující* je-li $\sigma_k = \min(k, \sigma_{max})$, kde σ_{max} je pevně zvolené číslo,
- *Kombinovanou* je-li σ_k je voleno kombinovaně dle předchozích dvou metod jako $\sigma_k = \min((k-1)\ mod\ \sigma_{res} + 1, \sigma_{max})$.

4.4 Tříkroková rekurence pro přesnou CG

Naším cílem v tomto odstavci je určit podmínky, za jakých lze vektory generované přesnou CG počítat tříkrokovou rekurencí a zformulovat tuto tříkrokovou rekurenci.

Lemma 4.2 : Předpokládejme, že $r_i^T Z r_i \neq 0$ pro $i = 0, \ldots, k$. Potom platí

$$sp\{r_0, \ldots, r_k\} = \mathcal{K}_{k+1}(A, r_0). \tag{4.16}$$

Důkaz : Z definice 4.1 plyne, že $sp\{r_0, \ldots, r_k\} \subseteq \mathcal{K}_{k+1}(A, r_0)$. K důkazu opačné inkluze stačí dokázat lineární nezávislost reziduí. Nechť tedy platí

$$0 = \sum_{i=0}^{k} \gamma_i r_i \quad \Rightarrow \quad 0 = \sum_{i=0}^{k} \gamma_i r_j^T Z r_i = \gamma_j r_j^T Z r_j \quad \text{pro} \quad j = 0, \ldots, k.$$

Protože je $r_j^T Z r_j \neq 0$ pro $j = 0, \ldots, k$, je $\gamma_j = 0$ pro $j = 0, \ldots, k$.

■

Lemma 4.3 : Je-li $r_i^T Z r_i \neq 0$ pro $i = 0, \ldots, k$, lze psát vektory r_k ve tvaru

$$r_k = \beta_0^{(k)} A \; r_{k-1} + \sum_{i=1}^{k} \beta_i^{(k)} r_{k-i}, \text{ kde } \sum_{i=1}^{k} \beta_i^{(k)} = 1. \tag{4.17}$$

Důkaz : Podle podmínky 2. v definici 4.1 existují koeficienty $\nu_i^{(k)}$ $(i = 1, \ldots, k)$ takové, že

$$r_k = r_0 + A \cdot \sum_{i=1}^{k} \nu_i^{(k)} A^{i-1} r_0.$$

Podle lemmatu 4.2 existují koeficienty $\mu_i^{(k)}$ $(i = 1, \ldots, k)$ takové, že

$$\sum_{i=1}^{k} \nu_i^{(k)} A^{i-1} r_0 = \sum_{i=1}^{k} \mu_i^{(k)} r_{k-i} \quad \text{neboli} \quad r_k = r_0 + A \cdot \sum_{i=1}^{k} \mu_i^{(k)} r_{k-i}.$$

Jinak napsáno

$$r_k = \mu_1^{(k)} A r_{k-1} + (r_0 + A \cdot \sum_{i=2}^{k} \mu_i^{(k)} r_{k-i}).$$

Protože

$$r_0 + A \cdot \sum_{i=2}^{k} \mu_i^{(k)} r_{k-i} \in \mathcal{K}_k,$$

existují koeficienty $\beta_i^{(k)}$ $(i = 1, \ldots, k)$ takové, že

$$r_0 + A \cdot \sum_{i=2}^{k} \mu_i^{(k)} r_{k-i} = \sum_{i=1}^{k} \beta_i^{(k)} r_{k-i}.$$

Položíme-li $\beta_0^{(k)} = \mu_1^{(k)}$, můžeme r_k psát ve tvaru

$$r_k = \beta_0^{(k)} A \; r_{k-1} + \sum_{i=1}^{k} \beta_i^{(k)} r_{k-i}. \tag{4.18}$$

Podmínka $\sum_{i=1}^{k} \beta_i^{(k)} = 1$ plyne z toho, že $r_{k-i} \in r_0 + A\mathcal{K}_{k-i}(A, r_0)$ pro $i = 0, \ldots, k-1$, neboli součet koeficientů u vektoru r_0 v (4.18) musí být roven jedné. ■

Lemma 4.4 : Nechť $\beta_i^{(k)}$ pro $i = 1, \ldots, k$ jsou koeficienty definované v předchozím lemmatu. Je-li $\beta_0^{(k)} \neq 0$, pak platí

$$r_k^T Z r_{k-j} = 0 \Leftrightarrow (A r_{k-1} - \sum_{i=1}^{k} \alpha_i^{(k)} r_{k-i})^T Z r_{k-j} = 0 \quad \text{pro} \quad j = 1, \ldots, k, \tag{4.19}$$

kde $\alpha_i^{(k)} = -\frac{\beta_i^{(k)}}{\beta_0^{(k)}}$.

Důkaz : Plyne z předchozího lemmatu. ■

Poznámka : Rovnice na pravé straně ekvivalence (4.19) lze psát jako

$$\begin{pmatrix} r_{k-1}^T A^T Z r_{k-1} \\ r_{k-1}^T A^T Z r_{k-2} \\ \vdots \\ r_{k-1}^T A^T Z r_0 \end{pmatrix} = \begin{pmatrix} r_{k-1}^T Z r_{k-1} & r_{k-2}^T Z r_{k-1} & \ldots & r_0^T Z r_{k-1} \\ r_{k-1}^T Z r_{k-2} & r_{k-2}^T Z r_{k-2} & \ldots & r_0^T Z r_{k-2} \\ \vdots & \vdots & \ddots & \vdots \\ r_{k-1}^T Z r_0 & r_{k-2}^T Z r_0 & \ldots & r_0^T Z r_0 \end{pmatrix} \begin{pmatrix} \alpha_1^{(k)} \\ \alpha_2^{(k)} \\ \vdots \\ \alpha_k^{(k)} \end{pmatrix}. \tag{4.20}$$

Lemma 4.5 : Nechť $r_i^T Z\ r_i \neq 0$ pro $i = 0, \ldots, k-1$. Je-li

$$r_{k-1}^T A^T Z r_{k-i} = 0 \quad \text{pro} \quad 3 \leq i \leq k\ , \tag{4.21}$$

potom

$$r_k = \beta_0^{(k)} (A r_{k-1} - \alpha_1^{(k)} r_{k-1} - \alpha_2^{(k)} r_{k-2}),$$

kde

$$\alpha_2^{(k)} = \frac{r_{k-2}^T Z^T A r_{k-1}}{r_{k-2}^T Z r_{k-2}}, \quad \alpha_1^{(k)} = \frac{r_{k-1}^T Z^T A r_{k-1}}{r_{k-1}^T Z r_{k-1}} - \alpha_2^{(k)} \frac{r_{k-2}^T Z r_{k-1}}{r_{k-1}^T Z r_{k-1}},$$

$$\beta_0^{(k)} = -\frac{1}{\alpha_1^{(k)} + \alpha_2^{(k)}}.$$

Důkaz : Využijeme-li ortogonálních podmínek, lemmatu 4.3 a podmínky (4.21) , můžeme (4.20) psát ve tvaru

$$\begin{pmatrix} r_{k-1}^T A^T Z r_{k-1} \\ r_{k-1}^T A^T Z r_{k-2} \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} r_{k-1}^T Z r_{k-1} & r_{k-2}^T Z r_{k-1} & \ldots & r_0^T Z r_{k-1} \\ 0 & r_{k-2}^T Z r_{k-2} & \ldots & r_0^T Z r_{k-2} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & r_0^T Z r_0 \end{pmatrix} \begin{pmatrix} \alpha_1^{(k)} \\ \alpha_2^{(k)} \\ \vdots \\ \alpha_k^{(k)} \end{pmatrix} \tag{4.22}$$

a odtud

$$\alpha_2^{(k)} = \frac{r_{k-2}^T Z^T A r_{k-1}}{r_{k-2}^T Z r_{k-2}}, \quad \alpha_1^{(k)} = \frac{r_{k-1}^T Z^T A r_{k-1}}{r_{k-1}^T Z r_{k-1}} - \alpha_2^{(k)} \frac{r_{k-2}^T Z r_{k-1}}{r_{k-1}^T Z r_{k-1}},$$

$$\alpha_i^{(k)} = 0 \quad \text{pro} \quad i = 3, \ldots, k.$$

$\beta_0^{(k)}$ spočteme z podmínky

$$1 = \sum_{i=1}^{k} \beta_i^{(k)} = \beta_0^{(k)} \sum_{i=1}^{k} \frac{\beta_i^{(k)}}{\beta_0^{(k)}} = \beta_0^{(k)}(-\sum_{i=1}^{k} \alpha_i^{(k)}) = \beta_0^{(k)}(-\alpha_1^{(k)} - \alpha_2^{(k)}).$$

Celkem tedy pro r_k dostáváme

$$r_k = \beta_0^{(k)}(Ar_{k-1} + \sum_{i=1}^{k} \frac{\beta_i^{(k)}}{\beta_i^{(k)}} r_{k-i}) = \beta_0^{(k)}(Ar_{k-1} - \alpha_1^{(k)} r_{k-1} - \alpha_2^{(k)} r_{k-2}).$$

■

Poznámka : Je-li matice Z symetrická, platí

$$\alpha_1^{(k)} = \frac{r_{k-1}^T Z^T A r_{k-1}}{r_{k-1}^T Z r_{k-1}}.$$

Lemma 4.6 : Nechť $r_i^T Z\ r_i \neq 0$ pro $i = 0, \ldots, k-1$ a nechť $A^T Z = ZA$. Potom platí

$$r_{k-1}^T A^T Z r_{k-i} = 0 \quad \text{pro} \quad 3 \leq i \leq k.$$

Důkaz : Protože $Ar_{k-i} \in \mathcal{K}_{k-i}(A, r_0)$ a protože dle definice vektoru r_{k-1} a dle lemmatu 4.2 platí $r_{k-1} \perp Z \cdot \mathcal{K}_{k-i}(A, r_0)$ pro $i = 3, \ldots, k$, platí též $r_{k-1}^T Z A r_{k-i} = 0$. Využijeme-li předpokladu $A^T Z = ZA$, dostáváme $r_{k-1}^T A^T Z r_{k-i} = 0$ pro $i = 3, \ldots, k$.

■

Poznámka : Podmínka (4.21) je tedy splněna například platí-li $A^T Z = ZA$.

Lemma 4.7 : Vektor x_k lze počítat jako

$$x_k = \frac{1}{\alpha_1^{(k)} + \alpha_1^{(k)}}(r_{k-1} + \alpha_1^{(k)} x_{k-1} + \alpha_2^{(k)} x_{k-2}).$$

Důkaz : Platí

$$\begin{aligned} b - Ax_k &= -\frac{1}{\alpha_1^{(k)} + \alpha_1^{(k)}}(Ar_{k-1} - \alpha_1^{(k)}(b - Ax_{k-1}) - \alpha_2^{(k)}(b - Ax_{k-2})) \\ -Ax_k &= -\frac{1}{\alpha_1^{(k)} + \alpha_1^{(k)}}(Ar_{k-1} + \alpha_1^{(k)} Ax_{k-1} + \alpha_2^{(k)} Ax_{k-2}) \\ x_k &= \frac{1}{\alpha_1^{(k)} + \alpha_1^{(k)}}(r_{k-1} + \alpha_1^{(k)} x_{k-1} + \alpha_2^{(k)} x_{k-2}). \end{aligned}$$

■

Důsledek : Platí $x_k - x_0 \in \mathcal{K}_k(A, r_0)$.

4.5 BCG - speciální případ CG

V tomto odstavci si ukážeme, že metoda BCG je speciálním případem přesné metody CG. Naším úkolem je tedy volit matici Z tak, aby přesná CG generovala stejné vektory r_k a x_k, jako metoda BCG. My zde ukážeme dva způsoby, jak provést tuto volbu.

4.5.1 Volba $Z = Z^{CG}$

Definice 4.3 : Nechť v tomto odstavci jsou $W \in L(\mathbb{R}^n)$, $V \in L(\mathbb{R}^n)$, $D \in L(\mathbb{R}^n)$, $H \in L(\mathbb{R}^n)$ a $\widetilde{H} \in L(\mathbb{R}^n)$ matice generované Lanczosovou metodou, která je interpretací metody BCG s volbou $v_0 = r_0$ a $w_0 = \widetilde{r}_0$.

Definice 4.4 : Definujme matici

$$Z^{CG} = WD^{-1}W^T.$$

Označme r_k^{CG} rezidua generovaná přesnou CG metodou s volbou $Z = Z^{CG}$ a $r_0^{CG} = r_0$ a definujme $\widetilde{r}_k^{CG} = Z^{CG} r_k^{CG}$. Dále označme $V_i = r_{i-1}^{BCG}$ i-tý sloupec matice V, D_i i-tý sloupec matice D a nechť $e_i = (0 \ldots, 0, 1, 0, \ldots, 0)^T$ značí i-tý jednotkový vektor. Nechť r_k^{BCG} a $\widetilde{r}_k^{BCG}$ značí reziduové vektory generované metodou BCG s volbou $r_0^{BCG} = r_0$ a $\widetilde{r}_0^{BCG} = \widetilde{r}_0$.

Lemma 4.8 : Matice Z^{CG} je symetrická a platí $A^T Z^{CG} = Z^{CG} A$.

Důkaz : Platí

$$Z^{CG} = WD^{-1}W^T = WD^{-T}W^T = {Z^{CG}}^T,$$

tedy Z^{CG} je symetrická. Zároveň dle lemmatu 4.1 platí $\gamma_k = \beta_k$, kde γ_k a β_k jsou koeficienty z Lanczosovy metody, neboli $H = \widetilde{H}$. Platí tedy

$$A^T Z^{CG} = A^T W D^{-1} W^T = W\widetilde{H}D^{-1}W^T = WHD^{-1}W^T = WHV^{-1}W^{-T}W^T =$$

$$= WV^{-1}AVV^{-1} = WV^{-1}A = WD^{-1}W^T A \; = Z^{CG}A. \qquad \blacksquare$$

Lemma 4.9 : Je-li $r_k^{BCG} = r_k^{CG}$, platí $\widetilde{r}_k^{BCG} = \widetilde{r}_k^{CG}$ pro $0 \le k \; \le n-1$.

Důkaz : Je-li $r_k^{BCG} = r_k^{CG}$, platí pro $\widetilde{r}_k^{CG}$

$$\widetilde{r}_k^{CG} = Z^{CG} r_k^{CG} = WD^{-1}W^T r_k^{BCG} = WD^{-1}W^T V_{k+1} = WD^{-1}D_{k+1} = We_{k+1} = \widetilde{r}_k^{BCG}.$$

$\blacksquare$

Lemma 4.10 : Platí $r_k^{CG} = r_k^{BCG}$ a $\widetilde{r}_k^{BCG} = \widetilde{r}_k^{CG}$ pro $0 \le k \; \le n-1$.

Důkaz : Indukcí. Protože platí $A^T Z^{CG} = Z^{CG} A$, můžeme metodu CG s s volbou $Z = Z^{CG}$ formulovat dle lemmatu 4.5 pomocí tříkrokové rekurence. Navíc ze symetrie Z^{CG} plyne, že

$$\alpha_1^{(k)} = \frac{r_{k-1}^T Z^T A r_{k-1}}{r_{k-1}^T Z r_{k-1}}.$$

- Je-li $k = 0$ platí dle definice 4.4 $r_0^{BCG} = r_0^{CG} = r_0$ a dle předchozího lemmatu je i $\widetilde{r}_0^{CG} = \widetilde{r}_0^{BCG} = \widetilde{r}_0$.

- Z algoritmu 4.1 dostáváme

  $$\alpha_1 = \frac{(\widetilde{r}_0^{BCG})^T A r_0^{BCG}}{(\widetilde{r}_0^{BCG})^T r_0^{BCG}} = \frac{(r_0^{CG})^T {Z^{CG}}^T A r_0^{CG}}{(r_0^{CG})^T {Z^{CG}}^T r_0^{CG}} = \frac{(r_0^{CG})^T {Z^{CG}}^T A r_0^{CG}}{(r_0^{CG})^T Z^{CG} r_0^{CG}} = \alpha_1^{(1)}.$$

  Platí

  $$r_1^{CG} = -\frac{1}{\alpha_1^{(1)}}(Ar_0^{CG} - \alpha_1^{(1)} r_0^{CG}) = -\frac{1}{\alpha_1} A r_0^{BCG} + r_0^{BCG} = r_1^{BCG}.$$

  Dle předchozího lemmatu platí tedy též $\widetilde{r}_1^{CG} = \widetilde{r}_1^{BCG}$.

- Pro $k = 2$ je

$$\alpha_2 = \frac{(\tilde{r}_1^{BCG})^T A r_1^{BCG}}{(\tilde{r}_1^{BCG})^T r_1^{BCG}} = \frac{(r_1^{CG})^T Z^{CG^T} A r_1^{CG}}{(r_1^{CG})^T Z^{CG^T} r_1^{CG}} = \frac{(r_1^{CG})^T Z^{CG^T} A r_1^{CG}}{(r_1^{CG})^T Z^{CG} r_1^{CG}} = \alpha_1^{(2)}.$$

a

$$\beta_2 = \frac{(\tilde{r}_0^{BCG})^T A r_1^{BCG}}{(\tilde{r}_0^{BCG})^T r_0^{BCG}} = \frac{(r_0^{CG})^T Z^{CG^T} A r_1^{CG}}{(r_0^{CG})^T Z^{CG^T} r_0^{CG}} = \frac{(r_0^{CG})^T Z^{CG^T} A r_1^{CG}}{(r_0^{CG})^T Z^{CG} r_0^{CG}} = \alpha_2^{(2)}.$$

Platí

$$\begin{aligned} r_2^{CG} &= -\frac{1}{\alpha_1^{(2)} + \alpha_2^{(2)}}(Ar_1^{CG} - \alpha_1^{(2)} r_1^{CG} - \alpha_2^{(2)} r_0^{CG}) = \\ &= -\frac{1}{\alpha_2 + \beta_2}(Ar_1^{BCG} - \alpha_2 r_1^{BCG} - \beta_2 r_0^{BCG}) = r_2^{BCG} \end{aligned}$$

a tedy i $\tilde{r}_2^{CG} = \tilde{r}_2^{BCG}$.

- Dokažme nyní indukční krok. Nechť tedy platí $r_i^{CG} = r_i^{BCG}$ a $\tilde{r}_i^{CG} = \tilde{r}_i^{BCG}$ pro $i = 0, \ldots, k-1$. Dokážeme, že $r_k^{CG} = r_k^{BCG}$ a tedy i $\tilde{r}_k^{CG} = \tilde{r}_k^{BCG}$. Z algoritmu 4.1 a z indukčního předpokladu dostáváme

$$\begin{aligned} \alpha_k &= \frac{(\tilde{r}_{k-1}^{BCG})^T A r_{k-1}^{BCG}}{(\tilde{r}_{k-1}^{BCG})^T r_{k-1}^{BCG}} = \frac{(r_{k-1}^{CG})^T Z^{CG^T} A r_{k-1}^{CG}}{(r_{k-1}^{CG})^T Z^{CG^T} r_{k-1}^{CG}} = \frac{(r_{k-1}^{CG})^T Z^{CG^T} A r_{k-1}^{CG}}{(r_{k-1}^{CG})^T Z^{CG} r_{k-1}^{CG}} = \alpha_1^{(k)}, \\ \beta_k &= \frac{(\tilde{r}_{k-2}^{BCG})^T A r_{k-1}^{BCG}}{(\tilde{r}_{k-2}^{BCG})^T r_{k-2}^{BCG}} = \frac{(r_{k-2}^{CG})^T Z^{CG^T} A r_{k-1}^{CG}}{(r_{k-2}^{CG})^T Z^{CG^T} r_{k-2}^{CG}} = \frac{(r_{k-2}^{CG})^T Z^{CG^T} A r_{k-1}^{CG}}{(r_{k-2}^{CG})^T Z^{CG} r_{k-2}^{CG}} = \alpha_2^{(k)}, \\ r_k^{CG} &= -\frac{1}{\alpha_1^{(k)} + \alpha_2^{(k)}}(Ar_{k-1}^{CG} - \alpha_1^{(k)} r_{k-1}^{CG} - \alpha_2^{(k)} r_{k-2}^{CG}) = \\ &= -\frac{1}{\alpha_k + \beta_k}(Ar_{k-1}^{BCG} - \alpha_k r_{k-1}^{BCG} - \beta_k r_{k-2}^{BCG}) = r_k^{BCG} \end{aligned}$$

a tedy i $\tilde{r}_k^{CG} = \tilde{r}_k^{BCG}$. ■

4.5.2 BCG jako aplikace CG na dvojsystém

Druhým způsobem, jak získat BCG z CG je položit

$$Z = Z^{BC} = \begin{pmatrix} O & I \\ I & O \end{pmatrix}, \quad \hat{A} = \begin{pmatrix} A & O \\ O & A^T \end{pmatrix}, \quad \hat{x} = \begin{pmatrix} x \\ \tilde{x} \end{pmatrix}, \quad \hat{b} = \begin{pmatrix} b \\ \tilde{b} \end{pmatrix},$$

přičemž položíme $\tilde{x}_0 = 0$, tedy $\tilde{b} = \tilde{r}_0$. Reziduum $\hat{r}_k$ definujeme jako

$$\hat{r}_k = \begin{pmatrix} b \\ \tilde{b} \end{pmatrix} - \begin{pmatrix} A & O \\ O & A^T \end{pmatrix}\begin{pmatrix} x_k \\ \tilde{x}_k \end{pmatrix} = \begin{pmatrix} b - Ax_k \\ \tilde{b} - A^T\tilde{x}_k \end{pmatrix} = \begin{pmatrix} r_k \\ \tilde{r}_k \end{pmatrix}.$$

Lemma 4.11 : Platí

$$\hat{A}^T Z^{BC} = Z^{BC} \hat{A}.$$

Důkaz : Platí
$$\widehat{A}^T Z^{BC} = \begin{pmatrix} A^T & O \\ O & A \end{pmatrix} \begin{pmatrix} O & I \\ I & O \end{pmatrix} = \begin{pmatrix} O & A^T \\ A & O \end{pmatrix} = \begin{pmatrix} O & I \\ I & O \end{pmatrix} \begin{pmatrix} A & O \\ O & A^T \end{pmatrix} = Z^{BC}\widehat{A}. \quad \blacksquare$$

Poznámka : S využitím lemmatu 4.5 můžeme $\widehat{r}_k$ počítat pomocí tříkrokové rekurence. Platí tedy

(4.23)
$$\widehat{r}_k = \beta_0^{(k)}(\widehat{A}\widehat{r}_{k-1} - \alpha_1^{(k)}\widehat{r}_{k-1} - \alpha_2^{(k)}\widehat{r}_{k-2}),$$

kde
$$\alpha_1^{(k)} = \frac{\widehat{r}_{k-1}^T Z^{BC}\widehat{A}\widehat{r}_{k-1}}{\widehat{r}_{k-1}^T Z^{BC}\widehat{r}_{k-1}}, \quad \alpha_2^{(k)} = \frac{\widehat{r}_{k-2}^T Z^{BC}\widehat{A}\widehat{r}_{k-1}}{\widehat{r}_{k-2}^T Z^{BC}\widehat{r}_{k-2}}, \quad \beta_0^{(k)} = -\frac{1}{\alpha_1^{(k)} + \alpha_2^{(k)}}.$$

Rozepíšeme-li (4.23) pro vektory r_k a $\widetilde{r}_k$ dostáváme
$$\begin{aligned} r_k &= \beta_0^{(k)}(Ar_{k-1} - \alpha_1^{(k)} r_{k-1} - \alpha_2^{(k)} r_{k-2}), \\ \widetilde{r}_k &= \beta_0^{(k)}(A^T\widetilde{r}_{k-1} - \alpha_1^{(k)}\widetilde{r}_{k-1} - \alpha_2^{(k)}\widetilde{r}_{k-2}). \end{aligned}$$

Vidíme, že koeficienty u příslušných vektorů jsou stejné, zřejmě tedy platí že existují $\gamma_i^{(k)}$ takové, že
$$\begin{aligned} r_k &= r_0 + \sum_{i=1}^{k} \gamma_i^{(k)} A^i r_0, \\ \widetilde{r}_k &= \widetilde{r}_0 + \sum_{i=1}^{k} \gamma_i^{(k)} (A^T)^i \widetilde{r}_0. \end{aligned}$$

Lemma 4.12 : Platí
$$r_i^T\widetilde{r}_j = \widetilde{r}_i^T r_j.$$

Důkaz : Rozepíšeme-li výraz $r_i^T\widetilde{r}_j$, dostáváme
$$r_i^T\widetilde{r}_j = (r_0 + \sum_{k=1}^{i}\gamma_k^{(i)}A^k r_0)^T(\widetilde{r}_0 + \sum_{l=1}^{j}\gamma_l^{(j)}(A^T)^l\widetilde{r}_0) = r_0^T\widetilde{r}_0 + \sum_{l=1}^{j}\gamma_l^{(j)} r_0^T (A^T)^l \widetilde{r}_0 +$$
$$+\sum_{k=1}^{i}\gamma_k^{(i)} r_0^T (A^T)^k\widetilde{r}_0 + r_0^T\sum_{k=1}^{i}\gamma_k^{(i)}(A^T)^k\sum_{l=1}^{j}\gamma_l^{(j)}(A^T)^l\widetilde{r}_0 = \widetilde{r}_0^T r_0 + \widetilde{r}_0^T\sum_{l=1}^{j}\gamma_l^{(j)}A^l r_0 +$$
$$+\widetilde{r}_0^T\sum_{k=1}^{i}\gamma_k^{(i)}A^k r_0 + \widetilde{r}_0^T\sum_{l=1}^{j}\gamma_l^{(j)}A^l\sum_{k=1}^{i}\gamma_k^{(i)}A^k r_0 = \widetilde{r}_0^T r_0 + \widetilde{r}_0^T\sum_{l=1}^{j}\gamma_l^{(j)}A^l r_0 + \widetilde{r}_0^T\sum_{k=1}^{i}\gamma_k^{(i)}A^k r_0 +$$
$$+\widetilde{r}_0^T\sum_{k=1}^{i}\gamma_k^{(i)}A^k\sum_{l=1}^{j}\gamma_l^{(j)}A^l r_0 = (\widetilde{r}_0 + \sum_{k=1}^{i}\gamma_k^{(i)}(A^T)^k\widetilde{r}_0)^T(r_0 + \sum_{l=1}^{j}\gamma_l^{(j)}A^l r_0) = \widetilde{r}_i^T r_j^T. \quad \blacksquare$$

Lemma 4.13 : Platí
$$\widetilde{r}_i^T r_j = 0 \quad \text{pro} \quad i \neq j.$$

Důkaz : Aplikujeme-li CG s maticí $Z = Z^{BC}$ na systém $\widehat{A}\widehat{x} = \widehat{b}$, platí pro vektory $\widehat{r}_i$ a $\widehat{r}_j$ $(i \neq j)$
$$0 = \widehat{r}_i^T Z^{BC}\widehat{r}_j = (\, r_i^T \quad \widetilde{r}_i^T\,)\begin{pmatrix} O & I \\ I & O \end{pmatrix}\begin{pmatrix} r_j \\ \widetilde{r}_j \end{pmatrix} = \widetilde{r}_i^T r_j + r_i^T\widetilde{r}_j = 2\widetilde{r}_i^T r_j. \quad \blacksquare$$

Věta 4.1 : Aplikujeme-li CG s maticí $Z = Z^{BC}$ na systém $\widehat{A}\widehat{x} = \widehat{b}$, platí

$$r_k = r_k^{BCG}, \quad x_k = x_k^{BCG}, \quad \widetilde{r}_k = \widetilde{r}_k^{BCG}. \tag{4.24}$$

Důkaz : Vektory r_k, $\widetilde{r}_k$, x_k generované CG splňují dle předchozích lemmat podmínky

1. $x_k \in x_0 + \mathcal{K}_k(A, r_0)$,
2. $r_k \in r_0 + A\ \mathcal{K}_k(A, r_0)$,
3. $\widetilde{r}_k \in \widetilde{r}_0 + A^T\mathcal{K}_k(A^T, \widetilde{r}_0)$,
4. $\widetilde{r}_i^T r_j = 0$ pro $i \neq j$.

Jak jsme si ukázali v kapitole 2. vektory r_k a $\widetilde{r}_k$ jsou těmito podmínkami určeny jednoznačně a platí tedy $r_k = r_k^{BCG}$, $\widetilde{r}_k = \widetilde{r}_k^{BCG}$ a $x_k = x_k^{BCG}$. ■

Kapitola 5

Numerické výsledky

Definice 5.1 : V následující tabulce definujeme zkratky používané v při popisu grafů.

Zkratka	Skutečný význam	Zkratka	Skutečný význam
Lg10(R)	$\log_{10}(\frac{\|r_k\|}{\|r_0\|})$	**Lg10(X)**	$\log_{10}(\frac{\|x_k - x^*\|}{\|x_0 - x^*\|})$
Čas	čas v sekundách	**Dim**	dimenze řešeného systému
NX	$\| x_k - x^* \|$	**NR**	$\| r_k \|$

Dále zkratkou **CGS LU** budeme značit předpodmíněnou metodu CGS, kde k předpodmínění bylo užito neúplného LU rozkladu. Zkratkou **CGS SK** potom značíme metodu CGS, která počítá skutečná rezidua (vektory $b - Ax_k$). Symbolem **CGS 1 2** označujeme zároveň dvě metody a sice metodu SSCGS a DSCGS (1 značí jednoduché zhlazení, 2 značí dvojnásobné zhlazení) .

5.1 Metody CGM a CGMS

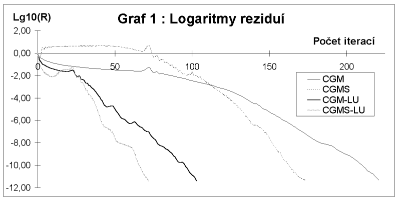
Obě metody jsme se snažili srovnat z hlediska rychlosti konvergence, počtu iterací a časové náročnosti. Pro test jsme použily Laplaceovu rovnici s Dirickletovou okrajovou podmínkou

$$-\frac{\partial^2 u(x,y)}{\partial x^2} - \frac{\partial^2 u(x,y)}{\partial y^2} = g \text{ na } \Omega,$$

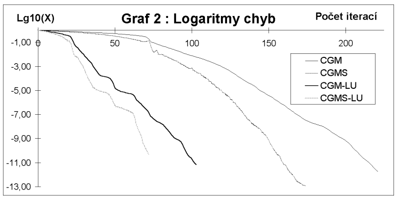
$$u/\partial\Omega = f, \text{ kde } \Omega \text{ je čtverec.}$$

Diskretizací úlohy na síti 100 x 100 a volbou $f = g = 0$ vznikne soustava se symetrickou, pozitivně definitní maticí řádu 10000 a nulovou pravou stranou. Známe tedy přesné řešení $x^* = (0, \ldots, 0)^T$. K počátečnímu odhadu řešení jsme použili vektor $x_0 = (1, \ldots, 1)^T$. Podmínkou zastavení algoritmů bylo $\| r_k \| < 10^{-10}$. Připomeňme, že umocnění metody CGM s sebou přineslo zlepšení odhadu vektoru chyby, ale též zvýšení počtu operací na jednu iteraci metody. Zajímá nás tedy především, zda rychlejší konvergence metody CGMS kompenzuje početní náročnost.

Konvergence metod

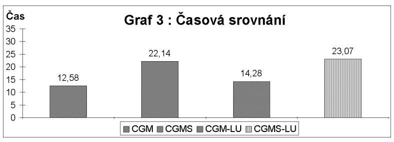


Podle očekávání konvergují metody s LU předpodmíněním na grafu 1 rychleji (co do počtu iterací) než metody bez předpodmínění. Povšimeme si, že normy reziduí nemusí klesat monotónně.



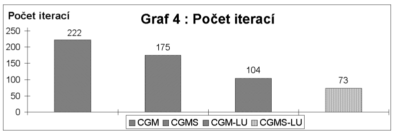
Metoda CGM zajišťuje monotónní klesání čísla $\| x_k - x^* \|_A$. Na grafu 2 je vidět, že metoda CGMS tuto podmínku zhruba u 75 iterace na několika místech porušuje. Co se týče odhadu chyby je vidět, že chyba u metody CGMS klesá zpočátku přibližně stejně rychle jako u CGM, avšak s přibývajícími iteracemi se rozdíl chyb začíná podstatně zvětšovat.

Časová náročnost



Metoda CGM užívá pouze jedno násobení matice-vektor zatímco metoda CGMS dvě násobení matice-vektor. Dá se tedy očekávat, že CGMS bude o něco pomalejší i při rychlejší konvergenci k přesnému řešení. Podobný názor můžeme vyslovit i při porovnání CGM LU a CGMS LU. Porovnáme-li metody CGM a CGM LU z hlediska rychlosti, vidíme na grafu 3, že obě metody nalezly přibližné řešení zhruba za stejný čas. Metoda CGM LU však potřebovala pouze poloviční množství iterací než metoda CGM viz. následující graf.

Rychlost konvergence



Na grafu 4 je vidět, že metody s předpodmíněním potřebují k dosažení požadované přesnosti přibližně dvakrát méně iterací než metody bez předpodmínění.

Závěrem můžeme zřejme konstatovat, že CGMS sice konverguje rychleji než CGM, ale spotřebuje na dosažení požadované přesnosti zhruba dvojnásobek času než metoda CGM.

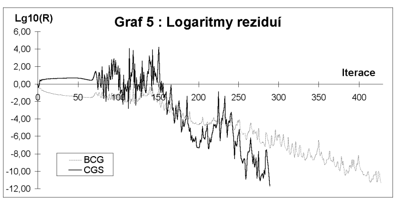
5.2 Metody BCG, CGS, SSCGS a DSCGS

Pro test metod jsme použili soustavu, kterou obdržíme diskretizací diferenciální rovnice na síti 100 x 100

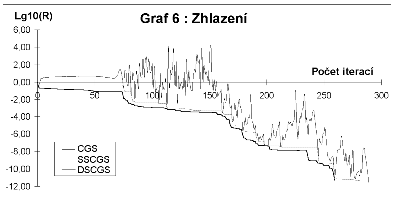
$$-\frac{\partial^2 u(x,y)}{\partial x^2} - \frac{\partial^2 u(x,y)}{\partial y^2} + 2\frac{\partial u(x,y)}{\partial x} + \frac{\partial u(x,y)}{\partial y} + u(x,y) = 0 \text{ na } \Omega, \tag{5.1}$$

$$u/\partial\Omega = 0, \tag{5.2}$$

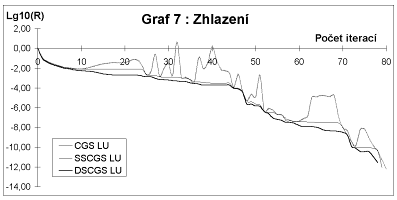
kde Ω je čtverec. Matice soustavy je řádu 10000 a soustava má nulovou pravou stranu. Opět tedy známe přesné řešení $x^* = (0, \ldots, 0)^T$. K počátečnímu odhadu řešení jsme použili vektor $x_0 = (1, \ldots, 1)^T$. Podmínkou zastavení algoritmů bylo $\| r_k \| < 10^{-10}$.

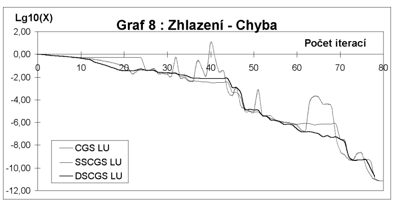


Na grafu 5 si povšimeme o mnoho větší oscilace metody CGS. Další zajímavost můžeme dobře vidět zhruba v rozmezí dvousté až třísté iterace a sice to, že metody oscilují na stejných místech, metoda CGS však s mnohem větší amplitudou. Z počátku se rezidua metody CGS drží vysoko nad rezidui metody BCG. Zlom však nastává přibližně u hodnoty -4, kde CGS rychle zkonverguje, zatímco BCG konverguje stále stejnou rychostí k cíli.



Z grafu 6 vidíme, že normy SSCGS skutečně klesají monotónně a tvoří jakýsi dolní obal reziduí oscilující CGS. Metoda DSCGS pak vylepšuje schodovitý tvar křivky klesání reziduí metody SSCGS.

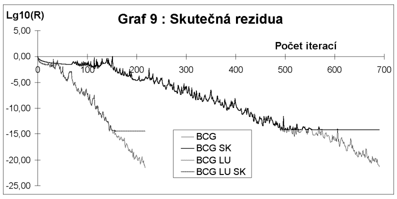


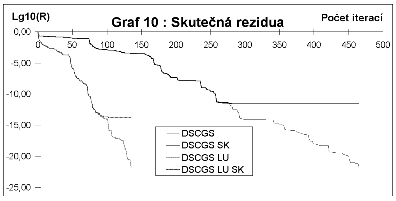


Vidíme, že zhlazením jsme se u chyby zbavili velkých oscilací, naplatí však již monotonie, což lze zahlédnout např. u dvacáté iterace.

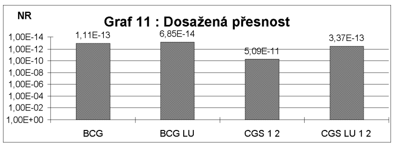
5.3 Dosažená přesnost

Řešme úlohu (5.1) s okrajovými podmínkami (5.2) na síti 100 x 100. Oproti minulému testu položme epsilon pro test zastavení $\varepsilon = 10^{-20}$. Generujme zároveň skutečná rezidua (skutečným reziduem rozumíme vektor $\widehat{r}_k = b - Ax_k$) a povšimeme si na následujících grafech rozdílu mezi skutečným a spočteným reziduem.

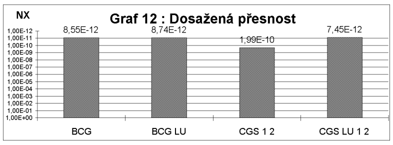




Na grafech je vidět, že zaokrouhlovací chyby znemožní lepší přiblížení k řešení, jestliže překročíme určitý požadavek přesnosti.



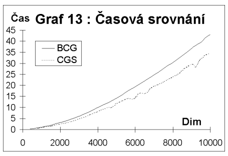
Z grafu 11 je vidět, že nejvyšší přesnosti dosáhla metoda BCG LU a nejhorší přesnosti metody CGS, SSCGS, DSCGS. Vysvětlení snad spočívá v tom, že u metody CGS vznikly větší zaokrouhlovací chyby vlivem velkých oscilací. Metody SSCGS a DSCGS, které vznikly z CGS zhlazením, pak byly postiženy stejnými chybami. U metody BCG LU zřejmě rozhodl malý počet iterací a malá aplituda oscilací, což nedovolilo vzniknout velkým zaokrouhlovacím chybám.

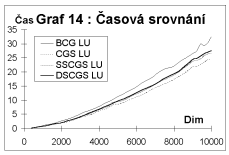


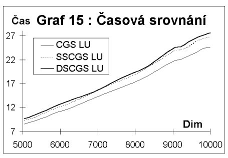
Zde můžeme konstatovat pouze to, co již bylo řečeno u předchozího grafu.

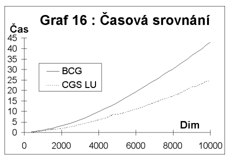
5.4 Testy pro různé řády matic

Řešme opět úlohu (5.1) s okrajovými podmínkami (5.2) . Provedeme-li diskretizaci úlohy na různě velkých oblastech, dostáváme matice různých řádů. Následující grafy shrnují chování metod s rostoucím řádem matice.

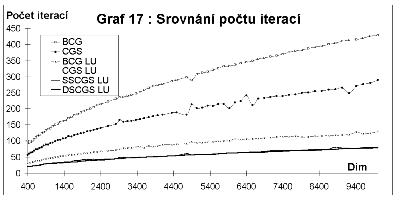








Na grafech můžeme sledovat, že s rostoucím řádem matice rychleji vyřeší soustavu metody s předpodmíněním. Na grafu 16 pak můžeme porovnat z hlediska času nejrychlejší a nejpomalejší metodu.



Porovnáme-li metody z hlediska počtu iterací vidíme, že nejlépe jsou na tom metody CGS LU, SSCGS LU, DSCGS LU u nichž počet iterací potřebných na vyřešení úlohy roste s řádem matice jen velmi mírně.

5.5 Srovnání metod CGS a BSMRZS

V programu metody BSMRZS jsme byli nuceni trochu poopravit testy, jež rozhodují o skoku. Testy

$$\begin{aligned}\mu_i &= (\tilde{r}_0, A^{i+1} z_k) < \varepsilon_1, \\ \sigma_{i+1} &= (\tilde{r}_0, s_{i+1}) < \varepsilon_1,\end{aligned}$$

jsme pozměnili na testy

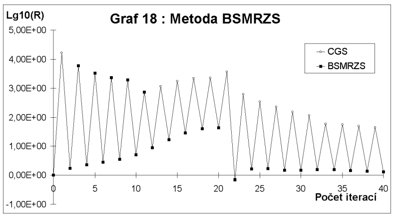
$$\begin{aligned}\frac{\mu_i}{\| \tilde{r}_0 \| \cdot \| A^{i+1} z_k \|} &= \frac{(\tilde{r}_0, A^{i+1} z_k)}{\| \tilde{r}_0 \| \cdot \| A^{i+1} z_k \|} < \varepsilon_1, \\ \frac{\sigma_{i+1}}{\| \tilde{r}_0 \| \cdot \| s_{i+1} \|} &= \frac{(\tilde{r}_0, s_{i+1})}{\| \tilde{r}_0 \| \cdot \| s_{i+1} \|} < \varepsilon_1.\end{aligned}$$

Největším problémem metody BSMRZS je počítání soustav pro koeficienty $\tilde{w}_i$, $\tilde{v}_i$, $\tilde{q}_i$, $\tilde{t}_i$. (Označme matici soustavy v dalším textu například B.) Numerické výsledky ukazují, že soustava s maticí B je s přibývajícími iteracemi stále hůře počitatelná. Pokud hledáme příčinu zjistíme, že velikosti prvků jednotlivých částí matice B (oddělené čárou) se od sebe s přibývajícími iteracemi začínají řádově lišit. Je tedy zřejmé, že při počítání soustavy s takovouto maticí gaussovou eliminací s částečnou pivotací dochází při dělení čísel velmi rozdílného řádu k velké ztrátě informace a tedy k velmi nepřesnému spočtení řešení. Původně jsme mysleli, že tento nedostatek je způsoben tím, že jsou polynomy $\mathcal{P}_k$ a $\mathcal{M}_k$ různě normovány. Normujeme-li polynom $\mathcal{M}_k$ podmínkou $\mathcal{M}_k(0) = 1$, docílíme sice toho, že $\pi_i \approx \mu_i$, ale nikoli toho, aby např. $g_{ij} \approx f_{ij}$ a soustava s maticí B se stává opět neřešitelným problémem. Dospěli jsme k názoru, že metoda BSMRZS je zajímavým teoretickým pokusem, jak se vyhnou zhroucení, není však vhodná pro počítání soustav s velkou řídkou maticí a to hned ze dvou příčin. První příčinou je velká početní a paměťová náročnost metody a druhou příčinou je nepočitatelnost soustav pro koeficiety $\widetilde{w}_i,\ \widetilde{v}_i,\ \widetilde{q}_i,\ \widetilde{t}_i$ s přibývajícím počtem iterací. Je však zajímavé alespoň na pár počátečních iteracích sledovat, kdy začíná metoda BSMRZS svůj skok.

Pro test metod jsme použili soustavu řádu 400, kterou obdržíme diskretizací diferenciální rovnice

$$\begin{aligned} -10^{-3}\frac{\partial^2 u(x,y)}{\partial x^2} - 10^{-3}\frac{\partial^2 u(x,y)}{\partial y^2} + 20\frac{\partial u(x,y)}{\partial x} + \frac{\partial u(x,y)}{\partial y} + u(x,y) &= 0 \text{ na } \Omega, \\ u/\partial\Omega &= 0,\ \Omega \text{ čtverec,} \end{aligned}$$

na síti 20 x 20.



Kapitola 6

Programy BCG, CGS, BSMRZS

6.1 Program BCG.FOR

Základní vstupní soubor programu : soubor BCG.INP

Struktura souboru BCG.INP

'matice.nsy'	Název vstupního souboru s maticí
'bcg.txt'	Název souboru s pomocným výpisem
1E-10	Epsilon pro test zastavení
1E-06	Epsilon podílu
1E-04	TOL pro LU rozklad
1000	Maximální počet iterací
1	Předpodmínění 1 .. LU 0 .. nic
0	Nastavení $\tilde{r}_0$ "0 .. = r_0" "1 .. = 1" "2 .. načíst"
1	Použito přesné řešení ? 0 .. NE 1.. ANO
1	Zapisovat průběh výpočtu do souboru ?
0	Nastavení x_0 "0 .. = 0" "1 .. = 1" "2 .. načíst"
0	Generovat skutečná rezidua

Poznámky k souboru BCG.INP :

Vstupní soubor s maticí je textový soubor, v němž je uložena nesymetrická matice běžným způsobem, dále je zde uložen vektor b a popřípadě vektor $\tilde{r}_0$.

Epsilon pro test zastavení je číslo, na kterém závisí ukončení iterací. Testuje se, zda v i-té iteraci není norma příslušného rezidua menší než toto epsilon.

Epsilon podílu je číslo, kterým určujeme, zda se mají v algoritmu počítat podíly čísel, je pro něž je $a/b > 1/Eps$.

TOL pro LU rozklad Je-li $TOL = 0$, provadí se neúplný LU rozklad matice A. Jinak se provádí neúplný LU rozklad matice $A + TOL * I$.

Maximální počet iterací — je-li iterací více než zadaná hodnota, ukončí se program.

Předpodmínění :

1 .. Je použito neúplného LU rozkladu

0 .. Bez použití předpodmínění

Volba $\widetilde{r}_0$:

0 .. $\widetilde{r}_0$ je voleno jako $\widetilde{r}_0 = r_0$

1 .. $\widetilde{r}_0$ je voleno jako $\widetilde{r}_0 = (1, \ldots, 1)^T$

2 .. $\widetilde{r}_0$ je načteno ze souboru matice.nsy

Použito přesné řešení : Určuje, zda lze během výpočtu použít přesné řešení k odhadu chyby.

1 .. Ano

0 .. Ne

Zapisovat průběh výpočtu do bcg.txt :

1 .. Ano

0 .. Ne

Volba x_0 :

0 .. x_0 je voleno jako $x_0 = (0, \ldots, 0)^T$

1 .. x_0 je voleno jako $x_0 = (1, \ldots, 1)^T$

2 .. x_0 je načteno ze souboru bcg.x0

Generovat skutečná rezidua :

1 .. Ano - Počítá v každém kroku $b - Ax_k$

0 .. Ne

Použité soubory

bcg.inp — soubor vstupních parametrů

bcg.for — zdrojový text programu

bcg.x — výstup - řešení spočteného programem

bcg.x0 — počáteční aproximace řešení

bcg.xp — vstup - přesné řešení použité k odhadu chyby - není-li přítomen, je nutno nastavit v souboru bcg.inp parametr " Použít přesné řešení " na 0

bcgmenu.exe — program, jež umí modifikovat vstupní soubor bcg.inp.

bcg.txt — výstupní soubor se zápisem průběhu výpočtu. Označme $\widehat{r}_k = b - Ax_k$.

1.sl	2.sl	3.sl	4.sl	5.sl	6.sl	7.sl	8.sl	9.sl
Itr	$\ln\left(\frac{\|r_k\|}{\|r_0\|}\right)$	$\ln\left(\frac{\|\widehat{r}_k\|}{\|\widehat{r}_0\|}\right)$	$\| r_k \|$	$\| \widehat{r}_k \|$	$\| x_k - x^* \|$	$\ln\left(\frac{\|x_k - x^*\|}{\|x_0 - x^*\|}\right)$	α_k	β_k

matice.nsy — soubor s maticí A, vektorem b a vektorem $\widetilde{r_0}$.

Reprezentace algoritmu v programu

Input $x_0,\ A,\ b,\ \widetilde{r}_0$;
$r_0 = b - Ax_0$;
$p_0 = r_0;\ \widetilde{p}_0 = \widetilde{r}_0$;
for $k = 0$ **to** $n-1$ **do**
  if $r_k = 0$ **then** $x^* = x_k$ *and exit*
  else
    begin
      $\alpha_k = \frac{\widetilde{r}_k^T M r_k}{\widetilde{p}_k^T A p_k}$;
      $x_{k+1} = x_k + \alpha_k p_k$;
      $r_{k+1} = r_k - \alpha_k A p_k;\ \widetilde{r}_{k+1} = \widetilde{r}_k - \alpha_k A^T \widetilde{p}_k$;
      $\beta_k = \frac{\widetilde{r}_{k+1}^T M r_{k+1}}{\widetilde{r}_k^T M r_k}$
      $p_{k+1} = M r_{k+1} + \beta_k p_k;\ \widetilde{p}_{k+1} = M^T \widetilde{r}_{k+1} + \beta_k \widetilde{p}_k$;
    end;
$x^* = x_n$.

Popis proměnných v programu

ALFA	koeficient α_k
B	vektor b
BETA	koeficient β_k
D	vektor $\widetilde{r}_0$
DIA,JSL,IRA	matice A
EPS(1)	Epsilon pro ukončení iterací
EPS(2)	Epsilon podílu
EPS(3)	TOL pro LU rozklad
IER	Indikace výpočtu
INF	Proměnná pro LU
INIT	Nastaven na INIT=0 pokud 1.iterace
JMENO(1)	Soubor se vstupní maticí
JMENO(2)	Výstupní soubor (cgs.txt)
JMENO(3)	'BCG.X0'
JMENO(4)	'BCG.XP'
JMENO(5)	'BCG.X'
LGR	$\ln\left(\frac{\|r_k\|}{\|r_0\|}\right)$
LGSN	$\ln\left(\frac{\|\widehat{r}_k\|}{\|\widehat{r}_0\|}\right)$
LGX	$\ln\left(\frac{\|x_k - x^*\|}{\|x_0 - x^*\|}\right)$

MAXN	maximální řád matice
N	řád matice A
PAR(1)	Řád matice
PAR(2)	Maximální řád matice
PAR(3)	Maximální počet iterací
PAR(4)	Druh předpodmínění 1 = LU
PAR(5)	Volba vektoru $\widetilde{r}_0$
PAR(6)	Použít vektoru XP ?
PAR(7)	Zapisovat průběh výpočtu ?
PAR(8)	Volba vektoru x_0
PAR(9)	Generovat skutecna rezidua ?
PAS	Max. počet nenul. prvků na řádku
R	vektor r_k
Res	$\| r_k \|$
ResX	$\| x_k - x^* \|$
RR1,RR2	Proměnné pro skal.součiny $\widetilde{r}_k^T r_k$ a $\widetilde{r}_{k+1}^T r_{k+1}$
SAPK	Proměnná pro skal. součin $\widetilde{p}_k^T A p_k$
S(I)	Proměnné typu character pro výpis
SN	$\| \widehat{r}_k \|$
WD(DIALU)	Prvky matice M^{-1}
WD(P)	vektor p_k
WD(S)	vektor $\widetilde{p}_k$
WD(XP)	vektor x^*
WD(Y)	Pomocný vektor
WI(ID),WI(IW)	Vektory pro LU
WI(IRALU)	Pocet prvku na radku M^{-1}
WI(JSLLU)	Sloupkové souradnice M^{-1}
WI(PERM)	Vektor permutací pro LU
WO(1)	$\| r_0 \|$
WO(2)	$\| r_k \|$
WO(3)	$\| x_k - x^* \|$
WO(4)	α_k
WO(5)	β_k
WO(6)	$\| x_0 - x^* \|$
X	vektor x_k
Z	vektor $\widetilde{r}_k$

6.2 Program CGS.FOR

Základní vstupní soubor programu : soubor CGS.INP

Struktura souboru CGS.INP

'matice.nsy'	Název vstupního souboru
'cgs.txt'	Název souboru s pomocným výpisem
1E-10	Epsilon pro test zastavení
1E-06	Epsilon podílu
1E-04	TOL pro LU rozklad
1000	Maximalní počet iterací
1	Předpodmínění 1 .. LU 0 .. nic
0	Nastavení $\tilde{r}_0$ "0 .. = r_0" "1 .. = 1" "2 .. načíst"
1	Použito přesné řešení ? 0 .. NE 1.. ANO
1	Zapisovat průběh výpočtu do souboru ?
2	Zhlazení 0-žádné 1-jednoduché 2-dvojnásobné
0	Nastavení x_0 "0 .. = 0" "1 .. = 1" "2 .. načíst"
0	Generovat skutečná rezidua

Poznámky k souboru CGS.INP : viz poznámky k BCG.INP

Zhlazení definuje, zda má být v metodě CGS použito zhlazení.

0 .. Nebude použito zhlazení

1 .. Bude použito jednoduché zhlazení

2 .. Bude použito dvojnásobné zhlazení

Použité soubory

cgs.inp — soubor vstupních parametrů

cgs.for — zdrojový text programu

cgs.x — výstup - řešení spočteného programem

cgs.x0 — vstup - počáteční aproximace řešení

cgs.xp — vstup - přesné řešení použité k odhadu chyby

cgsmenu.exe — program, jež umí modifikovat vstupní soubor cgs.inp.

cgs.txt — výstupní soubor se zápisem průběhu výpočtu. Označme $\widehat{r}_k = b - Ax_k$.

1.sl	2.sl	3.sl	4.sl	5.sl	6.sl	7.sl	8.sl	9.sl
Itr	$\ln\left(\frac{\|r_k\|}{\|r_0\|}\right)$	$\ln\left(\frac{\|\widehat{r}_k\|}{\|\widehat{r}_0\|}\right)$	$\| r_k \|$	$\| \widehat{r}_k \|$	$\| x_k - x^* \|$	$\ln\left(\frac{\|x_k - x^*\|}{\|x_0 - x^*\|}\right)$	α_k	β_k

matice.nsy — soubor s maticí A, vektorem b a vektorem $\widetilde{r_0}$.

Reprezentace algoritmu v programu

Input $\bar{x}_0,\ A,\ b,\ \widetilde{r}_0,\ M;$
$\bar{r}_0 = b - A\bar{x}_0;$
$r_0 = \bar{r}_0;\ x_0 = \bar{x}_0;$
$\bar{p}_0 = u_0 = M\bar{r}_0;$
for $k = 0$ **to** $n-1$ **do**
  if $r_k = 0$ **then** $x^* = x_k$ *and exit*
  else
    begin
      $v_k = A\bar{p}_k;\ \ \alpha_k = \frac{\widetilde{r}_0^T M\bar{r}_k}{\widetilde{r}_0^T M v_k};$
      $q_k = u_k - \alpha_k M v_k;$
      $\bar{r}_{k+1} = \bar{r}_k - \alpha_k A(u_k + q_k);$
      $\bar{x}_{k+1} = \bar{x}_k + \alpha_k(u_k + q_k);$
      $\beta_k = \frac{\widetilde{r}_0^T M\bar{r}_{k+1}}{\widetilde{r}_0^T M\bar{r}_k};$
      $u_{k+1} = M\bar{r}_{k+1} + \beta_k q_k;$
      $\bar{p}_{k+1} = u_{k+1} + \beta_k(q_k + \beta_k\bar{p}_k);$
      $\begin{pmatrix}\lambda_k\\ \mu_k\end{pmatrix} = -(V_k^T V_k)^{-1} V_k^T \bar{r}_{k+1};$
      $x_{k+1} = \bar{x}_{k+1} + \lambda_k(x_k - \bar{x}_{k+1}) - \mu_k\bar{p}_k;$
      $r_{k+1} = \bar{r}_{k+1} + \lambda_k(r_k - \bar{r}_{k+1}) + \mu_k v_k;$
    end;
$x^* = x_n.$

Popis proměnných v programu

A,B,C,D,E,DET	Pom. proměnné pro výpočet λ_k a μ_k.
I,J	Proměnné pro cykly
LAMBDA	koeficient λ_k
MU	koeficient μ_k
PAR(8)	Kolikanásobné zhlazení ?
PAR(9)	Volba vektoru x_0
PAR(10)	Generovat skutečná rezidua ?
R	vektor $\bar{r}_k$
RMV	Proměnná pro skal. součin $\widetilde{r}_0^T M v_k$
WD(L),WD(Y)	Pomocné vektory
WD(P)	vektor $\bar{p}_k$
WD(Q)	vektor q_k
WD(RZ)	vektor r_k
WD(U)	vektor u_k
WD(V)	vektor v_k
WD(XZ)	vektor x_k
X	vektor $\bar{x}_k$
Z	vektor $M^T\widetilde{r}_0$

Ostatní proměnné viz. proměnné u programu BCG.FOR

6.3 Program BSMRZS.FOR

Základní vstupní soubor programu : soubor BSMRZS.INP

Struktura souboru BSMRZS.INP

'matice.nsy'	Název vstupního souboru s maticí
'bsmrzs.txt'	Název souboru s pomocným výpisem
1E-07	Epsilon - skoro-zhroucení
1E-06	Epsilon - Gauss
1E-10	Epsilon - Zastaveni
1000	Maximální počet iterací
1	Zapisovat průběh výpočtu do souboru ?
0	Nastavení x_0 "0 .. = 0" "1 .. = 1" "2 .. načíst"
0	Nastavení $\tilde{r}_0$ "0 .. = r_0" "1 .. = 1" "2 .. načíst"
1	Použito přesné řešení ? 0 .. NE 1.. ANO
0	Generovat skutečná rezidua

Poznámky k souboru BSMRZS.INP : viz poznámky k souboru BCG.INP

Epsilon - skoro-zhroucení definuje, kdy dojde ke skoku.

Epsilon - Gauss je-li pivot při počítání soustavy rovnic menší než toto epsilon, pokračuje skok

Epsilon - Zastavení je-li norma rezidua menší než toto epsilon, dochází k zastavení algoritmu

Použité soubory

bsmrzs.inp — soubor vstupních parametrů

bsmrzs.for — zdrojový text programu

bsmrzs.x — výstup - řešení spočteného programem

bsmrzs.x0 — počáteční aproximace řešení

bsmrzs.xp — vstup - přesné řešení použité k odhadu chyby

bsmrmenu.exe — program, jež umí modifikovat vstupní soubor bsmrzs.inp.

bsmrzs.txt — výstupní soubor se zápisem přůběhu výpočtu. Označme $\widehat{r}_k = b - Ax_k$.

1.sl	2.sl	3.sl	4.sl	5.sl	6.sl	7.sl
Iterace	$\ln\left(\frac{\|r_k\|}{\|r_0\|}\right)$	$\ln\left(\frac{\|\widehat{r}_k\|}{\|\widehat{r}_0\|}\right)$	$\| r_k \|$	$\| \widehat{r}_k \|$	$\| x_k - x^* \|$	$\ln\left(\frac{\|x_k-x^*\|}{\|x_0-x^*\|}\right)$

matice.nsy — soubor s maticí A, vektorem b a popř. s vektorem $\widetilde{r_0}$.

Popis proměnných v programu

AB	Pracovní vektor
ACT	$\widehat{r}_k$
DIA,JSL,IRA	A
C(i)	μ_i
CB	b
D(i)	π_i
EPS	ε_1
EPS1	ε_2
EPS2	ε_3
EPSI(1)	Epsilon Skoro-zhrouceni
EPSI(2)	Epsilon Gauss
EPSI(3)	Epsilon Zastavení
ETA	$\mathcal{Q}_k$
ETAP	$\mathcal{T}_k$
GAMMA	$\mathcal{W}_k$
GAMMAP	$\mathcal{V}_k$
IER	Indikace výpočtu
INIT	Nastaven na INIT=0 pokud 1.iterace
IR	Maximální řád matice
JMENO(i)	viz. program BCG.FOR
LGR	$\ln\left(\frac{\lVert r_k\rVert}{\lVert r_0\rVert}\right)$
LGSN	$\ln\left(\frac{\lVert\widehat{r}_k\rVert}{\lVert\widehat{r}_0\rVert}\right)$
LGX	$\ln\left(\frac{\lVert x_k-x^*\rVert}{\lVert x_0-x^*\rVert}\right)$
LMAT	Pracovní matice (soustavy pro polynomy)
MAXBCK	MJ
MAXN	Maximální možná hodnota $n_k+\delta_k$
MK	δ_k
N	Řád matice A
NBC	Maximální počet iterací pro dannou úlohu
NK	n_k
NORM	$\parallel r_k \parallel$
NX	$\parallel x_k - x^* \parallel$
P	$\mathcal{P}_k$
P1	$\mathcal{M}_k$
PAS	Max. počet nenul. prvků na řádku
PAR(1)	Řád matice
PAR(2)	Maximální řád matice
PAR(3)	Maximální počet iterací
PAR(4)	Druh předpodmínění 1 = LU
PAR(5)	Volba vektoru $\widetilde{r}_0$
PAR(6)	Použít vektoru XP ?
PAR(7)	Zapisovat průběh výpočtu ?
PAR(8)	Volba vektoru x_0
PAR(9)	Generovat skutečná rezidua ?

PWK..................	Pracovní vektor
R	r_k
R0	$\| r_0 \|$
RMAT	Pracovní matice (soustavy pro polynomy)
S(I)	Proměnné typu character pro výpis
SIGMA	σ_k
SN	$\| \widehat{r}_k \|$
U(i)	$A^i s_k$
V(i)	$A^i z_k$
W(i)	$A^i r_k$
X	x_k
XP	x^*
Y	$\widetilde{r}_0$

Literatura

[Brezn–91] C.Brezinski,M.Redivo-Zaglia and H. Sadok : Avoiding breakdown and near-breakdown in Lanczos type algorithms, Numerical Algorithms 1 (1991) 261-284, J.C.Baltzer AG, Science Publishers.

[Brezn–94] C.Brezinski,M.Redivo-Zaglia : Treatment of near-breakdown in the CGS algorithm, Numerical Algorithms 7 (1994) 33-73, J.C.Baltzer AG, Science Publishers.

[Weiss–95] R.Weiss : A theoretical overview of Krylov subspace methods, Applied Numerical Mathematics 19 (1995) 207-233.

[Leary–80] D.P.O'Leary : The Block Conjugate Gradient Algorithm, Linear Algebra Appl. 99 (1980) 293-322.

[Fiedl–81] M.Fiedler : Speciální matice a jejich použití v numerické matematice, SNTL, Praha (1981).

[Golub–80] G.H.Golub : Iterative Methods for Linear Systems, The John Hopkins University Press, Baltimore 1984.