MATEMATICKO-FYZIKÁLNÍ FAKULTA

UNIVERZITY KARLOVY

Diplomová práce

Iterační metody pro obsáhlé řídké soustavy lineárních rovnic

C t i r a d M a t o n o h a

KATEDRA NUMERICKÉ MATEMATIKY

Vedoucí diplomové práce: Doc. RNDr. Jan Zítko, CSc.
Obor: Matematika
Zaměření: Výpočtová matematika

Prohlašuji, že jsem svoji diplomovou práci zpracoval samostatně s použitím uvedené literatury.

V Praze dne 7.4.1998 .....

Obsah

Úvod

Jedním ze základních úkolů v numerické matematice je řešení soustav lineárních algebraických rovnic $Ax = f$. Takové systémy vznikají např. při řešení parciálních diferenciálních rovnic, ať už metodou konečných diferencí nebo metodou konečných prvků, anebo mohou vzniknout i při řešení nelineárních rovnic. Algoritmy na řešení nelineárních rovnic vyžadují obvykle opakované řešení lineárních algebraických rovnic (tj. v cyklu). V této práci budeme předpokládat, že matice soustavy je čtvercová, regulární a že pravá strana je vektor a nikoli obdélníková matice. Symetrii předpokládat nebudeme a zaměříme se hlavně na řešení soustav s nesymetrickou maticí.

V zásadě existují dva přístupy k řešení soustav, a sice přímé a iterační postupy. Přímé postupy včetně všech rozkladů jsou založeny na Gaussově eliminaci. V tomto případě obdržíme řešení po konečném počtu aritmetických operací. V případě iteračních metod sestrojujeme posloupnost vektorů, které konvergují k přesnému řešení soustavy. Přímé metody se používají hlavně v případech, ve kterých je řád matice malý a řeší se soustavy s jednou maticí pro více pravých stran. S rozvojem rychlých počítačů si postupně získaly na popularitě postupy iterační, které se používají hlavně pro řešení řídkých a obsáhlých systémů.

V roce 1952 Hestenes a Stiefel odvodili metodu sdružených gradientů pro řešení soustavy lineárních rovnic se symetrickou a positivně definitní maticí. I když je tato metoda takovou metodou, u které bychom měli teoreticky obdržet přesné řešení po konečném počtu kroků, ukázalo se v praxi, že zejména pro špatně podmíněné systémy tomu tak není a z tohoto hlediska byla zařazována i mezi iterační metody. Plného zhodnocení se této metodě dostalo asi po dvaceti letech, kdy se použitím předpodmínění ukázala tato metoda být efektivní. Pak se odvodila celá řada postupů nejen pro symetrické, ale i pro nesymetrické problémy. Obecně se těmto postupům říká zobecněné CG-metody.

V této práci se budeme zabývat metodami, které studují různé projekce na Krylovovy podprostory. Cílem práce bude systematický výklad všech známých metod tohoto typu, uvedení algoritmů a základních vlastností včetně numerických testů. Pod pojmem *ortogonalizační metoda* rozumíme iterační metodu splňující následující relace, viz [Weiss].

Pro $k \geq 1$ sestrojujeme aproximace x_k k přesnému řešení $x^\star$ rovnice $Ax = f$ ve tvaru

$$x_k \in \tilde{x}_k + \mathrm{span}(q_{k-\sigma_k,k}, ..., q_{k-1,k}),$$

kde

$$\sigma_k \leq k, \quad \tilde{x}_k \in sp(x_{k-\sigma_k}, ..., x_{k-1}).$$

Vektory $q_{k-i,k} \in \mathbb{R}^n$ jsou dané směry, které splňují ortogonální podmínky

$$(r_k, Z_k q_{k-i,k}) = 0 \text{ pro } i = 1, ..., \sigma_k,$$

kde Z_k jsou pomocné regulární ortogonalizační matice a $r_k = f - Ax_k$ je residuum.

Mezi nejpopulárnější ortogonalizační metody patří metody Krylovových podprostorů (*Krylov subspace methods*), které vycházejí z předpokladu, že

$$q_{k-i,k} \in K_{k-i}(B,z) = sp(z, Bz, ..., B^{k-i}z) \text{ a } \tilde{x}_k = x_{k-\sigma_k}, \text{ kde } B \in \mathbb{R}^{n\times n} \text{ a } z \in \mathbb{R}^n.$$

Je-li $B = A$ a $z = r_0$, pak mluvíme o sdružených metodách Krylovova podprostoru (*Conjugate Krylov subspace methods*).

Na začátek uvedeme označení, které budeme v této práci používat. Je-li $v \in \mathbb{R}^n$, $v \neq 0$, pak prostor

$$\mathcal{K}_i(v, A) = (v, Av, ..., A^{i-1}v)$$

nazveme Krylovovým podprostorem generovaným maticí A a vektorem v.
Označme

$$M = (A + A^T)/2, \text{ resp. } R = (A^T - A)/2$$

symetrickou, resp. antisymetrickou část matice A. Předpokládejme navíc, že M je positivně definitní. Evidentně platí $A = M - R$ a $(y, Ry) = 0\ \forall y$.

Symbol $\sigma(A)$ značí množinu vlastních čísel matice A a $\lambda(A)$ je nějaké vlastní číslo matice A. Je-li matice A symetrická, pak nejmenší, resp. největší vlastní číslo této matice označíme $\lambda_{\min}$, resp. $\lambda_{\max}$. Symbolem $\rho(A)$ označíme spektrální poloměr matice A, přičemž platí $\rho(A) = |\lambda_{\max}(A)|$. Je-li A regulární, pak spektrální číslo podmíněnosti $\kappa(A)$ matice A je definováno vztahem $\kappa(A) := \| A \|_2 \cdot \| A^{-1} \|_2$. Poznamenáváme, že je-li A symetrická, pak $\kappa(A) = |\lambda_{\max}(A)|/|\lambda_{\min}(A)|$. Dále ke každé matici existuje regulární matice T tak, že platí $J = T^{-1}AT$, kde J je Jordanův kanonický tvar matice A.

Algoritmy, které uvedeme, budou s předpodmíněním. Předpodmínění znamená, že místo soustavy $Ax = f$ řešíme soustavu

$$Q^{-1}Ax = Q^{-1}f \quad \text{– předpodmínění zleva, nebo}$$

$$AQ^{-1}Qx = f \quad \text{– předpodmínění zprava,}$$

kde Q je regulární matice. Ve druhém případě je řešením soustavy vektor Qx. Protože matice $Q^{-1}A$ a AQ^{-1} jsou si podobné, dá se očekávat, že v případě symetrické matice A budou použité metody v obou případech předpodmínění počítat stejně rychle. Pro nesymetrickou matici A budeme uvažovat předpodmínění zprava i zleva. Abychom to zapsali jedním způsobem, budeme předpokládat, že máme dvě regulární matice Q_1 a Q_2 a kromě soustavy $Ax = f$ budeme též uvažovat soustavu

$$[Q_1^{-1}AQ_2^{-1}][Q_2x] = Q_1^{-1}f, \text{ neboli } \tilde{A}\tilde{x} = \tilde{f},$$

kde položíme

$$\tilde{A} = Q_1^{-1}AQ_2^{-1},\ \tilde{x} = Q_2x,\ \tilde{f} = Q_1^{-1}f.$$

Poznamenáváme, že pro předpodmínění zprava je $Q_1 = I$ a $Q_2 = Q$ a pro předpodmínění zleva je $Q_1 = Q$ a $Q_2 = I$. Případ $Q_1 \neq I$ a $Q_2 \neq I$ se v praxi neužívá.

Pro residuum platí

$$\tilde{r} = Q_1^{-1}r,$$

neboť

$$\tilde{r} = \tilde{f} - \tilde{A}\tilde{x} = Q_1^{-1}f - Q_1^{-1}AQ_2^{-1}Q_2x = Q_1^{-1}f - Q_1^{-1}Ax = Q_1^{-1}(f - Ax) = Q_1^{-1}r.$$

Dále označíme

$$\tilde{M} = Q_1^{-1} M Q_2^{-1} = \frac{Q_1^{-1} A Q_2^{-1} + (Q_1^{-1} A Q_2^{-1})^T}{2} = \frac{\tilde{A} + \tilde{A}^T}{2},$$

resp.

$$\tilde{R} = Q_1^{-1} R Q_2^{-1} = \frac{(Q_1^{-1} A Q_2^{-1})^T - Q_1^{-1} A Q_2^{-1}}{2} = \frac{\tilde{A}^T - \tilde{A}}{2}$$

symetrickou, resp. antisymetrickou část matice $\tilde{A}$. I v tomto případě platí $\tilde{A} = \tilde{M} - \tilde{R}$ a $(y, \tilde{R}y) = 0 \quad \forall y$.

Cílem předpodmínění je to, abychom zmenšili číslo podmíněnosti matice soustavy. Je-li matice Q dobrá aproximace matice A, tj. $Q \approx A$, pak se dá očekávat, že matice $\tilde{A} = Q^{-1}A$ nebo $\tilde{A} = AQ^{-1}$ je „blízko“ jednotkové matici I a iterační metody konvergují rychleji, jsou-li aplikovány na předpodmíněnou soustavu, než na soustavu bez předpodmínění. Na druhé straně ovšem přibude dodatečná práce násobení matic a vektorů maticí Q.

V uvedených algoritmech budeme psát: „Pro $i = 1, 2, ...$ provedeme“, přičemž automaticky předpokládáme, že počet iterací je ohraničen maximálním počtem iterací a navíc zastavení algoritmu je testováno podmínkou, že norma residua je menší než zadaná tolerance. Vzhledem k tomu, že v této práci představujeme mnoho algoritmů, neuvádíme tyto testovací podmínky.

V první kapitole nejprve stručně popíšeme některé metody na řešení systému $Ax = f$, předvedeme algoritmy, odvodíme algoritmy s předpodmíněním, uvedeme některé vlastnosti konstruovaných vektorů, odhadneme normy residuí a tím získáme konvergenci. Metody rozdělíme na dvě skupiny podle toho, jak se navzájem chovají residua, tzn. metody, ve kterých jsou residua navzájem A-ortogonální a kde ortogonální.

Ve druhé kapitole se podrobněji zaměříme na Axelssonovu metodu. Opět odvodíme algoritmus a to i s předpodmíněním, dokážeme některé vlastnosti vektorů a koeficientů, které vystupují v algoritmu a ukážeme, po kolika krocích získáme nulové residuum. Nakonec odpovíme na otázku, kdy lze algoritmus useknout, což znamená pracovat s méně vektory, aby byl useknutý algoritmus totožný s algoritmem neuseknutým, čili s plnou verzí.

Ve třetí kapitole představíme opět trochu podrobněji metodu GMRES, u které uvedeme algoritmus, převedeme ho opět do předpodmíněného tvaru, uvedeme vztahy mezi vektory a koeficienty v algoritmu a ukážeme, kdy získáme přesné řešení soustavy $Ax = f$. Dále odhadneme normu residua a ukážeme, na čem tento odhad závisí a na závěr dokážeme, kdy konverguje restartovaná metoda.

Ve čtvrté kapitole vyzkoušíme uvedené algoritmy v praxi na konkrétní soustavě, kterou získáme diskretizací nějaké diferenciální rovnice. Srovnáme použití algoritmů na menší, střední a větší matice a vyzkoušíme také předpodmínění dané soustavy. Pokusíme se sledovat rychlost výpočtu, ale čas budeme považovat pouze za informativní. Také budeme sledovat, jak se spočtené řešení jednotlivých metod liší od přesného řešení, které budeme znát. Na závěr to vše zhodnotíme a řekneme, které metody jsou nejlepší.

Na závěr úvodu je mojí milou povinností poděkovat za pomoc vedoucímu diplomové práce Doc. RNDr. Janu Zítkovi, CSc., který mi v průběhu celé práce a při úpravě textu poskytoval cenné rady a připomínky.

Kapitola 1

Přehled ortogonalizačních metod na řešení nesymetrických soustav

V této kapitole uvedeme přehled metod na řešení soustavy s nesymetrickou maticí (viz [Elman]). Ke každé metodě uvedeme algoritmus, základní vlastnosti a věty o konvergenci. Všechny algoritmy i věty budou formulovány s oboustranným předpodmíněním, tak, jak je uvedeno v úvodu.

1.1 Metody založené na A-ortogonalitě residuí

Nejprve představíme metody, ve kterých jsou residua $\{r_i\}$ iterací $\{x_i\}$ navzájem A-ortogonální, tj. platí

$$(r_i, Ar_j) = 0 \quad \text{pro} \quad i > j.$$

Tato rovnost platí pro $i \neq j$ pouze pro symetrické systémy. Uvažujeme-li předpodmínění, platí pro tato residua podmínka

$$(Q_1^{-1} r_i, Q_1^{-1} A Q_2^{-1} Q_1^{-1} r_j) = 0 \quad \text{pro} \quad i > j,$$

eventuelně pro $i \neq j$ v symetrickém případě.

1.1.1 Zobecněná metoda sdružených residuí GCR

Pro soustavu

$$Ax = f, \tag{1.1}$$

kde A je symetrická a positivně definitní matice, splňuje i-tá aproximace x_i při použití metody sdružených residuí následující minimalizační podmínku:

$$\| x_i - x^\star \| = \min_{x \in x_0 + \mathcal{K}_i(r_0, A)} (A[x^\star - x], A[x^\star - x])^{\frac{1}{2}} = \min_{x \in x_0 + \mathcal{K}_i(r_0, A)} \| f - Ax \|, \tag{1.2}$$

kde $x^\star$ je přesné řešení soustavy (1.1) a $r_0 = f - Ax_0$. Posloupnost iterací $\{x_i\}$ se dá počítat jednokrokovou rekurencí

$$x_{i+1} = x_i + \alpha_i p_i, \tag{1.3}$$

kde směrové vektory p_i se počítají podle rekurencí

$$p_0 = r_0, \quad p_{i+1} = r_{i+1} + \beta_i p_i, \tag{1.4}$$

kde koeficienty β_i jsou voleny tak, že pro směrové vektory platí

$$(Ap_i, Ap_j) = 0 \quad \text{pro} \quad i \neq j. \tag{1.5}$$

Číslo α_i se volí tak, že platí

$$\alpha_i = \arg\min_{\alpha>0} \parallel f - A(x_i + \alpha p_i) \parallel . \tag{1.6}$$

Z této úvahy vyplyne následující algoritmus.

Algoritmus 1.1
Zvolíme x_0.
Spočteme $r_0 = f - Ax_0$.
Položíme $p_0 = r_0$.
Pro $i = 0, 1, 2, ...$ **provedeme**
$\alpha_i = \frac{(r_i, Ap_i)}{(Ap_i, Ap_i)}$
$x_{i+1} = x_i + \alpha_i p_i$
$r_{i+1} = r_i - \alpha_i Ap_i$
$\beta_i = \frac{(r_{i+1}, Ar_{i+1})}{(r_i, Ar_i)}$
$p_{i+1} = r_{i+1} + \beta_i p_i$
`Konec cyklu pro` i.
Konec Algoritmu.

Metoda vycházející z tohoto algoritmu se nazývá „**metoda sdružených residuí (CR)** *(Conjugate Residual method)*“.

Nyní přejdeme k nesymetrickému případu. Tady lze množinu směrů $\{p_i\}$ splňujících podmínky

$$(Ap_i, Ap_j) = 0 \text{ pro } i \neq j$$

spočítat takto:

$$p_{i+1} = r_{i+1} + \sum_{j=0}^{i} \beta_j^{(i)} p_j,$$

kde

$$\beta_j^{(i)} = -\frac{(Ar_{i+1}, Ap_j)}{(Ap_j, Ap_j)} \text{ pro } j \leq i.$$

Délku kroku α_i, pro kterou platí (1.6), je stejně jako v symetrickém případě dána vzorcem

$$\alpha_i = \frac{(r_i, Ap_i)}{(Ap_i, Ap_i)}.$$

Teď již můžeme napsat celý algoritmus.

Algoritmus 1.2

Zvolíme x_0.
Spočteme $r_0 = f - Ax_0$.
Položíme $p_0 = r_0$.
Pro $i = 0, 1, 2, ...$ **provedeme**
$\alpha_i = \frac{(r_i, Ap_i)}{(Ap_i, Ap_i)}$
$x_{i+1} = x_i + \alpha_i p_i$
$r_{i+1} = r_i - \alpha_i Ap_i$
$\beta_j^{(i)} = -\frac{(Ar_{i+1}, Ap_j)}{(Ap_j, Ap_j)}$ pro $j \leq i$
$p_{i+1} = r_{i+1} + \sum_{j=0}^{i} \beta_j^{(i)} p_j$
`Konec cyklu pro` i.
Konec Algoritmu.

Metoda vycházející z tohoto algoritmu se nazývá „**zobecněná metoda sdružených residuí (GCR)** *(Generalized Conjugate Residual method)*“.

Jak jsme již uvedli, budou všechny algoritmy s předpodmíněním. Na metodě GCR ukážeme, jak se odvodí algoritmus s předpodmíněním z algoritmu bez předpodmínění. Ostatní metody budou již pouze s předpodmíněním, tj. daný algoritmus bez předpodmínění se dostane tak, že se za Q_1 a Q_2 dosadí jednotkové matice I. Stejně tak pro předpodmínění zleva je $Q_2 = I$ a zprava je $Q_1 = I$.
Jak je to s ortogonalitou směrů $\{p_i\}$? Víme, že vlnkované směry jsou $A^T A$-ortogonální. Definujme

$$Q_2^{-1}\tilde{p}_i = p_i.$$

Platí tedy

$$0 = (\tilde{A}\tilde{p}_i, \tilde{A}\tilde{p}_j) = (Q_1^{-1}AQ_2^{-1}Q_2p_i, Q_1^{-1}AQ_2^{-1}Q_2p_j) = (Q_1^{-1}Ap_i, Q_1^{-1}Ap_j) \text{ pro } i \neq j.$$

Vidíme, že v případě předpodmínění jsou směry $\{p_i\}$ $[(Q_1^{-1}A)^T(Q_1^{-1}A)]$-ortogonální.

Odvoďme tedy metodu GCR s předpodmíněním. Algoritmus uvedený výše platí pro vlnkované hodnoty. Nás však zajímají hodnoty bez vlnek x_{i+1} a r_{i+1}. V tomto případě se proto změní vzorce pro α_i, $\beta_j^{(i)}$ a p_{i+1}. Najděme všechny tyto vzorce.

1. α_i:

$$\tilde{\alpha}_i = \frac{(\tilde{r}_i, \tilde{A}\tilde{p}_i)}{(\tilde{A}\tilde{p}_i, \tilde{A}\tilde{p}_i)} = \frac{(Q_1^{-1}r_i, Q_1^{-1}AQ_2^{-1}\tilde{p}_i)}{(Q_1^{-1}AQ_2^{-1}\tilde{p}_i, Q_1^{-1}AQ_2^{-1}\tilde{p}_i)} = \frac{(Q_1^{-1}r_i, Q_1^{-1}Ap_i)}{(Q_1^{-1}Ap_i, Q_1^{-1}Ap_i)} =: \alpha_i.$$

2. $\beta_j^{(i)}$:

$$\begin{aligned}\tilde{\beta}_j^{(i)} &= -\frac{(\tilde{A}\tilde{r}_{i+1}, \tilde{A}\tilde{p}_j)}{(\tilde{A}\tilde{p}_j, \tilde{A}\tilde{p}_j)} = -\frac{(Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_{i+1}, Q_1^{-1}AQ_2^{-1}Q_2p_j)}{(Q_1^{-1}AQ_2^{-1}Q_2p_j, Q_1^{-1}AQ_2^{-1}Q_2p_j)} = \\ &= -\frac{(Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_{i+1}, Q_1^{-1}Ap_j)}{(Q_1^{-1}Ap_j, Q_1^{-1}Ap_j)} =: \beta_j^{(i)},\end{aligned}$$

vše pro $j \leq i$.

3. p_{i+1}:

$$Q_2 p_{i+1} = \tilde{p}_{i+1} = \tilde{r}_{i+1} + \sum_{j=0}^{i} \beta_j^{(i)} \tilde{p}_j = Q_1^{-1} r_{i+1} + \sum_{j=0}^{i} \beta_j^{(i)} Q_2 p_j,$$

a tudíž

$$p_{i+1} = Q_2^{-1} Q_1^{-1} r_{i+1} + \sum_{j=0}^{i} \beta_j^{(i)} p_j.$$

Dále

$$Q_2 p_0 = \tilde{p}_0 = \tilde{r}_0 = Q_1^{-1} r_0,$$

neboli

$$p_0 = Q_2^{-1} Q_1^{-1} r_0.$$

4. x_{i+1}:

$$Q_2 x_{i+1} = \tilde{x}_{i+1} = \tilde{x}_i + \alpha_i \tilde{p}_i = Q_2 x_i + \alpha_i Q_2 p_i$$

a tedy

$$x_{i+1} = x_i + \alpha_i p_i,$$

což jsme chtěli (stejně jako v nepředpodmíněné metodě).

5. r_{i+1}:

$$r_{i+1} = f - A x_{i+1} = f - A(x_i + \alpha_i p_i) = r_i - \alpha_i A p_i$$

jednoduchým dosazením za x_{i+1}.

Závěr: Označíme $\tilde{A} = Q_1^{-1} A Q_2^{-1}$, $\tilde{f} = Q_1^{-1} f$ a pro každé i pokládáme $\tilde{x}_i = Q_2 x_i$. Po dosazení vyjde $\tilde{r}_i = Q_1^{-1} r_i$ a nakonec definujeme $\tilde{p}_i = Q_2 p_i$.
Pak z algoritmu 1.2 napsaného s vlnovkami nad všemi písmeny obdržíme následující algoritmus s předpodmíněním.

Algoritmus 1.3

Zvolíme x_0.
Spočteme $r_0 = f - A x_0$.
Položíme $p_0 = Q_2^{-1} Q_1^{-1} r_0$.
Pro $i = 0, 1, 2, \ldots$ **provedeme**
$\alpha_i = \frac{(Q_1^{-1} r_i, Q_1^{-1} A p_i)}{(Q_1^{-1} A p_i, Q_1^{-1} A p_i)}$
$x_{i+1} = x_i + \alpha_i p_i$
$r_{i+1} = r_i - \alpha_i A p_i$
$\beta_j^{(i)} = -\frac{(Q_1^{-1} A Q_2^{-1} Q_1^{-1} r_{i+1}, Q_1^{-1} A p_j)}{(Q_1^{-1} A p_j, Q_1^{-1} A p_j)}$ pro $j \leq i$
$p_{i+1} = Q_2^{-1} Q_1^{-1} r_{i+1} + \sum_{j=0}^{i} \beta_j^{(i)} p_j$
`Konec cyklu pro` i.
Konec Algoritmu.

Metodu danou tímto algoritmem nazveme „**zobecněná metoda sdružených residuí (GCR) s předpodmíněním**“.

Nyní odvodíme vztahy mezi vektory generovanými metodou GCR.
Věta 1.1: Nechť $\{x_i\}, \{r_i\}, \{p_i\}$ jsou iterace generované metodou GCR v řešení lineárního systému $Ax = f$ a nechť vektory $\{r_i\}_{i=0}^{t}$ pro nějaké $t \geq 0$ jsou lineárně nezávislé. (To aby se mezi těmito residui nevyskytlo již nulové residuum.) Pak platí:

1. $(Q_1^{-1}Ap_i, Q_1^{-1}Ap_j) = 0$ pro $i \neq j$, což znamená, že směry $\{p_i\}$ jsou lineárně nezávislé.
2. $(Q_1^{-1}r_i, Q_1^{-1}Ap_j) = 0$ pro $i > j$
3. $(Q_1^{-1}r_i, Q_1^{-1}Ap_i) = (Q_1^{-1}r_i, Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_i)$
4. $(Q_1^{-1}r_i, Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_j) = 0$ pro $i > j$
5. $(Q_1^{-1}Ap_i, Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_j) = 0$ pro $i > j$
6. $(Q_1^{-1}Ap_i, Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_i) = (Q_1^{-1}Ap_i, Q_1^{-1}Ap_i)$
7. $(Q_1^{-1}r_j, Q_1^{-1}Ap_i) = (Q_1^{-1}r_0, Q_1^{-1}Ap_i)$ pro $i \geq j$
8. $sp(Q_2p_0, ..., Q_2p_i) = sp(Q_1^{-1}r_0, (Q_1^{-1}AQ_2^{-1})Q_1^{-1}r_0, ..., (Q_1^{-1}AQ_2^{-1})^iQ_1^{-1}r_0) =$
$= sp(Q_1^{-1}r_0, ..., Q_1^{-1}r_i)$
9. $r_i \neq 0 \quad \Rightarrow \quad p_i \neq 0$
10. x_{i+1} minimalizuje $E(w) = \parallel Q_1^{-1}(f - Aw) \parallel_2$ přes afinní prostor $x_0 + sp(p_0, ..., p_i)$

Důkaz:

1. Směry $\{p_i\}$ jsou $[(Q_1^{-1}A)^T(Q_1^{-1}A)]$-ortogonální, platí tedy rovnost. Nyní dokážeme lineární nezávislost směrů $\{p_i\}$.
Sporem: Nechť j je první index takový, že směry $\{p_k\}_{i=1}^{j}$ jsou již lineárně závislé. Pak existuje alespoň jedno $\nu_k \neq 0,\ k = 0, ..., j-1$ tak, že

$$p_j = \sum_{k=0}^{j-1} \nu_k p_k.$$

Teď využijeme $[(Q_1^{-1}A)^T(Q_1^{-1}A)]$-ortogonalitu směrů $\{p_k\}_{k=0}^{j}$ a pro $k = 0, ...j-1$ dostaneme:

$$\begin{aligned} 0 &= (Q_1^{-1}Ap_j, Q_1^{-1}Ap_k) = (Q_1^{-1}A(\sum_{l=0}^{j-1} \nu_l p_l), Q_1^{-1}Ap_k) = \\ &= \sum_{l=0}^{j-1} \nu_l (Q_1^{-1}Ap_l, Q_1^{-1}Ap_k) = \nu_k \cdot (Q_1^{-1}Ap_k, Q_1^{-1}Ap_k) \neq 0 \end{aligned}$$

alespoň pro jedno k a to je spor.
Směry $\{p_i\}$ jsou tedy lineárně nezávislé, a protože Q_2 je regulární matice, tak i směry $\{Q_2p_i\}$ jsou lineárně nezávislé.

2. Indukcí: $i = 1 \ \Rightarrow j = 0$.

$$\begin{aligned} (Q_1^{-1}r_1, Q_1^{-1}Ap_0) &= (Q_1^{-1}(r_0 - \alpha_0 Ap_0), Q_1^{-1}Ap_0) = \\ &= (Q_1^{-1}r_0, Q_1^{-1}Ap_0) - \alpha_0(Q_1^{-1}Ap_0, Q_1^{-1}Ap_0) = \\ &= (Q_1^{-1}r_0, Q_1^{-1}Ap_0) - \frac{(Q_1^{-1}r_0, Q_1^{-1}Ap_0)}{(Q_1^{-1}Ap_0, Q_1^{-1}Ap_0)}(Q_1^{-1}Ap_0, Q_1^{-1}Ap_0) \\ &= 0 \end{aligned}$$

Teď přejdeme od i k $i+1$. Vezmeme rovnost $r_{i+1} = r_i - \alpha_i A p_i$, vynásobíme ji Q_1^{-1} a provedeme skalární součin s $Q_1^{-1} A p_j$:

$$(Q_1^{-1} r_{i+1}, Q_1^{-1} A p_j) = (Q_1^{-1} r_i, Q_1^{-1} A p_j) - \alpha_i (Q_1^{-1} A p_i, Q_1^{-1} A p_j)$$

- je-li $j < i$, pak jsou členy na pravé straně nulové podle indukčního předpokladu a rovnosti 1.
- je-li $j = i$, pak je pravá strana nulová podle definice α_i

Platí-li tedy rovnost 2. pro i, pak platí i pro $i+1$.

3. Vynásobíme rovnost

$$p_i = Q_2^{-1} Q_1^{-1} r_i + \sum_{j=0}^{i-1} \beta_j^{(i-1)} p_j$$

maticí $Q_1^{-1} A$ a provedeme skalární součin s vektorem $Q_1^{-1} r_i$. Dostaneme:

$$\begin{aligned} (Q_1^{-1} r_i, Q_1^{-1} A p_i) &= (Q_1^{-1} r_i, Q_1^{-1} A Q_2^{-1} Q_1^{-1} r_i) + \sum_{j=0}^{i-1} \beta_j^{(i-1)} (Q_1^{-1} r_i, Q_1^{-1} A p_j) = \\ &= (Q_1^{-1} r_i, Q_1^{-1} A Q_2^{-1} Q_1^{-1} r_i), \end{aligned}$$

neboť všechny členy v součtu jsou rovny nule podle rovnosti 2.

4. Přepíšeme znovu rovnost

$$p_j = Q_2^{-1} Q_1^{-1} r_j + \sum_{k=0}^{j-1} \beta_k^{(j-1)} p_k$$

takto:

$$Q_2^{-1} Q_1^{-1} r_j = p_j - \sum_{k=0}^{j-1} \beta_k^{(j-1)} p_k.$$

Vynásobíme maticí $Q_1^{-1} A$ a provedeme skalární součin s vektorem $Q_1^{-1} r_i$, $i > j$. Dostaneme:

$$(Q_1^{-1} r_i, Q_1^{-1} A Q_2^{-1} Q_1^{-1} r_j) = (Q_1^{-1} r_i, Q_1^{-1} A p_j) - \sum_{k=0}^{j-1} \beta_k^{(j-1)} (Q_1^{-1} r_i, Q_1^{-1} A p_k) = 0$$

podle rovnosti 2.

5. Podobně jako v rovnosti 4. dosazením za $Q_2^{-1} Q_1^{-1} r_j$ dostaneme:

$$\begin{aligned} (Q_1^{-1} A p_i, Q_1^{-1} A Q_2^{-1} Q_1^{-1} r_j) &= (Q_1^{-1} A p_i, Q_1^{-1} A p_j) - \\ &- \sum_{k=0}^{j-1} \beta_k^{(j-1)} (Q_1^{-1} A p_i, Q_1^{-1} A p_k) = 0 \end{aligned}$$

pro $i > j$ podle rovnosti 1.

6. Podobně:

$$\begin{aligned}(Q_1^{-1}Ap_i, Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_i) &= (Q_1^{-1}Ap_i, Q_1^{-1}Ap_i) - \\ &- \sum_{k=0}^{i-1}\beta_k^{(i-1)}(Q_1^{-1}Ap_i, Q_1^{-1}Ap_k) = (Q_1^{-1}Ap_i, Q_1^{-1}Ap_i)\end{aligned}$$

podle rovnosti 1.

7. Indukcí podle j, $j \leq i$.
Pro $j = 0$ to platí triviálně.
Přejdeme od j k $j+1$, kde $j+1 \leq i$.
Vezmeme rovnost $r_{j+1} = r_j - \alpha_j Ap_j$:

$$(Q_1^{-1}r_{j+1}, Q_1^{-1}Ap_i) = (Q_1^{-1}r_j, Q_1^{-1}Ap_i) - \alpha_j(Q_1^{-1}Ap_j, Q_1^{-1}Ap_i) = (Q_1^{-1}r_0, Q_1^{-1}Ap_i)$$

podle indukčního předpokladu a rovnosti 1.

8. Indukcí podle i, $i \leq t$.
Uvedené tři prostory jsou stejné, když $i = 0$.
Přejdeme tudíž od i k $i+1$, $i+1 \leq t$.
Platí $sp(Q_2p_0, ..., Q_2p_i) \subset sp(Q_1^{-1}r_0, ..., Q_1^{-1}r_{i+1})$, protože jsme přidali jeden vektor $Q_1^{-1}r_{i+1}$. Z rovnosti

$$p_{i+1} = Q_2^{-1}Q_1^{-1}r_{i+1} + \sum_{j=0}^{i}\beta_j^{(i)}p_j,$$

plyne,že

$$sp(Q_2p_0, ..., Q_2p_{i+1}) \subseteq sp(Q_1^{-1}r_0, ..., Q_1^{-1}r_{i+1}),$$

protože Q_2p_{i+1} je lineární kombinací $Q_1^{-1}r_{i+1}, Q_2p_0, ..., Q_2p_i$. Podle rovnosti 1. jsou vektory $\{Q_2p_j\}_{j=0}^{i+1}$ lineárně nezávislé, neboť vektory $\{p_j\}_{j=0}^{i+1}$ jsou lineárně nezávislé, což plyne z $[(Q_1^{-1}A)^T(Q_1^{-1}A)]$-ortogonality, a Q_2 je regulární matice. Je tedy dimenze

$$i+1 \;\geq\; \dim[sp(Q_1^{-1}r_0, ..., Q_1^{-1}r_{i+1})] \;\geq \dim[sp(Q_2p_0, ..., Q_2p_{i+1})] = i+1,$$

z čehož plyne, že $\{Q_1^{-1}r_j\}_{j=0}^{i+1}$ jsou lineárně nezávislé a

$$sp(Q_2p_0, ..., Q_2p_{i+1}) = sp(Q_1^{-1}r_0, ..., Q_1^{-1}r_{i+1}).$$

Podobně podle rovností

$$r_{i+1} = r_i - \alpha_i Ap_i, \quad p_{i+1} = Q_2^{-1}Q_1^{-1}r_{i+1} + \sum_{j=0}^{i}\beta_j^{(i)}p_j$$

je

$$Q_2p_{i+1} = Q_1^{-1}r_i - \alpha_i Q_1^{-1}Ap_i + \sum_{j=0}^{i}\beta_j^{(i)}Q_2p_j.$$

Podle indukčního předpokladu je

$$sp(Q_1^{-1}r_0, ..., Q_1^{-1}r_i) = sp(Q_1^{-1}r_0, (Q_1^{-1}AQ_2^{-1})Q_1^{-1}r_0, ..., (Q_1^{-1}AQ_2^{-1})^iQ_1^{-1}r_0);$$

$$sp(Q_2p_0, ..., Q_2p_i) = sp(Q_1^{-1}r_0, (Q_1^{-1}AQ_2^{-1})Q_1^{-1}r_0, ..., (Q_1^{-1}AQ_2^{-1})^iQ_1^{-1}r_0),$$

a tedy

$$sp(Q_1^{-1}r_0, ..., Q_1^{-1}r_i) \subseteq sp(Q_1^{-1}r_0, (Q_1^{-1}AQ_2^{-1})Q_1^{-1}r_0, ..., (Q_1^{-1}AQ_2^{-1})^{i+1}Q_1^{-1}r_0);$$

$$sp(Q_2p_0, ..., Q_2p_i) \subseteq sp(Q_1^{-1}r_0, (Q_1^{-1}AQ_2^{-1})Q_1^{-1}r_0, ..., (Q_1^{-1}AQ_2^{-1})^{i+1}Q_1^{-1}r_0).$$

Teď se budeme zabývat vektory $\{Q_1^{-1}Ap_j\}_{j=0}^{i}$.

- $j = 0$:

$$Q_1^{-1}Ap_0 = Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_0,$$

takže

$$sp(Q_1^{-1}Ap_0) \subseteq sp(Q_1^{-1}r_0, (Q_1^{-1}AQ_2^{-1})Q_1^{-1}r_0).$$

- $j = 1$:

$$\begin{aligned}
Q_1^{-1}Ap_1 &= Q_1^{-1}A(Q_2^{-1}Q_1^{-1}r_1 + \beta_0^{(1)}p_0) = Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_1 + \\
&+ \beta_0^{(1)}Q_1^{-1}Ap_0 = \\
&= Q_1^{-1}AQ_2^{-1}Q_1^{-1}(r_0 - \alpha_0Ap_0) + \beta_0^{(1)}Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_0 = \\
&= Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_0 - \alpha_0Q_1^{-1}AQ_2^{-1}Q_1^{-1}Ap_0 + \\
&+ \beta_0^{(1)}Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_0 = \\
&= (1 + \beta_0^{(1)}) \cdot Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_0 - \alpha_0Q_1^{-1}AQ_2^{-1}Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_0 = \\
&= (1 + \beta_0^{(1)}) \cdot Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_0 - \alpha_0(Q_1^{-1}AQ_2^{-1})^2Q_1^{-1}r_0,
\end{aligned}$$

a tedy

$$sp(Q_1^{-1}Ap_0, Q_1^{-1}Ap_1) \subseteq sp(Q_1^{-1}r_0, (Q_1^{-1}AQ_2^{-1})Q_1^{-1}r_0, (Q_1^{-1}AQ_2^{-1})^2Q_1^{-1}r_0).$$

- Analogicky spočteme

$$\begin{aligned}
&sp(Q_1^{-1}Ap_0, ..., Q_1^{-1}Ap_i) \subseteq \\
\subseteq\ &sp(Q_1^{-1}r_0, (Q_1^{-1}AQ_2^{-1})Q_1^{-1}r_0, ..., (Q_1^{-1}AQ_2^{-1})^{i+1}Q_1^{-1}r_0).
\end{aligned}$$

Ze všech těchto úvah plyne, že

$$sp(Q_2p_0, ..., Q_2p_{i+1}) \subseteq sp(Q_1^{-1}r_0, (Q_1^{-1}AQ_2^{-1})Q_1^{-1}r_0, ..., (Q_1^{-1}AQ_2^{-1})^{i+1}Q_1^{-1}r_0).$$

Ale $\{Q_2p_j\}_{j=0}^{i+1}$ jsou lineárně nezávislé, takže oba prostory se rovnají.

9. Využijeme toho, že symetrická část $\tilde{M} = Q_1^{-1}MQ_2^{-1}$ matice $\tilde{A}$ je positivně definitní. Je-li $r_i \neq 0$, pak také $Q_1^{-1}r_i \neq 0$, a podle rovnosti 3. je

$$(Q_1^{-1}r_i, Q_1^{-1}Ap_i) = (Q_1^{-1}r_i, Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_i) = (Q_1^{-1}r_i, Q_1^{-1}MQ_2^{-1}Q_1^{-1}r_i) > 0,$$

takže

$$(Q_1^{-1}r_i, Q_1^{-1}Ap_i) \neq 0$$

a tedy

$$p_i \neq 0,$$

neboť matice Q_1^{-1} a A jsou regulární.

10. Nechť

$$w \in x_0 + sp(p_0, ..., p_i), \quad \text{tj.} \quad w = x_0 + \sum_{j=0}^{i} \alpha_j p_j.$$

Použitím rovnosti 1. upravíme $E(w)^2$.

$$\begin{aligned} E(w)^2 &= \| Q_1^{-1}(f - Aw) \|_2^2 = \| Q_1^{-1}(f - A(x_0 + \sum_{j=0}^{i} \alpha_j p_j)) \|_2^2 = \\ &= \| Q_1^{-1}(f - Ax_0) - \sum_{j=0}^{i} \alpha_j Q_1^{-1} A p_j \|_2^2 = \| Q_1^{-1} r_0 - \sum_{j=0}^{i} \alpha_j Q_1^{-1} A p_j \|_2^2 = \\ &= (Q_1^{-1} r_0, Q_1^{-1} r_0) - 2 \cdot \sum_{j=0}^{i} \alpha_j (Q_1^{-1} r_0, Q_1^{-1} A p_j) + \\ &+ \sum_{j=0}^{i} \alpha_j^2 (Q_1^{-1} A p_j, Q_1^{-1} A p_j). \end{aligned}$$

Provedeme-li Gateauxovu derivaci podle $(\alpha_1, ..., \alpha_i)$, obdržíme soustavu s diagonální maticí, kterou položíme rovnou nule a spočítáme koeficienty α_j.

$$-2 \cdot \sum_{j=0}^{i} (Q_1^{-1} r_0, Q_1^{-1} A p_j) + 2 \cdot \sum_{j=0}^{i} \alpha_j (Q_1^{-1} A p_j, Q_1^{-1} A p_j) = 0$$

$\Longrightarrow$

$$\alpha_j = \frac{(Q_1^{-1} r_0, Q_1^{-1} A p_j)}{(Q_1^{-1} A p_j, Q_1^{-1} A p_j)} = \frac{(Q_1^{-1} r_j, Q_1^{-1} A p_j)}{(Q_1^{-1} A p_j, Q_1^{-1} A p_j)}$$

podle rovnosti 7., což jsou koeficienty v Algoritmu 1.3. Funkcionál $E(w)$ tedy nabývá svého minima pro $w = x_{i+1}$. □

Věta 1.2: Metoda GCR dává přesné řešení soustavy $Ax = f$ po nejvýše N iteracích.
Důkaz:

- Je-li $r_i = 0$ pro nějaké $i \leq N - 1$ pak $Ax_i = f$ a tvrzení platí.
- Je-li $r_i \neq 0$ pro všechna $i \leq N - 1$, pak $p_i \neq 0$ pro všechna $i \leq N - 1$ podle tvrzení 9. Podle rovnosti 1. jsou $\{p_i\}_{i=0}^{N-1}$ lineárně nezávislé, takže

$$sp(p_0, ..., p_{N-1}) = \mathbb{R}^N.$$

Odtud podle tvrzení 10. x_N minimalizuje funkcionál E přes $\mathbb{R}^N$ a tedy x_N je řešení systému $Ax = f$. □

Protože s rostoucím indexem i rostou nároky na uložení vektorů $p_1, ..., p_i$, provádí se restartování metody GCR periodicky po každém k-tém kroku. To znamená, že spočteme k iterací $\{x_i\}_{i=1}^{k}$, které označíme $\{x_{i,j}\}_{i=1}^{k}$ a $(k+1)$-ní iteraci položíme jako novou počáteční hodnotu, tedy $x_{k+1,j} = x_{0,j+1}$. Znovu vypočítáme k iterací a $(k+1)$-ní iteraci budeme uvažovat jako novou počáteční hodnotu. Algoritmus má tuto podobu:

Algoritmus 1.4

Zvolíme $x_{0,0}$.
Spočteme $r_0 = f - Ax_{0,0}$.
Položíme $p_0 = Q_2^{-1}Q_1^{-1}r_0$.
Položíme $l = 0$.
100: Pro $i = 0, 1, 2, ..., k$ provedeme

$\alpha_i = \frac{(Q_1^{-1}r_i, Q_1^{-1}Ap_i)}{(Q_1^{-1}Ap_i, Q_1^{-1}Ap_i)}$
$x_{i+1,l} = x_{i,l} + \alpha_i p_i$
$r_{i+1} = r_i - \alpha_i Ap_i$
$\beta_j^{(i)} = -\frac{(Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_{i+1}, Q_1^{-1}Ap_j)}{(Q_1^{-1}Ap_j, Q_1^{-1}Ap_j)}$ pro $j \leq i$
$p_{i+1} = Q_2^{-1}Q_1^{-1}r_{i+1} + \sum_{j=0}^{i} \beta_j^{(i)} p_j$

`Konec cyklu pro` i.

$x_{0,l+1} = x_{k+1,l}$
$l = l + 1$.
Návrat na 100.

Konec Algoritmu.

Metodu vycházející z tohoto restartovaného algoritmu nazveme „**GCR(k)**“.
Věta 1.3: Nechť symetrická část $\tilde{M}$ matice $\tilde{A} = Q_1^{-1}AQ_2^{-1}$ je positivně definitní. Pak platí

1.

$$\begin{aligned} \lambda_{\min}((Q_1^{-1}AQ_2^{-1})^T(Q_1^{-1}AQ_2^{-1})) &\geq \lambda_{\min}(Q_1^{-1}MQ_2^{-1})^2, \quad \text{neboli} \\ \lambda_{\min}(\tilde{A}^T\tilde{A}) &\geq \lambda_{\min}(\tilde{M})^2 \end{aligned}$$

2.

$$\begin{aligned} \lambda_{\max}((Q_1^{-1}AQ_2^{-1})^T(Q_1^{-1}AQ_2^{-1})) &\leq (\lambda_{\max}(Q_1^{-1}MQ_2^{-1}) + \rho(Q_1^{-1}RQ_2^{-1}))^2, \quad \text{neboli} \\ \lambda_{\max}(\tilde{A}^T\tilde{A}) &\leq [\lambda_{\max}(\tilde{M}) + \rho(\tilde{R})]^2 \end{aligned}$$

3.

$$\begin{aligned} \kappa(Q_1^{-1}AQ_2^{-1}) &\leq \kappa(Q_1^{-1}MQ_2^{-1}) + \frac{\rho(Q_1^{-1}RQ_2^{-1})}{\lambda_{\min}(Q_1^{-1}MQ_2^{-1})}, \quad \text{neboli} \\ \kappa(\tilde{A}) &\leq \kappa(\tilde{M}) + \frac{\rho(\tilde{R})}{\lambda_{\min}(\tilde{M})} \end{aligned}$$

Důkaz:

1. Nechť S je taková symetrická a positivně definitní matice, že $S^2 = \tilde{M}$. Pak

$$\begin{aligned} (\tilde{A}^T\tilde{A}x, x) &= (\tilde{A}x, \tilde{A}x) = ([\tilde{M} - \tilde{R}]x, [\tilde{M} - \tilde{R}]x) = \\ &= (S[S - S^{-1}\tilde{R}]x, S[S - S^{-1}\tilde{R}]x) = \\ &= (\tilde{M}[S - S^{-1}\tilde{R}]x, [S - S^{-1}\tilde{R}]x). \end{aligned}$$

Ale víme, že $\forall y \in \mathbb{R}$ platí $(\tilde{M}y, y) \geq \lambda_{\min}(\tilde{M}) \cdot (y, y)$ a $(\tilde{R}y, y) = 0$. Tedy

$$\begin{aligned}
(\tilde{A}^T\tilde{A}x, x) &\geq \lambda_{\min}(\tilde{M}) \cdot ([S - S^{-1}\tilde{R}]x, [S - S^{-1}\tilde{R}]x) = \\
&= \lambda_{\min}(\tilde{M}) \cdot [(Sx, Sx) - 2(S^{-1}\tilde{R}x, Sx) + (S^{-1}\tilde{R}x, S^{-1}\tilde{R}x)] = \\
&= \lambda_{\min}(\tilde{M}) \cdot [(Sx, Sx) - 2(S^{-T}\tilde{R}x, Sx) + (S^{-1}\tilde{R}x, S^{-1}\tilde{R}x)] = \\
&= \lambda_{\min}(\tilde{M}) \cdot [(Sx, Sx) - 2(\tilde{R}x, x) + (S^{-1}\tilde{R}x, S^{-1}\tilde{R}x)] = \\
&= \lambda_{\min}(\tilde{M}) \cdot [(\tilde{M}x, x) + (S^{-1}\tilde{R}x, S^{-1}\tilde{R}x)] \geq \lambda_{\min}(\tilde{M})^2 \cdot (x, x),
\end{aligned}$$

neboť symetrická a positivně definitní matice $\tilde{M}$ má kladná vlastní čísla a $(S^{-1}\tilde{R}x, S^{-1}\tilde{R}x) \geq 0$. Odtud máme

$$\lambda_{\min}(\tilde{A}^T\tilde{A}) = \min_{x \neq 0} \frac{(\tilde{A}^T\tilde{A}x, x)}{(x, x)} \geq \lambda_{\min}(\tilde{M})^2$$

2. Matice $\tilde{R}$ je antisymetrická a platí $\| \tilde{R} \|_2 = \rho(\tilde{R})$, tedy

$$\lambda_{\max}(\tilde{A}^T\tilde{A}) = \| \tilde{A} \|_2^2 = \| \tilde{M} - \tilde{R} \|_2^2 \leq (\| \tilde{M} \|_2 + \| \tilde{R} \|_2)^2 = (\lambda_{\max}(\tilde{M}) + \rho(\tilde{R}))^2$$

3. Nakonec máme

$$\kappa(\tilde{A}) = \sqrt{\kappa(\tilde{A}^T\tilde{A})} = \sqrt{\frac{\lambda_{\max}(\tilde{A}^T\tilde{A})}{\lambda_{\min}(\tilde{A}^T\tilde{A})}}$$

a dosazením příslušných výrazů 1. a 2. dostaneme požadovanou rovnost. □

Věta 1.4: Nechť $\{r_i\}$ je posloupnost residuí generovaná metodou GCR, nechť symetrická část $\tilde{M}$ matice $Q_1^{-1}AQ_2^{-1} =: \tilde{A}$ je positivně definitní. Nechť $J = T^{-1}Q_1^{-1}AQ_2^{-1}T$ je Jordanův kanonický tvar matice $\tilde{A}$.

1. Pak

$$\begin{aligned}
\| r_i \|_2 &\leq \kappa(Q_1) \cdot \min_{q_i \in P_i} \| q_i(\tilde{A}) \| \cdot \| r_0 \|_2 \leq \\
&\leq \kappa(Q_1) \cdot \left[\sqrt{1 - \frac{\lambda_{\min}(\tilde{M})^2}{\lambda_{\max}(\tilde{A}^T\tilde{A})}}\right]^i \cdot \| r_0 \|_2,
\end{aligned}$$

kde P_i je množina všech polynomů stupně i takových, že $q_i(0) = 1$. Odtud metoda GCR konverguje.

2. Má-li matice $\tilde{A}$ úplnou množinu vlastních vektorů, pak

$$\| r_i \|_2 \leq \kappa(Q_1) \cdot \kappa(T) \cdot m_i \cdot \| r_0 \|_2,$$

kde

$$m_i = \min_{q_i \in P_i} \max_{\lambda \in \sigma(\tilde{A})} |q_i(\lambda)|.$$

3. Kromě toho, je-li $\tilde{A}$ normální, tj. $\tilde{A}\tilde{A}^T = \tilde{A}^T\tilde{A}$, pak

$$\| r_i \|_2 \leq \kappa(Q_1) \cdot m_i \cdot \| r_0 \|_2 .$$

Důkaz: Nejprve poznamenáváme, že

$$\| r_i \| = \| Q_1 Q_1^{-1} r_i \| \leq \| Q_1 \| \cdot \| Q_1^{-1} r_i \| = \| Q_1 \| \cdot \| \tilde{r}_i \|$$

a

$$\kappa(Q_1) = \| Q_1 \| \cdot \| Q_1^{-1} \| .$$

Nyní budeme dokazovat postupně všechny tři části.

1. Podle rovnosti 8. věty 1.1 jsou residua $\{\tilde{r}_i\}$ generovaná metodou GCR tvaru

   $$\tilde{r}_i = q_i(\tilde{A}) \cdot \tilde{r}_0 \quad \text{pro nějaký polynom} \quad q_i \in P_i.$$

   Podle rovnosti 10. (minimalizace normy) je

   $$\| \tilde{r}_i \|_2 = \min_{q_i \in P_i} \| q_i(\tilde{A}) \cdot \tilde{r}_0 \|_2 \leq \min_{q_i \in P_i} \| q_i(\tilde{A}) \|_2 \cdot \| \tilde{r}_0 \|_2,$$

   což dokazuje první nerovnost.
   K důkazu druhé nerovnosti vezmeme libovolný polynom $q_1 \in P_1$, tj.

   $$q_1(z) := 1 + \alpha z \ \in P_1.$$

   Platí

   $$\min_{q_i \in P_i} \| q_i(\tilde{A}) \|_2 \leq \| q_1(\tilde{A})^i \|_2 \leq \| q_1(\tilde{A}) \|_2^i .$$

   Ale

   $$\begin{aligned} \| q_1(\tilde{A}) \|_2^2 &= \max_{\tilde{x} \neq 0} \frac{\Big((I + \alpha\tilde{A})\tilde{x}, (I + \alpha\tilde{A})\tilde{x}\Big)}{(\tilde{x}, \tilde{x})} = \\ &= \max_{\tilde{x} \neq 0} \Big[1 + 2\alpha \cdot \frac{(\tilde{x}, \tilde{A}\tilde{x})}{(\tilde{x}, \tilde{x})} + \alpha^2 \cdot \frac{(\tilde{A}\tilde{x}, \tilde{A}\tilde{x})}{(\tilde{x}, \tilde{x})}\Big]. \end{aligned}$$

   Kromě toho je

   $$\frac{(\tilde{A}\tilde{x}, \tilde{A}\tilde{x})}{(\tilde{x}, \tilde{x})} = \frac{(\tilde{x}, \tilde{A}^T\tilde{A}\tilde{x})}{(\tilde{x}, \tilde{x})} \leq \lambda_{\max}(\tilde{A}^T\tilde{A}).$$

   Protože matice $\tilde{M}$ je positivně definitní, platí

   $$\frac{(\tilde{x}, \tilde{A}\tilde{x})}{(\tilde{x}, \tilde{x})} = \frac{(\tilde{x}, \tilde{M}\tilde{x})}{(\tilde{x}, \tilde{x})} \geq \lambda_{\min}(\tilde{M}) > 0.$$

   Tedy pro $\alpha < 0$ je

   $$\| q_1(\tilde{A}) \|_2^2 \leq \ 1 + 2\alpha \cdot \lambda_{\min}(\tilde{M}) + \alpha^2 \cdot \lambda_{\max}(\tilde{A}^T\tilde{A}).$$

   (Uvažujeme α záporná, protože pro kladná α bychom nedostali, že $\| q_1(\tilde{A}) \|_2 \leq 1$, což potřebujeme ke konvergenci.)
   Tento výraz chceme minimalizovat, proto ho zderivujeme a položíme roven nule:

   $$0 = 2 \cdot \lambda_{\min}(\tilde{M}) + 2\alpha \cdot \lambda_{\max}(\tilde{A}^T\tilde{A}) \ \Rightarrow \ \alpha = -\frac{\lambda_{\min}(\tilde{M})}{\lambda_{\max}(\tilde{A}^T\tilde{A})}$$

a s touto volbou α je

$$\| q_1(\tilde{A}) \| \leq \Big[1 - \frac{\lambda_{\min}(\tilde{M})^2}{\lambda_{\max}(\tilde{A}^T\tilde{A})}\Big]^{\frac{1}{2}}.$$

Dáme-li všechny odhady dohromady, dostaneme

$$\begin{aligned} \| \tilde{r}_i \|_2 &\leq \min_{q_i \in P_i} \| q_i(\tilde{A}) \|_2 \cdot \| \tilde{r}_0 \|_2 \leq \| q_1(\tilde{A}) \|_2^i \cdot \| \tilde{r}_0 \|_2 \leq \\ &\leq \Big[1 - \frac{\lambda_{\min}(\tilde{M})^2}{\lambda_{\max}(\tilde{A}^T\tilde{A})}\Big]^{\frac{i}{2}} \cdot \| \tilde{r}_0 \|_2 . \end{aligned}$$

Podle první části věty 1.3 je $\lambda_{\min}(\tilde{M})^2 \leq \lambda_{\min}(\tilde{A}^T\tilde{A}) \leq \lambda_{\max}(\tilde{A}^T\tilde{A})$, takže celá závorka je < 1. Metoda GCR tedy konverguje, což dokončuje důkaz prvního tvrzení.

2. Přepíšeme rovnost

$$\| \tilde{r}_i \|_2 = \min_{q_i \in P_i} \| q_i(\tilde{A}) \cdot \tilde{r}_0 \|_2$$

jako

$$\begin{aligned} \| \tilde{r}_i \|_2 &= \min_{q_i \in P_i} \| T q_i(J) T^{-1} \cdot \tilde{r}_0 \|_2 \leq \\ &\leq \| T \|_2 \cdot \| T^{-1} \|_2 \cdot \min_{q_i \in P_i} \| q_i(J) \|_2 \cdot \| \tilde{r}_0 \|_2 = \\ &= \kappa(T) \cdot \min_{q_i \in P_i} \| q_i(J) \|_2 \cdot \| \tilde{r}_0 \|_2 . \end{aligned}$$

Protože $\tilde{A}$ má úplnou množinu vlastních vektorů, je J diagonální, takže

$$\min_{q_i \in P_i} \| q_i(J) \|_2 = \min_{q_i \in P_i} \max_{\lambda \in \sigma(\tilde{A})} |q_i(\lambda)| =: m_i.$$

3. Platí Shurova věta: Pro matici $\tilde{A}$ existuje unitární matice U, že $U^H \tilde{A} U = R$, kde R je horní trojúhelníková matice, která má na diagonále vlastní čísla matice $\tilde{A}$, a to $\lambda_1, ..., \lambda_N$.
Protože předpokládáme, že matice $\tilde{A}$ je normální, tj. $\tilde{A}\tilde{A}^T = \tilde{A}^T\tilde{A}$ (nebo obecně $\tilde{A}\tilde{A}^H = \tilde{A}^H\tilde{A}$), plyne odtud, že

$$RR^H = U^H\tilde{A}UU^H\tilde{A}^HU = U^H\tilde{A}\tilde{A}^HU = U^H\tilde{A}^H\tilde{A}U = U^H\tilde{A}^HUU^H\tilde{A}U = R^HR,$$

neboli, že matice R je také normální. Ukážeme, že je navíc diagonální.
Spočítáme prvek matice $R^HR = RR^H$ na pozici (1,1):

$$(R^HR)_{(1,1)} = \sum_{k=1}^{N} r_{1,k}^H r_{k,1} = \overline{\lambda_1} \cdot \lambda_1 = |\lambda_1|^2$$

$$(RR^H)_{(1,1)} = \sum_{k=1}^{N} r_{1,k} r_{k,1}^H = \lambda_1 \cdot \overline{\lambda_1} + \sum_{k=2}^{N} r_{1,k} r_{k,1}^H = |\lambda_1|^2 + \sum_{k=2}^{N} |r_{1,k}|^2$$

Protože $R^HR = RR^H$, vidíme, že

$$\sum_{k=2}^{N} |r_{1,k}|^2 = 0 \quad \Rightarrow \quad r_{1,k} = 0 \ \ \forall k = 2, 3, ..., N.$$

Totéž provedeme s ostatními mimodiagonálními prvky, čímž máme dokázáno, že matice R je skutečně diagonální a na diagonále má vlastní čísla matice $\tilde{A}$. Protože rovněž předpokládáme, že $\tilde{A}$ má úplnou množinu vlastních vektorů, pak platí $R = J$. Odtud dostáváme, že

$$U^H \tilde{A} U = U^{-1} \tilde{A} U = J$$

a tedy matici T lze volit jako unitární matici U, pro kterou platí:

$$\| U \| = \max_{x \neq 0} \frac{\| Ux \|}{\| x \|} = \max_{x \neq 0} \sqrt{\frac{(Ux, Ux)}{(x, x)}} = \max_{x \neq 0} \sqrt{\frac{(x, U^H U x)}{(x, x)}} = \max_{x \neq 0} \sqrt{\frac{(x, x)}{(x, x)}} = 1.$$

Odtud plyne, že

$$\kappa(T) = \kappa(U) = \| U \| \cdot \| U^{-1} \| = \| U \| \cdot \| U^H \| = \| U \|^2 = 1,$$

což dokončuje důkaz. □

Na závěr poznamenáváme, že požadavek, aby symetrická část M matice A byla positivně definitní, je nutný. Metoda sdružených residuí se může zhroutit pro problémy, ve kterých je matice M indefinitní (viz kapitola 4).

1.1.2 Orthomin

Z algoritmu 1.3 je vidět, že k výpočtu $(i+1)$-ní iterace je potřeba právě i směrových vektorů p_j. Existuje však modifikace metody GCR, která je v každém kroku omezena počtem směrových vektorů potřebných k výpočtu nové iterace. Těchto směrových vektorů je právě k, $k \geq 0$. To znamená, že potřebujeme vždy k posledních vektorů p_j. Algoritmus potom vypadá takto:

Algoritmus 1.5

Zvolíme x_0.
Spočteme $r_0 = f - Ax_0$.
Položíme $p_0 = Q_2^{-1} Q_1^{-1} r_0$.
Pro $i = 0, 1, 2, \ldots$ **provedeme**
$\alpha_i = \frac{(Q_1^{-1} r_i, Q_1^{-1} A p_i)}{(Q_1^{-1} A p_i, Q_1^{-1} A p_i)}$
$x_{i+1} = x_i + \alpha_i p_i$
$r_{i+1} = r_i - \alpha_i A p_i$
$\beta_j^{(i)} = -\frac{(Q_1^{-1} A Q_2^{-1} Q_1^{-1} r_{i+1}, Q_1^{-1} A p_j)}{(Q_1^{-1} A p_j, Q_1^{-1} A p_j)}$ pro $j \leq i$
$p_{i+1} = Q_2^{-1} Q_1^{-1} r_{i+1} + \sum_{j=i-k+1}^{i} \beta_j^{(i)} p_j$
`Konec cyklu pro` i.
Konec Algoritmu.

Prvních k směrů $\{p_j\}_{j=0}^{k-1}$ spočteme stejně jako v případě metody GCR podle vzorce

$$p_{j+1} = Q_2^{-1} Q_1^{-1} r_{j+1} + \sum_{l=0}^{j} \beta_l^{(j)} p_l,$$

kde

$$\beta_l^{(j)} = -\frac{(Q_1^{-1} A Q_2^{-1} Q_1^{-1} r_{j+1}, Q_1^{-1} A p_l)}{(Q_1^{-1} A p_l, Q_1^{-1} A p_l)} \text{ pro } l \leq j.$$

Metodu vycházející z tohoto algoritmu nazveme „**Orthomin(k)**“.

Metoda Orthomin(k) se od předchozí metody GCR liší pouze tím, že ve vyjádření p_{i+1} je dolní mez sumy „uříznuta“, tj. nepočítá se od nuly, ale od jistého $i-k+1$ pro nějaké zvolené k. Je-li hodnota $i-k+1<0$, pak se dolní mez bere 0.

Vektory generované metodou Orthomin(k) splňují vztahy podobné jako u metody GCR.

Věta 1.5: Nechť $\{x_i\},\{r_i\},\{p_i\}$ jsou iterace generované metodou Orthomin(k) použité k řešení lineárního systému $Ax=f$ a nechť vektory $\{r_j\}_{j=i-k+1}^{t}$ pro nějaké $t\geq 0$ jsou lineárně nezávislé. Pak platí:

1. $(Q_1^{-1}Ap_i, Q_1^{-1}Ap_j)=0$ pro $j=i-k,...,i-1;\ i\geq k$
2. $(Q_1^{-1}r_i, Q_1^{-1}Ap_j)=0$ pro $j=i-k-1,...,i-1;\ i\geq k+1$
3. $(Q_1^{-1}r_i, Q_1^{-1}Ap_i)=(Q_1^{-1}r_i, Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_i)$
4. $(Q_1^{-1}r_i, Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_{i-1})=0$
5. $(Q_1^{-1}Ap_i, Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_i)=(Q_1^{-1}Ap_i, Q_1^{-1}Ap_i)$
6. $(Q_1^{-1}r_j, Q_1^{-1}Ap_i)=(Q_1^{-1}r_{i-k}, Q_1^{-1}Ap_i)$ pro $j=i-k,...,i;\ i\geq k$
7. $r_i\neq 0 \quad\Rightarrow\quad p_i\neq 0$
8. pro $i\geq k,\ x_{i+1}$ minimalizuje $E(w)=\parallel Q_1^{-1}(f-Aw)\parallel_2$ přes afinní prostor $x_{i-k}+sp(p_{i-k},...,p_i)$

Důkaz: Stejný jako pro metodu GCR. □

Pokud je matice A symetrická a positivně definitní, pak jsou obě metody jak GCR, tak Orthomin(k) pro $k\geq 1$ matematicky ekvivalentní s metodou sdružených residuí.

Ve speciálním případě $k=0$ jsou metody GCR(k) a Orthomin(k) identické a směrové vektory se v tomto případě spočtou velmi snadno podle vzorce $p_{i+1}=r_{i+1}$, protože se ve vzorci pro p_{i+1} objeví suma, jejíž horní mez je i a dolní mez $i+1$. Tedy celá suma vymizí. Celý algoritmus pak vypadá takto:

Algoritmus 1.6

Zvolíme x_0.
Spočteme $r_0=f-Ax_0$.
Položíme $p_0=Q_2^{-1}Q_1^{-1}r_0$.
Pro $i=0,1,2,...$ **provedeme**
$\alpha_i=\frac{(Q_1^{-1}r_i,Q_1^{-1}Ap_i)}{(Q_1^{-1}Ap_i,Q_1^{-1}Ap_i)}$
$x_{i+1}=x_i+\alpha_i p_i$
$r_{i+1}=r_i-\alpha_i Ap_i$
$p_{i+1}=Q_2^{-1}Q_1^{-1}r_{i+1}$
`Konec cyklu pro` i.
Konec Algoritmu.

Metodu danou tímto algoritmem nazveme „**metoda minimálních residuí (MR)** *(Minimum Residual method)*“.
Je velmi jednoduchá, proto ji uvažujeme odděleně.

Samozřejmě lze též uvažovat dohromady metody Orthomin(k) a GCR(k). To znamená, že omezíme počet směrových vektorů na k_1 pro výpočet následující iterace a po každém k_2-tém kroku restartujeme. Tato metoda má smysl jen pro případ $k_1 < k_2$, protože pro případ $k_1 \geq k_2$ máme obyčejnou restartovanou metodu GCR(k_2). Pro úplnost uvádíme i tento algoritmus.

Algoritmus 1.7

Zvolíme $x_{0,0}$.
Spočteme $r_0 = f - Ax_{0,0}$.
Položíme $p_0 = Q_2^{-1}Q_1^{-1}r_0$.
Položíme $l = 0$.
100: Pro $i = 0, 1, 2, ..., k_2$ provedeme
$\alpha_i = \frac{(Q_1^{-1}r_i, Q_1^{-1}Ap_i)}{(Q_1^{-1}Ap_i, Q_1^{-1}Ap_i)}$
$x_{i+1,l} = x_{i,l} + \alpha_i p_i$
$r_{i+1} = r_i - \alpha_i Ap_i$
$\beta_j^{(i)} = -\frac{(Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_{i+1}, Q_1^{-1}Ap_j)}{(Q_1^{-1}Ap_j, Q_1^{-1}Ap_j)}$ pro $j \leq i$
$p_{i+1} = Q_2^{-1}Q_1^{-1}r_{i+1} + \sum_{j=i-k_1+1}^{i} \beta_j^{(i)} p_j$
`Konec cyklu pro` i.
$x_{0,l+1} = x_{k_2+1,l}$
$l = l + 1$.
Návrat na 100.
Konec Algoritmu.

Metoda vycházející z tohoto algoritmu nemá speciální název, je pouze spojením dvou předchozích metod. Lze ji nazvat např. „**kombinace Orthomin(k_1) a GCR(k_2)**“.

Následující výsledky (věty) se stejně dobře aplikují na všechny čtyři metody, které jsme zde uvedli, proto je uvádíme najednou pro všechny metody.

Věta 1.6: Směrové vektory $\{p_i\}$ a residua $\{r_i\}$ generované metodami GCR, GCR(k), Orthomin(k) a MR splňují:

$$(Q_1^{-1}Ap_i, Q_1^{-1}Ap_i) \ \leq \ (Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_i, Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_i),$$

neboli platí

$$(\tilde{A}\tilde{p}_i, \tilde{A}p_i) \ \leq \ (\tilde{A}\tilde{r}_i, \tilde{A}\tilde{r}_i).$$

Důkaz: Směrové vektory $\{\tilde{p}_i\}$ jsou dány takto:

$$\tilde{p}_i = \tilde{r}_i + \sum_j \beta_j^{(i-1)} \tilde{p}_j,$$

kde

$$\beta_j^{(i-1)} = -\frac{(\tilde{A}\tilde{r}_i, \tilde{A}\tilde{p}_j)}{(\tilde{A}\tilde{p}_j, \tilde{A}\tilde{p}_j)}$$

a meze sumy jsou podle toho, o jakou metodu se jedná.
Využijeme-li $(\tilde{A}^T\tilde{A})$-ortogonalitu směrů $\{\tilde{p}_i\}$, dostaneme:

$$(Q_1^{-1}Ap_i, Q_1^{-1}Ap_i) \quad = \quad (\tilde{A}\tilde{p}_i, \tilde{A}\tilde{p}_i) = (\tilde{A}[\tilde{r}_i + \sum_j \beta_j^{(i-1)}\tilde{p}_j], \tilde{A}[\tilde{r}_i + \sum_j \beta_j^{(i-1)}\tilde{p}_j]) =$$

$$
\begin{aligned}
&= (\tilde{A}\tilde{r}_i, \tilde{A}\tilde{r}_i) + 2\sum_j \beta_j^{(i-1)}(\tilde{A}\tilde{r}_i, \tilde{A}\tilde{p}_j) + \\
&+ \sum_j \sum_k \beta_j^{(i-1)}\beta_k^{(i-1)}(\tilde{A}\tilde{p}_j, \tilde{A}\tilde{p}_k) = \\
&= (\tilde{A}\tilde{r}_i, \tilde{A}\tilde{r}_i) + 2\sum_j \beta_j^{(i-1)}(\tilde{A}\tilde{r}_i, \tilde{A}\tilde{p}_j) + \sum_j (\beta_j^{(i-1)})^2(\tilde{A}\tilde{p}_j, \tilde{A}\tilde{p}_j) = \\
&= (\tilde{A}\tilde{r}_i, \tilde{A}\tilde{r}_i) - 2\sum_j \frac{(\tilde{A}\tilde{r}_i, \tilde{A}\tilde{p}_j)^2}{(\tilde{A}\tilde{p}_j, \tilde{A}\tilde{p}_j)} + \sum_j \frac{(\tilde{A}\tilde{r}_i, \tilde{A}\tilde{p}_j)^2}{(\tilde{A}\tilde{p}_j, \tilde{A}\tilde{p}_j)} = \\
&= (\tilde{A}\tilde{r}_i, \tilde{A}\tilde{r}_i) - \sum_j \frac{(\tilde{A}\tilde{r}_i, \tilde{A}\tilde{p}_j)^2}{(\tilde{A}\tilde{p}_j, \tilde{A}\tilde{p}_j)} \ \leq \\
&\leq (\tilde{A}\tilde{r}_i, \tilde{A}\tilde{r}_i) = (Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_i, Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_i),
\end{aligned}
$$

neboť v sumě se sčítají nezáporné členy. □

Věta 1.7: Pro každé reálné $x \neq 0$ platí:

$$
\frac{(Q_2x, Q_1^{-1}Ax)}{(Q_1^{-1}Ax, Q_1^{-1}Ax)} \ \geq \ \frac{\lambda_{\min}(Q_1^{-1}MQ_2^{-1})}{\lambda_{\min}(Q_1^{-1}MQ_2^{-1}) \cdot \lambda_{\max}(Q_1^{-1}MQ_2^{-1}) + \rho(Q_1^{-1}RQ_2^{-1})^2},
$$

neboli

$$
\frac{(\tilde{x}, \tilde{A}\tilde{x})}{(\tilde{A}\tilde{x}, \tilde{A}\tilde{x})} \ \geq \ \frac{\lambda_{\min}(\tilde{M})}{\lambda_{\min}(\tilde{M}) \cdot \lambda_{\max}(\tilde{M}) + \rho(\tilde{R})^2}.
$$

Důkaz: Nechť $y = \tilde{A}\tilde{x}$. Pak

$$
\begin{aligned}
\frac{(\tilde{x}, \tilde{A}\tilde{x})}{(\tilde{A}\tilde{x}, \tilde{A}\tilde{x})} &= \frac{(y, \tilde{A}^{-1}y)}{(y,y)} = \frac{1}{2} \cdot \frac{(y, \tilde{A}^{-1} + (\tilde{A}^{-1})^T y)}{(y,y)} \ \geq \\
&\geq \lambda_{\min}\Big(\frac{\tilde{A}^{-1} + (\tilde{A}^{-1})^T}{2}\Big).
\end{aligned}
$$

To nám stačí k odhadu $\lambda_{\min}(\frac{\tilde{A}^{-1}+(\tilde{A}^{-1})^T}{2})$.
Uvažujme identitu

$$
X^{-1} + Y^{-1} = [Y(X+Y)^{-1}X]^{-1},
$$

která platí pro každé dvě regulární matice X, Y za předpokladu, že matice $X+Y$ je také regulární.
Pro

$$
X = 2 \cdot \tilde{A}, \quad Y = 2 \cdot \tilde{A}^T
$$

máme:

$$
\begin{aligned}
\frac{\tilde{A}^{-1} + (\tilde{A}^{-1})^T}{2} &= [2 \cdot \tilde{A}^T \cdot (2 \cdot \tilde{A} + 2 \cdot \tilde{A}^T)^{-1} \cdot 2 \cdot \tilde{A}]^{-1} = \\
&= [2 \cdot \tilde{A}^T \cdot (4 \cdot \tilde{M})^{-1} \cdot 2 \cdot \tilde{A}]^{-1} = \\
&= [(\tilde{M} - \tilde{R}^T) \cdot \tilde{M}^{-1} \cdot (\tilde{M} - \tilde{R})]^{-1} = \\
&= [\tilde{M} + \tilde{R}^T\tilde{M}^{-1}\tilde{R}]^{-1}.
\end{aligned}
$$

Pro každé $\tilde{x} \neq 0$ je

$$
(\tilde{x}, [\tilde{M} + \tilde{R}^T\tilde{M}^{-1}\tilde{R}]\tilde{x}) = (\tilde{x}, \tilde{M}\tilde{x}) + (\tilde{R}\tilde{x}, [\tilde{M}^{-1}\tilde{R}]\tilde{x}) > 0,
$$

takže matice

$$\tilde{M} + \tilde{R}^T \tilde{M}^{-1} \tilde{R}$$

je positivně definitní.
Proto je také matice

$$\frac{\tilde{A}^{-1} + (\tilde{A}^{-1})^T}{2}$$

positivně definitní a

$$\lambda_{\min}\Big(\frac{\tilde{A}^{-1} + (\tilde{A}^{-1})^T}{2}\Big) = \frac{1}{\lambda_{\max}(\tilde{M} + \tilde{R}^T \tilde{M}^{-1} \tilde{R})}$$

Ale

$$\begin{aligned}
\lambda_{\max}(\tilde{M} + \tilde{R}^T \tilde{M}^{-1} \tilde{R}) &= \max_{\tilde{x} \neq 0}\Big[\frac{(\tilde{x}, \tilde{M}\tilde{x})}{(\tilde{x}, \tilde{x})} + \frac{(\tilde{x}, \tilde{R}^T \tilde{M}^{-1} \tilde{R}\tilde{x})}{(\tilde{x}, \tilde{x})}\Big] \leq \\
&\leq \lambda_{\max}(\tilde{M}) + \max_{\tilde{x}, \tilde{R}\tilde{x} \neq 0}\Big[\frac{(\tilde{R}\tilde{x}, \tilde{M}^{-1}\tilde{R}\tilde{x})}{(\tilde{R}\tilde{x}, \tilde{R}\tilde{x})} \cdot \frac{(\tilde{R}\tilde{x}, \tilde{R}\tilde{x})}{(\tilde{x}, \tilde{x})}\Big] \leq \\
&\leq \lambda_{\max}(\tilde{M}) + \lambda_{\max}(\tilde{M}^{-1}) \cdot \| \tilde{R}^T \tilde{R} \|_2 = \lambda_{\max}(\tilde{M}) + \frac{\rho(\tilde{R})^2}{\lambda_{\min}(\tilde{M})}
\end{aligned}$$

Dáme-li všechny úpravy dohromady, dostaneme výsledek

$$\lambda_{\min}\Big(\frac{\tilde{A}^{-1} + (\tilde{A}^{-1})^T}{2}\Big) \geq \frac{1}{\lambda_{\max}(\tilde{M}) + \frac{\rho(\tilde{R})^2}{\lambda_{\min}(\tilde{M})}}$$

Shrneme-li tento odhad s odhadem pro $\lambda_{\min}\Big(\frac{\tilde{A}^{-1} + (\tilde{A}^{-1})^T}{2}\Big)$ ze začátku důkazu, dostaneme:

$$\frac{(\tilde{x}, \tilde{A}\tilde{x})}{(\tilde{A}\tilde{x}, \tilde{A}\tilde{x})} \geq \lambda_{\min}\Big(\frac{\tilde{A}^{-1} + (\tilde{A}^{-1})^T}{2}\Big) \geq \frac{\lambda_{\min}(\tilde{M})}{\lambda_{\min}(\tilde{M}) \cdot \lambda_{\max}(\tilde{M}) + \rho(\tilde{R})^2},$$

což dokončuje důkaz. □

Věta 1.8: Nechť $\{r_i\}$ je posloupnost residuí generovaná metodami GCR, GCR(k), Orthomin(k) nebo MR. Nechť $\tilde{M}$, resp. $\tilde{R}$ je symetrická, resp. antisymetrická část matice $Q_1^{-1} A Q_2^{-1} =: \tilde{A}$. Pak platí:

1.
$$\| r_i \|_2 \leq \kappa(Q_1) \cdot \Big[1 - \frac{\lambda_{\min}(\tilde{M})^2}{\lambda_{\max}(\tilde{A}^T \tilde{A})}\Big]^{\frac{i}{2}} \cdot \| r_0 \|_2 =: o1$$

2.
$$\| r_i \|_2 \leq \kappa(Q_1) \cdot \Big[1 - \frac{\lambda_{\min}(\tilde{M})^2}{\lambda_{\min}(\tilde{M}) \cdot \lambda_{\max}(\tilde{M}) + \rho(\tilde{R})^2}\Big]^{\frac{i}{2}} \cdot \| r_0 \|_2 =: o2$$

avšak oba odhady nejsou srovnatelné, tj. neznáme vztah mezi $o1$ a $o2$. Lze jen zkráceně napsat, že

$$\| r_i \|_2 \leq \min\{o1, o2\},$$

čímž získáváme konvergenci.
Důkaz: Platí:

$$\| r_i \| = \| Q_1 Q_1^{-1} r_i \| \leq \| Q_1 \| \cdot \| Q_1^{-1} r_i \| = \| Q_1 \| \cdot \| \tilde{r}_i \|$$

a

$$\kappa(Q_1) = \| Q_1 \| \cdot \| Q_1^{-1} \| .$$

Nyní dokážeme oba odhady zvlášť.

1. První odhad je totožný odhadu v první části věty 1.4. Dokážeme ho trochu jinak. Uvažujme rovnost $\tilde{r}_{i+1} = \tilde{r}_i - \alpha_i \tilde{A}\tilde{p}_i$, kde $\alpha_i = \frac{(\tilde{r}_i, \tilde{A}\tilde{p}_i)}{(\tilde{A}\tilde{p}_i, \tilde{A}\tilde{p}_i)}$.
   Pak

$$\begin{aligned} \| \tilde{r}_{i+1} \|_2^2 &= (\tilde{r}_{i+1}, \tilde{r}_{i+1}) = (\tilde{r}_i - \alpha_i \tilde{A}\tilde{p}_i, \tilde{r}_i - \alpha_i \tilde{A}\tilde{p}_i) = \\ &= (\tilde{r}_i, \tilde{r}_i) - 2\alpha_i (\tilde{r}_i, \tilde{A}\tilde{p}_i) + \alpha_i^2 (\tilde{A}\tilde{p}_i, \tilde{A}\tilde{p}_i) = \\ &= \| \tilde{r}_i \|_2^2 - 2 \cdot \frac{(\tilde{r}_i, \tilde{A}\tilde{p}_i)^2}{(\tilde{A}\tilde{p}_i, \tilde{A}\tilde{p}_i)} + \frac{(\tilde{r}_i, \tilde{A}\tilde{p}_i)^2}{(\tilde{A}\tilde{p}_i, \tilde{A}\tilde{p}_i)} = \\ &= \| \tilde{r}_i \|_2^2 - \frac{(\tilde{r}_i, \tilde{A}\tilde{p}_i)^2}{(\tilde{A}\tilde{p}_i, \tilde{A}\tilde{p}_i)} \end{aligned}$$

   Tedy po vydělení $\| \tilde{r}_i \|_2^2$ dostaneme

$$\frac{\| \tilde{r}_{i+1} \|_2^2}{\| \tilde{r}_i \|_2^2} = 1 - \frac{(\tilde{r}_i, \tilde{A}\tilde{p}_i)}{(\tilde{r}_i, \tilde{r}_i)} \cdot \frac{(\tilde{r}_i, \tilde{A}\tilde{p}_i)}{(\tilde{A}\tilde{p}_i, \tilde{A}\tilde{p}_i)} \leq 1 - \frac{(\tilde{r}_i, \tilde{A}\tilde{r}_i)}{(\tilde{r}_i, \tilde{r}_i)} \cdot \frac{(\tilde{r}_i, \tilde{A}\tilde{r}_i)}{(\tilde{A}\tilde{r}_i, \tilde{A}\tilde{r}_i)}$$

   podle věty 1.6 a rovnosti 3. věty 1.5.
   Ale

$$\frac{(\tilde{r}_i, \tilde{A}\tilde{r}_i)}{(\tilde{r}_i, \tilde{r}_i)} \geq \lambda_{\min}(\tilde{M})$$

   a

$$\frac{(\tilde{r}_i, \tilde{A}\tilde{r}_i)}{(\tilde{A}\tilde{r}_i, \tilde{A}\tilde{r}_i)} = \frac{(\tilde{r}_i, \tilde{r}_i)}{(\tilde{r}_i, \tilde{A}^T\tilde{A}\tilde{r}_i)} \cdot \frac{(\tilde{r}_i, \tilde{A}\tilde{r}_i)}{(\tilde{r}_i, \tilde{r}_i)} \geq \frac{\lambda_{\min}(\tilde{M})}{\lambda_{\max}(\tilde{A}^T\tilde{A})},$$

   takže

$$\| \tilde{r}_{i+1} \|_2 \leq \left[1 - \frac{\lambda_{\min}(\tilde{M})^2}{\lambda_{\max}(\tilde{A}^T\tilde{A})}\right]^{\frac{1}{2}} \cdot \| \tilde{r}_i \|_2,$$

   což dokazuje první odhad. Výraz v závorce je < 1 podle první části věty 1.3 a tím máme konvergenci.

2. Podle věty 1.7 je

$$\frac{(\tilde{r}_i, \tilde{A}\tilde{r}_i)}{(\tilde{A}\tilde{r}_i, \tilde{A}\tilde{r}_i)} \geq \frac{\lambda_{\min}(\tilde{M})}{\lambda_{\min}(\tilde{M}) \cdot \lambda_{\max}(\tilde{M}) + \rho(\tilde{R})^2},$$

   takže

$$\| \tilde{r}_{i+1} \|_2 \leq \left[1 - \frac{\lambda_{\min}(\tilde{M})^2}{\lambda_{\min}(\tilde{M}) \cdot \lambda_{\max}(\tilde{M}) + \rho(\tilde{R})^2}\right]^{\frac{1}{2}} \cdot \| \tilde{r}_i \|_2,$$

což dokazuje druhý odhad. K důkazu konvergence stačí ukázat, že výraz v závorce je < 1, neboli, že

$$\lambda_{\min}(\tilde{M})^2 \leq \lambda_{\min}(\tilde{M}) \cdot \lambda_{\max}(\tilde{M}) + \rho(\tilde{R})^2.$$

Ale tato nerovnost platí právě tehdy, když

$$\lambda_{\min}(\tilde{M}) \leq \lambda_{\max}(\tilde{M}) + \frac{\rho(\tilde{R})^2}{\lambda_{\min}(\tilde{M})},$$

což platí vždy, neboť zlomek vpravo je kladný. □

Lemma 1.1: Je-li symetrická část $\tilde{M}$ matice $\tilde{A}$ jednotková matice I, tedy $\tilde{A} = I - \tilde{R}$, kde $\tilde{R}$ je antisymetrická část matice $\tilde{A}$, pak metoda Orthomin(1) je ekvivalentní s metodou GCR.

Důkaz: Stačí ukázat, že koeficienty $\beta_j^{(i)} = 0$ pro $j \leq i-1$ (u metody GCR). Vezměme čitatel:

$$\begin{aligned}(Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_{i+1}, Q_1^{-1}Ap_j) = (\tilde{A}\tilde{r}_{i+1}, \tilde{A}\tilde{p}_j) &= ([I-\tilde{R}]\tilde{r}_{i+1}, \tilde{A}\tilde{p}_j) = \\ &= (\tilde{r}_{i+1}, \tilde{A}\tilde{p}_j) - (\tilde{R}\tilde{r}_{i+1}, \tilde{A}\tilde{p}_j).\end{aligned}$$

Podle rovnosti 2. věty 1.1 je

$$(\tilde{r}_{i+1}, \tilde{A}\tilde{p}_j) = 0 = -(\tilde{r}_{i+1}, \tilde{A}\tilde{p}_j).$$

Teď využijeme antisymetrii matice $\tilde{R}$:

$$\begin{aligned}(\tilde{A}\tilde{r}_{i+1}, \tilde{A}\tilde{p}_j) &= (\tilde{r}_{i+1}, \tilde{A}\tilde{p}_j) - (\tilde{R}\tilde{r}_{i+1}, \tilde{A}\tilde{p}_j) = -(\tilde{r}_{i+1}, \tilde{A}\tilde{p}_j) - (\tilde{r}_{i+1}, \tilde{R}^T\tilde{A}\tilde{p}_j) = \\ &= -(\tilde{r}_{i+1}, \tilde{A}\tilde{p}_j) + (\tilde{r}_{i+1}, \tilde{R}\tilde{A}\tilde{p}_j) = (\tilde{r}_{i+1}, [\tilde{R}\tilde{A} - \tilde{A}]\tilde{p}_j) = \\ &= (\tilde{r}_{i+1}, [\tilde{R} - I]\tilde{A}\tilde{p}_j) = -(\tilde{r}_{i+1}, \tilde{A}^2\tilde{p}_j).\end{aligned}$$

Dále víme, že

$$\tilde{r}_{j+1} = \tilde{r}_j - \alpha_j\tilde{A}\tilde{p}_j \;\Rightarrow\; \tilde{A}\tilde{p}_j = \frac{1}{\alpha_j}(\tilde{r}_j - \tilde{r}_{j+1}).$$

Dosadíme:

$$\begin{aligned}(\tilde{A}\tilde{r}_{i+1}, \tilde{A}\tilde{p}_j) &= -(\tilde{r}_{i+1}, \tilde{A}^2\tilde{p}_j) = -(\tilde{r}_{i+1}, \tilde{A}[\frac{1}{\alpha_j}(\tilde{r}_j - \tilde{r}_{j+1})]) = \\ &= -\frac{1}{\alpha_j} \cdot [(\tilde{r}_{i+1}, \tilde{A}\tilde{r}_j) + (\tilde{r}_{i+1}, \tilde{A}\tilde{r}_{j+1})] = 0 \text{ pro } j \leq i-1\end{aligned}$$

podle 4. rovnosti věty 1.1. □

1.1.3 Axelssonovo zobecnění

Následuje jiné zobecnění metody sdružených residuí pro řešení lineárního systému $Ax = f$, kde A je regulární nesymetrická matice řádu N. Stejně jako předchozí metody i tyto metody minimalizují normu residua přes nějaký Krylovův podprostor generovaný maticí A. Axelssonovo zobecnění (viz [Axel 1]) metody sdružených residuí je aplikovatelné na systémy, ve kterých má matice koeficientů A symetrickou positivně definitní část M. Následující algoritmus je Axelssonovo zobecnění metody sdružených residuí.

Algoritmus 1.8

Zvolíme x_0.
Spočteme $r_0 = f - Ax_0$.
Položíme $p_0 = Q_2^{-1}Q_1^{-1}r_0$.
Pro $i = 0, 1, 2, ...$ **provedeme**
$x_{i+1} = x_i + \sum_{j=0}^{i} \alpha_j^{(i)} p_j$
kde $\{\alpha_j^{(i)}\}_{j=0}^{i}$ minimalizují $\| Q_1^{-1}r_{i+1} \|_2$
$r_{i+1} = r_i - \sum_{j=0}^{i} \alpha_j^{(i)} Ap_j$
$\beta_i = -\frac{(Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_{i+1}, Q_1^{-1}Ap_i)}{(Q_1^{-1}Ap_i, Q_1^{-1}Ap_i)}$
$p_{i+1} = Q_2^{-1}Q_1^{-1}r_{i+1} + \beta_i p_i$
`Konec cyklu pro` i.
Konec Algoritmu.

Výpočet délky kroků $\{\alpha_j^{(i)}\}_{j=0}^{i}$ vyžaduje řešení problému nejmenších čtverců: minimalizovat

$$\| B^{(i)}\underline{\alpha}^{(i)} - Q_1^{-1}r_i \|_2,$$

pro

$$\underline{\alpha}^{(i)} = (\alpha_0, ..., \alpha_i)^T,$$

kde

$$B^{(i)} := (Q_1^{-1}Ap_0, ..., Q_1^{-1}Ap_i).$$

Výběr $\{\alpha_j^{(i-1)}\}_{j=0}^{i-1}$ žádá, aby bylo splněno

$$\| Q_1^{-1}r_i \|_2 = \min_{q_i \in P_i} \| q_i(Q_1^{-1}AQ_2^{-1})r_0 \|_2,$$

neboť i zde stále platí rovnost

$$\begin{aligned} sp(Q_2p_0, ..., Q_2p_i) &= sp(Q_1^{-1}r_0, (Q_1^{-1}AQ_2^{-1})Q_1^{-1}r_0, ..., (Q_1^{-1}AQ_2^{-1})^iQ_1^{-1}r_0) = \\ &= sp(Q_1^{-1}r_0, ..., Q_1^{-1}r_i) \end{aligned}$$

a tudíž

$$r_i = q_i(A) \cdot r_0 \quad \text{pro nějaký polynom} \quad q_i \in P_i, \quad \text{že} \quad q_i(0) = 1,$$

takže tato metoda je ekvivalentní se zobecněnou metodou sdružených residuí GCR.
Metodu vycházející z tohoto algoritmu nazveme „**zobecněná metoda sdružených residuí ve smyslu nejmenších čtverců (LSGCR)** *(Least Squares Generalized Conjugate Residual method)*“.

Restartujeme-li po každých k krocích, pak výsledná metoda je ekvivalentní s metodou GCR(k).

Lze též uvažovat zkrácenou verzi předchozího algoritmu, kde k výpočtu následující iterace potřebujeme pouze k posledních směrových vektorů $\{p_j\}_{j=i-k+1}^{i}$, podobně jako v případě Orthomin(k). Algoritmus má tuto podobu:

Algoritmus 1.9

Zvolíme x_0.
Spočteme $r_0 = f - Ax_0$.
Položíme $p_0 = Q_2^{-1}Q_1^{-1}r_0$.

Pro $i = 0, 1, 2, ...$ **provedeme**
$x_{i+1} = x_i + \sum_{j=i-k+1}^{i} \alpha_j^{(i)} p_j$
kde $\{\alpha_j^{(i)}\}_{j=0}^{i}$ minimalizují $\| Q_1^{-1} r_{i+1} \|_2$
$r_{i+1} = r_i - \sum_{j=i-k+1}^{i} \alpha_j^{(i)} A p_j$
$\beta_i = -\frac{(Q_1^{-1} A Q_2^{-1} Q_1^{-1} r_{i+1}, Q_1^{-1} A p_i)}{(Q_1^{-1} A p_i, Q_1^{-1} A p_i)}$
$p_{i+1} = Q_2^{-1} Q_1^{-1} r_{i+1} + \beta_i p_i$
`Konec cyklu pro` i.
Konec Algoritmu.

Délky kroků $\{\alpha_j^{(i)}\}_{j=i-k+1}^{i}$, kde $k \geq 1$, jsou voleny tak, aby minimalizovali $\| Q_1^{-1} r_{i+1} \|_2$. To vyžaduje řešení tohoto problému nejmenších čtverců:
minimalizovat

$$\| B^{(i)} \underline{\alpha}^{(i)} - Q_1^{-1} r_i \|_2,$$

pro

$$\underline{\alpha}^{(i)} = (\alpha_{i-k+1}, ..., \alpha_i)^T,$$

kde

$$B^{(i)} := (Q_1^{-1} A p_{i-k+1}, ..., Q_1^{-1} A p_i).$$

Metodu danou tímto algoritmem nazveme „**Axel(k)**“.
Lemma 1.2: Pro $k = 1$ je metoda Axel(1) ekvivalentní s metodou Orthomin(1).
Důkaz: Dosadíme-li do vzorců pro x_{i+1} a r_{i+1} za $k = 1$, dostaneme výrazy

$$x_{i+1} = x_i + \alpha_i p_i, \quad r_{i+1} = r_i - \alpha_i A p_i,$$

tedy stejné jako v případě Orthomin(1).
Problém nejmenších čtverců je jednoduchý:
minimalizovat

$$\| Q_1^{-1} A p_i \alpha_i - Q_1^{-1} r_i \|_2 = \| Q_1^{-1} r_{i+1} \|_2 .$$

Jestliže rozepíšeme normu vlevo do skalárního součinu, upravíme, provedeme derivaci a spočítáme koeficienty α_i, získáme stejný vzorec jako v případě Orthomin(1).
Tím jsme ukázali, že oba algoritmy jsou totožné. □

Následuje množina vztahů podobná předchozím metodám.
Věta 1.9: Nechť $\{x_i\}, \{r_i\}$ a $\{p_i\}$ jsou iterace generované metodou Axel(k) v řešení lineárního systému $Ax = f$ a nechť vektory $\{r_j\}_{j=i-k+1}^{t}$ pro nějaké $t \geq 0$ jsou lineárně nezávislé. Pak platí:

1. $(Q_1^{-1} A p_i, Q_1^{-1} A p_{i-1}) = 0$
2. $(Q_1^{-1} r_i, Q_1^{-1} A p_j) = 0$ pro $j = i - k, ..., i - 1;\ i \geq k$
3. $(Q_1^{-1} r_i, Q_1^{-1} A p_i) = (Q_1^{-1} r_i, Q_1^{-1} A Q_2^{-1} Q_1^{-1} r_i)$
4. $(Q_1^{-1} r_i, Q_1^{-1} A Q_2^{-1} Q_1^{-1} r_j) = 0$ pro $j = i - k + 1, ..., i - 1;\ i \geq k$
5. $r_i \neq 0 \quad \Rightarrow \quad p_i \neq 0$
6. pro $i \geq k,\ x_{i+1}$ minimalizuje $E(w) = \| Q_1^{-1}(f - Aw) \|_2$ přes afinní prostor $x_i + sp(p_{i-k+1}, ..., p_i)$

7. $r_i \neq 0 \quad \Rightarrow \quad \{p_j\}_{j=i-k+1}^{i}$ jsou lineárně nezávislé, takže matice $B^{(i)}$ má plný řád

Důkaz:

- Důkaz rovností 1. až 6. je stejný jako ve větě 1.1.
- Rovnost 7. sporem:
  Nechť $\{p_j\}_{j=i-k+1}^{i}$ jsou lineárně závislé.
  Pak $p_i \in S := sp(p_{i-k+1}, ..., p_{i-1})$.
  Dále $p_i = Q_2^{-1}Q_1^{-1}r_i + \beta_{i-1}p_{i-1}$ a tedy $Q_2^{-1}Q_1^{-1}r_i \in S$.
  Ale podle rovnosti 2. je $(Q_1^{-1}r_i, Q_1^{-1}As) = 0 \ \forall s \in S$.
  Je tudíž $(Q_1^{-1}r_i, Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_i) = 0$, a využijeme-li positivní definitnost matice $Q_1^{-1}MQ_2^{-1} =: \tilde{M}$, dostaneme, že $Q_1^{-1}r_i \neq 0$ a protože Q_1 je regulární, tak $r_i \neq 0$.
  Platí tedy, že $\{p_j\}_{j=i-k+1}^{i}$ jsou lineárně nezávislé, takže matice
  $B^{(i)} = (Q_1^{-1}Ap_{i-k+1}, ..., Q_1^{-1}Ap_i)$ má opravdu plný řád a vektor délky kroku $\underline{\alpha}^{(i)}$ je jednoznačně určen. □

Další věta ukazuje, že metoda Axel(k) konverguje.
Věta 1.10: Nechť $\{r_i\}$ je posloupnost residuí generovaná metodou Axel(k). Pak platí:

1.
$$\| r_i \|_2 \leq \ \kappa(Q_1) \cdot \Big[1 - \frac{\lambda_{\min}(\tilde{M})^2}{\lambda_{\max}(\tilde{A}^T\tilde{A})}\Big]^{\frac{i}{2}} \cdot \| r_0 \|_2 =: o1$$

2.
$$\| r_i \|_2 \leq \ \kappa(Q_1) \cdot \Big[1 - \frac{\lambda_{\min}(\tilde{M})^2}{\lambda_{\min}(\tilde{M}) \cdot \lambda_{\max}(\tilde{M}) + \rho(\tilde{R})^2}\Big]^{\frac{i}{2}} \cdot \| r_0 \|_2 =: o2$$

Tedy
$$\| r_i \|_2 \leq \min\{o1, o2\}.$$

Důkaz: Nejprve připomeneme, že
$$\| r_i \| = \| Q_1Q_1^{-1}r_i \| \leq \| Q_1 \| \cdot \| Q_1^{-1}r_i \| = \| Q_1 \| \cdot \| \tilde{r}_i \|$$
a
$$\kappa(Q_1) = \| Q_1 \| \cdot \| Q_1^{-1} \| .$$
Protože norma residua metody Axel(k) je nejmenší mezi všemi ostatními metodami, platí
$$\| \tilde{r}_{i+1} \|_2 \leq \| \bar{r}_{i+1} \|_2,$$
kde $\tilde{r}_{i+1}$ je residuum metody Axel(k) a $\bar{r}_{i+1}$ je např. residuum metody Orthomin(1).
Pro každé i označme
$$\bar{x}_{i+1} := \tilde{x}_i + \bar{\alpha}_i\tilde{p}_i,$$
kde
$$\bar{\alpha}_i := \frac{(\tilde{r}_i, \tilde{A}\tilde{p}_i)}{(\tilde{A}\tilde{p}_i, \tilde{A}\tilde{p}_i)}.$$
To znamená, že z iterace $\tilde{x}_i$ a vektoru $\tilde{p}_i$ metody Axel(k) přejdeme k metodě Orthomin(1) a jedním krokem této metody získáme iteraci $\bar{x}_{i+1}$.
Pro residuum platí
$$\bar{r}_{i+1} = \tilde{r}_i - \bar{\alpha}_i\tilde{A}\tilde{p}_i$$

a dále podle volby $\{\alpha_j\}_{j=i-k+1}^{i}$ metody Axel(k) platí

$$\| \tilde{r}_{i+1} \|_2 \leq \| \bar{r}_{i+1} \|_2$$

(protože norma residua metody Axel(k) je nejmenší).
Víme, že

$$(\tilde{A}\tilde{p}_i, \tilde{A}\tilde{p}_i) \ \leq \ (\tilde{A}\tilde{r}_i, \tilde{A}\tilde{r}_i),$$

viz věta 1.6, a

$$(\tilde{r}_i, \tilde{A}\tilde{p}_i) = (\tilde{r}_i, \tilde{A}\tilde{r}_i),$$

viz 3. rovnost předchozí věty, takže

$$\begin{aligned}
\frac{\| \bar{r}_{i+1} \|_2^2}{\| \tilde{r}_i \|_2^2} &= \frac{(\bar{r}_{i+1}, \bar{r}_{i+1})}{(\tilde{r}_i, \tilde{r}_i)} = \frac{(\tilde{r}_i - \bar{\alpha}_i \tilde{A}\tilde{p}_i, \tilde{r}_i - \bar{\alpha}_i \tilde{A}\tilde{p}_i)}{(\tilde{r}_i, \tilde{r}_i)} = \\
&= \frac{(\tilde{r}_i, \tilde{r}_i) - 2\bar{\alpha}_i(\tilde{r}_i, \tilde{A}\tilde{p}_i) + \bar{\alpha}_i^2(\tilde{A}\tilde{p}_i, \tilde{A}\tilde{p}_i)}{(\tilde{r}_i, \tilde{r}_i)} = \\
&= 1 - 2\frac{(\tilde{r}_i, \tilde{A}\tilde{p}_i)}{(\tilde{A}\tilde{p}_i, \tilde{A}\tilde{p}_i)} \cdot \frac{(\tilde{r}_i, \tilde{A}\tilde{p}_i)}{(\tilde{r}_i, \tilde{r}_i)} + \frac{(\tilde{r}_i, \tilde{A}\tilde{p}_i)^2}{(\tilde{A}\tilde{p}_i, \tilde{A}\tilde{p}_i)^2} \cdot \frac{(\tilde{A}\tilde{p}_i, \tilde{A}\tilde{p}_i)}{(\tilde{r}_i, \tilde{r}_i)} = \\
&= 1 - \frac{(\tilde{r}_i, \tilde{A}\tilde{p}_i)}{(\tilde{r}_i, \tilde{r}_i)} \cdot \frac{(\tilde{r}_i, \tilde{A}\tilde{p}_i)}{(\tilde{A}\tilde{p}_i, \tilde{A}\tilde{p}_i)} \ \leq \\
&\leq 1 - \frac{(\tilde{r}_i, \tilde{A}\tilde{r}_i)}{(\tilde{r}_i, \tilde{r}_i)} \cdot \frac{(\tilde{r}_i, \tilde{A}\tilde{r}_i)}{(\tilde{A}\tilde{r}_i, \tilde{A}\tilde{r}_i)}
\end{aligned}$$

Dále budeme postupovat stejně jako v důkazu věty 1.8 a dostaneme

$$\| \bar{r}_{i+1} \|_2 \leq \ \Big[1 - \frac{\lambda_{\min}(\tilde{M})^2}{\lambda_{\max}(\tilde{A}^T\tilde{A})}\Big]^{\frac{1}{2}} \cdot \| \tilde{r}_i \|_2$$

a

$$\| \bar{r}_{i+1} \|_2 \leq \ \Big[1 - \frac{\lambda_{\min}(\tilde{M})^2}{\lambda_{\min}(\tilde{M}) \cdot \lambda_{\max}(\tilde{M}) + \rho(\tilde{R})^2}\Big]^{\frac{1}{2}} \cdot \| \tilde{r}_i \|_2 \ .$$

Protože platí

$$\| \tilde{r}_{i+1} \|_2 \leq \| \bar{r}_{i+1} \|_2,$$

je důkaz věty dokončen. □

1.1.4 Orthodir

Další alternativní výpočet směrových vektorů představili Young a Jea (viz [Young]). V případě Orthodir je algoritmus 1.3 skombinován s následující Lanczosovou metodou pro výpočet množiny $[(Q_1^{-1}A)^T(Q_1^{-1}A)]$-ortogonálních směrových vektorů:

$$p_{i+1} = Q_2^{-1}Q_1^{-1}Ap_i + \sum_{j=0}^{i} \beta_j^{(i)} p_j,$$

kde

$$\beta_j^{(i)} = -\frac{(Q_1^{-1}AQ_2^{-1}Q_1^{-1}Ap_i, Q_1^{-1}Ap_j)}{(Q_1^{-1}Ap_j, Q_1^{-1}Ap_j)}, \ j \leq i.$$

Pro přehlednost uvedeme celý algoritmus.

Algoritmus 1.10

Zvolíme x_0.
Spočteme $r_0 = f - Ax_0$.
Položíme $p_0 = Q_2^{-1}Q_1^{-1}r_0$.
Pro $i = 0, 1, 2, ...$ **provedeme**
$\alpha_i = \frac{(Q_1^{-1}r_i, Q_1^{-1}Ap_i)}{(Q_1^{-1}Ap_i, Q_1^{-1}Ap_i)}$
$x_{i+1} = x_i + \alpha_i p_i$
$r_{i+1} = r_i - \alpha_i Ap_i$
$\beta_j^{(i)} = -\frac{(Q_1^{-1}AQ_2^{-1}Q_1^{-1}Ap_i, Q_1^{-1}Ap_j)}{(Q_1^{-1}Ap_j, Q_1^{-1}Ap_j)},\ j \le i$
$p_{i+1} = Q_2^{-1}Q_1^{-1}Ap_i + \sum_{j=0}^{i} \beta_j^{(i)} p_j$
`Konec cyklu pro` i.
Konec Algoritmu.

Metoda daná tímto algoritmem se nazývá „**Orthodir**“.

Představíme ještě odseknutou variantu metody Orthodir, kde podobně jako u metody Orthomin vezmeme k výpočtu nového směru p_{i+1} pouze posledních k směrových vektorů $\{p_j\}_{j=i-k+1}^{i}$.

Algoritmus 1.11

Zvolíme x_0.
Spočteme $r_0 = f - Ax_0$.
Položíme $p_0 = Q_2^{-1}Q_1^{-1}r_0$.
Pro $i = 0, 1, 2, ...$ **provedeme**
$\alpha_i = \frac{(Q_1^{-1}r_i, Q_1^{-1}Ap_i)}{(Q_1^{-1}Ap_i, Q_1^{-1}Ap_i)}$
$x_{i+1} = x_i + \alpha_i p_i$
$r_{i+1} = r_i - \alpha_i Ap_i$
$\beta_j^{(i)} = -\frac{(Q_1^{-1}AQ_2^{-1}Q_1^{-1}Ap_i, Q_1^{-1}Ap_j)}{(Q_1^{-1}Ap_j, Q_1^{-1}Ap_j)},\ j \le i$
$p_{i+1} = Q_2^{-1}Q_1^{-1}Ap_i + \sum_{j=i-k+1}^{i} \beta_j^{(i)} p_j$
`Konec cyklu pro` i.
Konec Algoritmu.

Metodu danou tímto useknutým algoritmem nazveme „**Orthodir(k)**“.

Je-li symetrická část $\tilde{M}$ matice $\tilde{A}$ positivně definitní, což předpokládáme, $\{\hat{p}_i\}$ množina směrových vektorů generovaná metodou GCR a $p_0 = \hat{p}_0$, pak $p_i = \gamma_i \hat{p}_i$ pro nějaký skalár γ_i. Tedy metoda Orthodir je ekvivalentní s metodou GCR. Protože $p_0 = \hat{p}_0$, platí též $\tilde{p}_0 = \hat{\tilde{p}}_0$, neboť došlo pouze k vynásobení celé rovnosti maticí Q_2. Dokážeme jen, že $p_1 = \gamma_1 \hat{p}_1$.

- Orthodir:
$$p_1 = Q_2^{-1}Q_1^{-1}Ap_0 + \beta_0^{(0)}p_0 \quad \Leftrightarrow \quad \tilde{p}_1 = \tilde{A}\tilde{p}_0 + \beta_0^{(0)}\tilde{p}_0;$$
- GCR:
$$\begin{aligned} & \hat{p}_1 &=& Q_2^{-1}Q_1^{-1}\hat{r}_1 + \hat{\beta}_0^{(0)}\hat{p}_0 \quad \Leftrightarrow \\ \Leftrightarrow \quad & \hat{\tilde{p}}_1 &=& \hat{\tilde{r}}_1 + \hat{\beta}_0^{(0)}\hat{\tilde{p}}_0 = \hat{\tilde{r}}_0 - \hat{\alpha}_0\tilde{A}\hat{\tilde{p}}_0 + \hat{\beta}_0^{(0)}\tilde{p}_0 = \hat{\tilde{p}}_0 - \hat{\alpha}_0\tilde{A}\hat{\tilde{p}}_0 + \hat{\beta}_0^{(0)}\tilde{p}_0 = \\ & &=& -\hat{\alpha}_0\tilde{A}\tilde{p}_0 + (1 + \hat{\beta}_0^{(0)})\tilde{p}_0. \end{aligned}$$

Nyní porovnáme koeficienty z_1 a z_2 u $\tilde{p}_0$ a $\tilde{A}\tilde{p}_0$ a ukážeme, že jsou stejné. Tedy

$$p_1 = \gamma_1 \hat{p}_1 \quad \Leftrightarrow \quad Q_2 p_1 = \gamma_1 Q_2 \hat{p}_1 \quad \Leftrightarrow \quad \tilde{p}_1 = \gamma_1 \hat{\tilde{p}}_1$$

$\Longrightarrow$

$$\tilde{A}\tilde{p}_0 = -\hat{\alpha}_0 \tilde{A}\tilde{p}_0 \cdot z_1 \quad \& \quad \beta_0^{(0)} \tilde{p}_0 = (1 + \hat{\beta}_0^{(0)}) \tilde{p}_0 \cdot z_2.$$

Odtud dostáváme, že

$$z_1 = -\frac{1}{\hat{\alpha}_0} \quad \text{a} \quad z_2 = \frac{\beta_0^{(0)}}{1 + \hat{\beta}_0^{(0)}},$$

kde koeficienty se střechou patří metodě GCR a koeficient bez střechy metodě Orthodir. Ale dosadíme-li za tyto koeficienty příslušné podíly a rovnost $\tilde{r}_0 = \tilde{p}_0$, a vynecháme-li pro přehlednost střechy, dostaneme:

$$\begin{aligned} z_2 &= -\frac{(\tilde{A}^2\tilde{p}_0, \tilde{A}\tilde{p}_0)}{(\tilde{A}\tilde{p}_0, \tilde{A}\tilde{p}_0)} \cdot \frac{1}{1 - \frac{(\tilde{A}\tilde{r}_1, \tilde{A}\tilde{p}_0)}{(\tilde{A}\tilde{p}_0, \tilde{A}\tilde{p}_0)}} = \\ &= -\frac{(\tilde{A}^2\tilde{p}_0, \tilde{A}\tilde{p}_0)}{(\tilde{A}\tilde{p}_0, \tilde{A}\tilde{p}_0)} \cdot \frac{(\tilde{A}\tilde{p}_0, \tilde{A}\tilde{p}_0)}{(\tilde{A}\tilde{p}_0, \tilde{A}\tilde{p}_0) - (\tilde{A}(\tilde{r}_0 - \alpha_0 \tilde{A}\tilde{p}_0), \tilde{A}\tilde{p}_0)} = \\ &= -\frac{(\tilde{A}^2\tilde{p}_0, \tilde{A}\tilde{p}_0)}{(\tilde{A}\tilde{p}_0, \tilde{A}\tilde{p}_0) - (\tilde{A}\tilde{p}_0, \tilde{A}\tilde{p}_0) + \alpha_0(\tilde{A}^2\tilde{p}_0, \tilde{A}\tilde{p}_0)} = -\frac{1}{\alpha_0} = z_1. \end{aligned}$$

Odtud jsme dostali, že $p_1 = \gamma_1 \hat{p}_1$, kde $\gamma_1 = -\frac{1}{\alpha_0}$ a α_0 je koeficient metody GCR. Obecně to znamená tohle:

- Směr $\tilde{p}_i$ metody Orthodir je lineární kombinací vektorů $\tilde{p}_0, \tilde{A}\tilde{p}_0, ..., \tilde{A}^i\tilde{p}_0$;
- Směr $\hat{\tilde{p}}_i$ metody GCR je lineární kombinací vektorů $\hat{\tilde{p}}_0, \tilde{A}\hat{\tilde{p}}_0, ..., \tilde{A}^i\hat{\tilde{p}}_0$;

a podíly koeficientů u $\tilde{p}_0$ a $\hat{\tilde{p}}_0$, $\tilde{A}\tilde{p}_0$ a $\tilde{A}\hat{\tilde{p}}_0$, ... , $\tilde{A}^i\tilde{p}_0$ a $\tilde{A}^i\hat{\tilde{p}}_0$ jsou stejné.

Je-li matice $\tilde{A}$ symetrická, pak se vztah $p_{i+1} = Q_2^{-1}Q_1^{-1}Ap_i + \sum_{j=0}^{i} \beta_j^{(i)} p_j$, který je roven vztahu $\tilde{p}_{i+1} = \tilde{A}\tilde{p}_i + \sum_{j=0}^{i} \beta_j^{(i)} \tilde{p}_j$ redukuje na tříčlennou rekurenci, jak nyní ukážeme. Nechť

$$\bar{B}_i = \begin{pmatrix} -\beta_0^{(0)} & -\beta_0^{(1)} & \dots & -\beta_0^{(i)} \\ 1 & -\beta_1^{(1)} & \dots & -\beta_1^{(i)} \\ & \ddots & \ddots & \vdots \\ & & \ddots & -\beta_i^{(i)} \\ & & & 1 \end{pmatrix} \in \mathbb{R}^{(i+2)\times(i+1)}, \quad \tilde{P}_i = (\tilde{p}_0, ..., \tilde{p}_i) \in \mathbb{R}^{N\times(i+1)}.$$

Označme dále diagonální matici

$$\Omega_i = \begin{pmatrix} (\tilde{A}\tilde{p}_0, \tilde{A}\tilde{p}_0) & & & \\ & (\tilde{A}\tilde{p}_1, \tilde{A}\tilde{p}_1) & & \\ & & \ddots & \\ & & & (\tilde{A}\tilde{p}_i, \tilde{A}\tilde{p}_i) \end{pmatrix} \in \mathbb{R}^{(i+1)\times(i+1)}$$

a nechť matice $B_i \in \mathbb{R}^{(i+1)\times(i+1)}$ vznikne z matice $\bar{B}_i$ vynecháním posledního řádku. Indukcí dokážeme, že platí

$$\tilde{A}\tilde{P}_i = \tilde{P}_{i+1}\bar{B}_i.$$

- $i = 0$:

$$\tilde{A}\tilde{P}_0 = \tilde{A}\tilde{p}_0 = \tilde{p}_1 - \beta_0^{(0)}\tilde{p}_0 = (\tilde{p}_0, \tilde{p}_1) \cdot \begin{pmatrix} -\beta_0^{(0)} \\ 1 \end{pmatrix} = \tilde{P}_1 \bar{B}_0$$

- Nyní přejdeme od i k $i+1$ za předpokladu, že platí $\tilde{A}\tilde{P}_{i-1} = \tilde{P}_i \bar{B}_{i-1}$. Pro vektor $\tilde{p}_i$ platí

$$\begin{aligned} \tilde{A}\tilde{p}_i &= \tilde{p}_{i+1} - \sum_{j=0}^{i} \beta_j^{(i)} \tilde{p}_j = \tilde{p}_{i+1} - \beta_0^{(i)}\tilde{p}_0 - \ldots - \beta_i^{(i)}\tilde{p}_i = \\ &= (\tilde{p}_0, ..., \tilde{p}_{i+1}) \cdot \begin{pmatrix} -\beta_0^{(i)} \\ \vdots \\ -\beta_i^{(i)} \\ 1 \end{pmatrix} = \tilde{P}_{i+1} \cdot \begin{pmatrix} -\beta_0^{(i)} \\ \vdots \\ -\beta_i^{(i)} \\ 1 \end{pmatrix} \end{aligned}$$

  Tedy

$$\begin{aligned} \tilde{A}\tilde{P}_i &= (\tilde{A}\tilde{P}_{i-1}, \tilde{A}\tilde{p}_i) = (\tilde{P}_i \bar{B}_{i-1}, \tilde{P}_{i+1} \cdot \begin{pmatrix} -\beta_0^{(i)} \\ \vdots \\ -\beta_i^{(i)} \\ 1 \end{pmatrix}) = \\ &= (\tilde{P}_{i+1} \cdot \begin{pmatrix} \bar{B}_{i-1} \\ 0 \end{pmatrix}, \tilde{P}_{i+1} \cdot \begin{pmatrix} -\beta_0^{(i)} \\ \vdots \\ -\beta_i^{(i)} \\ 1 \end{pmatrix}) = \tilde{P}_{i+1}\bar{B}_i. \end{aligned}$$

Platí tedy

$$\tilde{A}\tilde{P}_i = \tilde{P}_{i+1}\bar{B}_i \quad \Rightarrow \quad \tilde{A}^2\tilde{P}_i = \tilde{A}\tilde{P}_{i+1}\bar{B}_i \quad \Rightarrow \quad (\tilde{A}\tilde{P}_i)^T\tilde{A}^2\tilde{P}_i = (\tilde{A}\tilde{P}_i)^T\tilde{A}\tilde{P}_{i+1}\bar{B}_i \quad \Rightarrow$$

$$\Rightarrow \quad \tilde{P}_i^T\tilde{A}^3\tilde{P}_i = (\tilde{A}\tilde{P}_i)^T(\tilde{A}\tilde{P}_{i+1})\bar{B}_i$$

a dále

$$\tilde{P}_i^T\tilde{A}^3\tilde{P}_i = (\tilde{P}_i^T\tilde{A}^3\tilde{P}_i)^T = ((\tilde{A}\tilde{P}_i)^T(\tilde{A}\tilde{P}_{i+1})\bar{B}_i)^T = \bar{B}_i^T(\tilde{A}\tilde{P}_{i+1})^T(\tilde{A}\tilde{P}_i).$$

Odtud dostáváme, že

$$(\tilde{A}\tilde{P}_i)^T(\tilde{A}\tilde{P}_{i+1})\bar{B}_i = \bar{B}_i^T(\tilde{A}\tilde{P}_{i+1})^T(\tilde{A}\tilde{P}_i).$$

Ale z $(\tilde{A}^T\tilde{A})$-ortogonality směrů $\tilde{p}_j$ plyne, že

$$\begin{aligned} (\tilde{A}\tilde{P}_i)^T(\tilde{A}\tilde{P}_{i+1})\bar{B}_i &= \begin{pmatrix} \tilde{A}\tilde{p}_0 \\ \tilde{A}\tilde{p}_1 \\ \vdots \\ \tilde{A}\tilde{p}_i \end{pmatrix} \cdot (\tilde{A}\tilde{p}_0, ..., \tilde{A}\tilde{p}_i, \tilde{A}\tilde{p}_{i+1}) \cdot \bar{B}_i = \\ &= \begin{pmatrix} (\tilde{A}\tilde{p}_0, \tilde{A}\tilde{p}_0) & & & & 0 \\ & (\tilde{A}\tilde{p}_1, \tilde{A}\tilde{p}_1) & & & 0 \\ & & \ddots & & \vdots \\ & & & (\tilde{A}\tilde{p}_i, \tilde{A}\tilde{p}_i) & 0 \end{pmatrix} \cdot \bar{B}_i = \Omega_i B_i. \end{aligned}$$

Obdobně

$$\bar{B}_i^T(\tilde{A}\tilde{P}_{i+1})^T(\tilde{A}\tilde{P}_i) = B_i^T\Omega_i.$$

Z těchto úprav jsme nakonec dospěli k závěru, že

$$\Omega_i B_i = B_i^T\Omega_i \quad \Rightarrow \quad B_i = B_i^T$$

a tedy matice B_i je třídiagonální. To znamená, že pro koeficienty $\beta_j^{(i)}$ platí

$$\beta_j^{(i)} = 0 \;\text{ pro }\; i-j \geq 2 \quad \text{a} \quad \beta_{i-1}^{(i)} = 1 \;\; \forall i,$$

odkud dostáváme výše zmíněnou tříčlennou rekurenci pro výpočet směrů p_{i+1}, pokud je matice $\tilde{A} = Q_1^{-1}AQ_2^{-1}$ symetrická. Ale tato rekurence je vlastně metodou Orthodir(2), kde $\beta_{i-1}^{(i)} = 1$ a počítal by se jen koeficient $\beta_i^{(i)}$.

Na závěr vyslovíme jedno lemma pro nesymetrické matice.

Lemma 1.3: Je-li symetrická část $\tilde{M}$ matice $\tilde{A}$ jednotková matice I, tedy $\tilde{A} = I - \tilde{R}$, kde $\tilde{R}$ je antisymetrická část matice $\tilde{A}$, pak metoda Orthodir(2) je ekvivalentní s metodou Orthodir.

Důkaz: Obdobně jako důkaz lemmatu 1.1. □

Podotýkáme, že na rozdíl od metody Orthomin(1) ekvivalence nenastává pro případ metody Orthodir(1).

Dosud nemáme žádné teoretické výsledky, které nám zaručují konvergenci Orthodir(k) pro obecnější nesymetrické systémy. Říká se, že metoda Orthodir v plné, tj. neuseknuté verzi konverguje vždy, i když symetrická část $\tilde{M}$ matice $\tilde{A}$ není positivně definitní. Numerické pokusy, které jsme provedli, ukázali, že tato metoda skutečně konverguje, ale jen pro malé soustavy. Pro větší dochází ke zhroucení. Tato metoda tedy není příliš stabilní. Viz kapitola 4.

1.2 Metody založené na ortogonalitě residuí

V další části této kapitoly podiskutujeme o ortogonalitě residuí, tj. platí podmínka

$$(r_i, r_j) = 0 \quad \text{pro} \quad i \neq j$$

a v předpodmíněném případě

$$(Q_1^{-1}r_i, Q_1^{-1}r_j) = 0 \quad \text{pro} \quad i \neq j.$$

1.2.1 Zobecněná metoda sdružených gradientů GCG

Nejprve ukážeme, jak vypadá metoda sdružených gradientů pro symetrické positivně definitní soustavy

$$Ax = f. \tag{1.7}$$

Aproximace $x_i \in x_0 + \mathcal{K}_i(r_0, A)$, kde $\mathcal{K}_i(r_0, A) = sp(r_0, Ar_0, ..., A^{i-1}r_0)$, splňuje následující minimalizační podmínku

$$\| x_i - x^\star \| = \min_{x \in x_0 + \mathcal{K}_i(r_0,A)} (x^\star - x, A[x^\star - x])^{\frac{1}{2}} = \min_{x \in x_0 + \mathcal{K}_i(r_0,A)} \| x^\star - x \|_A, \tag{1.8}$$

kde $x^\star$ je přesné řešení soustavy (1.7) a $r_0 = f - Ax_0$. V praxi se posloupnost aproximací x_i počítá z rekurencí

$$x_{i+1} = x_i + \alpha_i p_i, \tag{1.9}$$

kde směrové vektory p_i splňují vztahy

$$p_0 = r_0, \quad p_{i+1} = r_{i+1} + \beta_i p_i, \tag{1.10}$$

a koeficienty β_i jsou voleny tak, že pro směrové vektory platí

$$(p_i, Ap_j) = 0 \quad \text{pro} \quad i \neq j. \tag{1.11}$$

Čísla α_i se volí tak, aby platilo

$$\alpha_i = \arg\min_{\alpha>0} \| x^\star - (x_i + \alpha A p_i) \|_A \, . \tag{1.12}$$

Podrobně je tento postup vyložen v knize [Golub]. Algoritmus pak vypadá následovně:

Algoritmus 1.12

Zvolíme x_0.
Spočteme $r_0 = f - Ax_0$.
Položíme $p_0 = r_0$.
Pro $i = 0, 1, 2, ...$ **provedeme**
$\alpha_i = \frac{(r_i, r_i)}{(p_i, Ap_i)}$
$x_{i+1} = x_i + \alpha_i p_i$
$r_{i+1} = r_i - \alpha_i A p_i$
$\beta_i = \frac{(r_{i+1}, r_{i+1})}{(r_i, r_i)}$
$p_{i+1} = r_{i+1} + \beta_i p_i$
`Konec cyklu pro` i.
Konec Algoritmu.

Metodu danou tímto algoritmem nazveme „**metoda sdružených gradientů (CG)** *(Conjugate Gradient method)*“ (ve smyslu optimality).

Residua r_i lze vyjádřit vztahem $r_i = q_i(A) \cdot r_0$, což plyne z tvaru prostoru $\mathcal{K}_i(r_0, A)$, kde $q_i \in P_i$, a kde P_i je prostor všech polynomů stupně i takový, že $q_i(0) = 1$. Opět, stejně jako v případě metody sdružených residuí, dostaneme přesné řešení po nejvýše N krocích. Přesný počet kroků, po kterých dostaneme použitím metody CG přesné řešení, je dán indexem, při kterém se zastaví růst dimense podprostoru $\mathcal{K}_i(r_0, A)$. Pro teoretické úvahy předpokládáme, že $\dim\{\mathcal{K}_i(r_0, A)\} = i$.

Metoda sdružených gradientů splňuje též Galerkinovu podmínku

$$(Ax_i, v) = (f, v) \quad \forall v \in \mathcal{K}_i,$$

tj. r_i je ortogonální na podprostor $\mathcal{K}_i$, což je ekvivalentní s ortogonalitou residuí:

$$(r_i, r_j) = 0 \text{ pro } i \neq j.$$

Tento výsledek použijeme k formulaci tříkrokové rekurence metody CG. Vezmeme Algoritmus 1.12 a dosazením obdržíme:

$$\begin{aligned} x_{i+1} &= x_i + \alpha_i p_i = x_i + \alpha_i(r_i + \beta_{i-1}p_{i-1}) = x_i + \alpha_i(r_i + \beta_{i-1}\frac{x_i - x_{i-1}}{\alpha_{i-1}}) = \\ &= x_i - x_{i-1} + x_{i-1} + \alpha_i r_i + \frac{\alpha_i\beta_{i-1}}{\alpha_{i-1}}(x_i - x_{i-1}) = \\ &= x_{i-1} + \Big(1 + \frac{\alpha_i\beta_{i-1}}{\alpha_{i-1}}\Big)\Big(\alpha_i(1 + \frac{\alpha_i\beta_{i-1}}{\alpha_{i-1}})^{-1}r_i + x_i - x_{i-1}\Big) \end{aligned}$$

Uvažujme tedy iterační postup

$$x_{i+1} = x_{i-1} + \omega_{i+1}(\gamma_i r_i + x_i - x_{i-1}), \tag{1.13}$$

kde $x_{-1} = 0$, γ_i a ω_{i+1} jsou reálná čísla, která dále určíme tak, abychom nemuseli počítat koeficienty α_i a β_i. Toto je obecný tvar pro několik iteračních metod, např. Čebyševova metoda nebo Richardsonova metoda.
Vynásobme celou rovnost maticí A, obraťme znaménko a přičtěme f. Dostaneme rovnost pro residua:

$$r_{i+1} = r_{i-1} - \omega_{i+1}(\gamma_i A r_i - r_i + r_{i-1}). \tag{1.14}$$

Věta 1.11: Položíme-li

$$\gamma_i = \frac{(r_i, r_i)}{(r_i, Ar_i)}, \quad \omega_1 = 1, \quad \omega_{i+1} = \left[1 - \frac{\gamma_i \cdot \| r_i \|_2^2}{\gamma_{i-1} \cdot \| r_{i-1} \|_2^2 \cdot \omega_i}\right]^{-1} \text{ pro } i \geq 1,$$

pak residua $\{r_i\}$ splňují podmínku ortogonality $(r_i, r_j) = 0 \quad \text{pro} \quad i \neq j$ a výsledný algoritmus je ekvivalentní s předchozím, tzn., že dostaneme stejnou posloupnost iterací.
Důkaz: Indukcí: $i = 1 \;\Rightarrow\; j = 0$.

$$\begin{aligned} (r_1, r_0) &= (r_{-1} - \omega_1[\gamma_0 A r_0 - r_0 + r_{-1}], r_0) = \\ &= (r_{-1}, r_0) - \omega_1[\gamma_0(Ar_0, r_0) - (r_0, r_0) + (r_{-1}, r_0)] = \\ &= (r_{-1}, r_0) - 1 \cdot [\frac{(r_0, r_0)}{(r_0, Ar_0)} \cdot (Ar_0, r_0) - (r_0, r_0) + (r_{-1}, r_0)] = 0. \end{aligned}$$

Teď přejdeme od i k $i + 1$.

- $$\begin{aligned} (r_{i+1}, r_i) &= (r_{i-1} - \omega_{i+1}[\gamma_i A r_i - r_i + r_{i-1}], r_i) = \\ &= (r_{i-1}, r_i) - \omega_{i+1}[\gamma_i(Ar_i, r_i) - (r_i, r_i) + (r_{i-1}, r_i)] = \\ &= 0 - \omega_{i+1}[\frac{(r_i, r_i)}{(r_i, Ar_i)} \cdot (Ar_i, r_i) - (r_i, r_i)] + 0 = 0 \end{aligned}$$

  podle indukčního předpokladu.

- $$\begin{aligned} (r_{i+1}, r_{i-1}) &= (r_{i-1} - \omega_{i+1}[\gamma_i A r_i - r_i + r_{i-1}], r_{i-1}) = \\ &= (r_{i-1}, r_{i-1}) - \omega_{i+1}[\gamma_i(Ar_i, r_{i-1}) - (r_i, r_{i-1}) + (r_{i-1}, r_{i-1})] = \\ &= (r_{i-1}, r_{i-1}) - \omega_{i+1}[\gamma_i(Ar_i, r_{i-1}) + (r_{i-1}, r_{i-1})] \end{aligned}$$

podle indukčního předpokladu.
Požadovanou rovnost

$$(r_{i+1}, r_{i-1}) = 0$$

dostaneme, dosadíme-li

$$\omega_{i+1} = \frac{(r_{i-1}, r_{i-1})}{\gamma_i(Ar_i, r_{i-1}) + (r_{i-1}, r_{i-1})}$$

Ukážeme však, že tento výraz pro ω_{i+1} je shodný s výrazem uvedeným v tvrzení věty. Tedy

$$\omega_{i+1} = \frac{(r_{i-1}, r_{i-1})}{\gamma_i(Ar_i, r_{i-1}) + (r_{i-1}, r_{i-1})} = \left[1 + \frac{\gamma_i(Ar_i, r_{i-1})}{(r_{i-1}, r_{i-1})}\right]^{-1} = \left[1 + \frac{\gamma_i(r_i, Ar_{i-1})}{(r_{i-1}, r_{i-1})}\right]^{-1},$$

neboť matice A je symetrická.
Ze vztahu pro residua

$$r_i = r_{i-2} - \omega_i(\gamma_{i-1}Ar_{i-1} - r_{i-1} + r_{i-2})$$

je patrné, že

$$Ar_{i-1} = -\frac{1}{\gamma_{i-1}\omega_i}r_i + v, \quad \text{kde} \quad v \in sp(r_{i-1}, r_{i-2}).$$

Vezměme skalární součin s r_i:

$$(r_i, Ar_{i-1}) = (r_i, -\frac{1}{\gamma_{i-1}\omega_i}r_i + v) = -\frac{1}{\gamma_{i-1}\omega_i}(r_i, r_i) + (r_i, v) = -\frac{1}{\gamma_{i-1}\omega_i}(r_i, r_i)$$

podle indukčního předpokladu. Dosadíme do výrazu pro ω_{i+1} a dostaneme:

$$\omega_{i+1} = \left[1 + \frac{\gamma_i(r_i, Ar_{i-1})}{(r_{i-1}, r_{i-1})}\right]^{-1} = \left[1 - \frac{\gamma_i(r_i, r_i)}{\gamma_{i-1}(r_{i-1}, r_{i-1}) \cdot \omega_i}\right]^{-1},$$

což je shodné s tvrzením věty.

- buď $j \leq i - 2$:

$$\begin{aligned}(r_{i+1}, r_j) &= (r_{i-1} - \omega_{i+1}[\gamma_i Ar_i - r_i + r_{i-1}], r_j) = \\ &= (r_{i-1}, r_j) - \omega_{i+1}[\gamma_i(Ar_i, r_j) - (r_i, r_j) + (r_{i-1}, r_j)] = \\ &= -\omega_{i+1}\gamma_i(Ar_i, r_j) = -\omega_{i+1}\gamma_i(r_i, Ar_j)\end{aligned}$$

podle indukčního předpokladu.
Ale opět

$$Ar_j \in sp(r_{j-1}, r_j, r_{j+1}),$$

takže

$$(r_i, Ar_j) = 0, \quad \text{a tudíž} \quad (r_{i+1}, r_j) = 0,$$

což dokončuje indukci. □

Algoritmus 1.13

Položíme $x_{-1} = 0$.
Zvolíme x_0.
Pro $i = 0, 1, 2, ...$ **provedeme**
$r_i = f - Ax_i$.
$\gamma_i = \frac{(r_i, r_i)}{(r_i, Ar_i)}$
$\omega_{i+1} = \left[1 - \frac{\gamma_i \cdot \|r_i\|_2^2}{\gamma_{i-1} \cdot \|r_{i-1}\|_2^2 \cdot \omega_i}\right]^{-1}$ pro $i \geq 1$, přičemž pro $i = 0$ je $\omega_1 = 1$
$x_{i+1} = x_{i-1} + \omega_{i+1}(\gamma_i r_i + x_i - x_{i-1})$
`Konec cyklu pro` i.
Konec Algoritmu.

Metodu danou tímto algoritmem nazveme „**metoda sdružených gradientů (CG)** *(Conjugate Gradient method)*“ (ve smyslu ortogonality).

Analogický postup odvodíme pro nesymetrické systémy (1.7) speciálního tvaru, kde A je nesymetrická matice řádu N taková, že $A = I - R$, kde R je antisymetrická část matice A. Ukážeme, že v tomto případě lze konstruovat analogicky iterační proces (1.13) tak, že jsou splněny podmínky kolmosti residuí. To tedy znamená, že přesné řešení soustavy dostaneme po nejvýše N iteracích. Stejně jako v symetrickém případě, i zde předpokládáme, že $\dim\{\mathcal{K}_i(r_0, A)\} = i$.
Pro residua platí:

$$\begin{aligned} r_{i+1} &= r_{i-1} - \omega_{i+1}(\gamma_i A r_i - r_i + r_{i-1}) = r_{i-1} - \omega_{i+1}(\gamma_i (I - R) r_i - r_i + r_{i-1}) = \\ &= (1 - \omega_{i+1}) \cdot r_{i-1} + \omega_{i+1}(1 - \gamma_i) \cdot r_i + \omega_{i+1}\gamma_i R r_i. \end{aligned}$$

Tato residua stejně jako v symetrickém případě splňují ortogonální podmínku

$$(r_i, r_j) = 0 \quad \text{pro} \quad i \neq j,$$

která je ekvivalentní Galerkinově podmínce

$$(Ax_i, v) = (f, v) \quad \forall v \in \mathcal{K}_i = sp(r_0, Ar_0, ..., A^{i-1}r_0),$$

a jsou tudíž lineárně nezávislé.
Důkaz ortogonality residuí je shodný s důkazem věty 1.11 s tím rozdílem, že v tomto případě vycházejí jednodušší výrazy pro γ a ω:

$$\gamma_i = 1, \quad \omega_1 = 1, \quad \omega_{i+1} = \left[1 + \frac{(r_i, r_i)}{(r_{i-1}, r_{i-1}) \cdot \omega_i}\right]^{-1} \quad \text{pro} \quad i \geq 1.$$

Je to dáno předpokladem, že $A = I - R$ a využitím antisymetrie matice R.

Algoritmus 1.14

Položíme $x_{-1} = 0$.
Zvolíme x_0.
Pro $i = 0, 1, 2, ...$ **provedeme**
$r_i = f - Ax_i$.
$\eta_i = (r_i, r_i)$
$\omega_{i+1} = \left[1 + \frac{\eta_i}{\eta_{i-1}} \cdot \frac{1}{\omega_i}\right]^{-1}$ pro $i \geq 1$, přičemž pro $i = 0$ je $\omega_1 = 1$
$x_{i+1} = x_{i-1} + \omega_{i+1}(r_i + x_i - x_{i-1})$
`Konec cyklu pro` i.
Konec Algoritmu.

Metoda vycházející z tohoto algoritmu se nazývá „**zobecněná metoda sdružených gradientů (GCG)** *(Generalized Conjugate Gradient method)*“.

Tento algoritmus nyní převedeme na algoritmus s předpodmíněním podobně jako pro metodu sdružených residuí.
Uvažujme soustavu

$$\tilde{A}\tilde{x} = [Q_1^{-1}AQ_2^{-1}][Q_2x] = [Q_1^{-1}f] = \tilde{f}.$$

Odvození vzorců pro algoritmus s předpodmíněním je jednoduché:

1. x_i:
$$\tilde{x}_i = Q_2x_i$$

2. r_i:
$$\tilde{r}_i = \tilde{f} - \tilde{A}\tilde{x}_i = Q_1^{-1}f - Q_1^{-1}AQ_2^{-1}Q_2x_i = Q_1^{-1}r_i$$

3. η_i:
$$\tilde{\eta}_i = (\tilde{r}_i, \tilde{r}_i) = (Q_1^{-1}r_i, Q_1^{-1}r_i) =: \eta$$

4. ω_{i+1}:
$$\tilde{\omega}_{i+1} =: \omega_{i+1}$$

5. x_{i+1}:
$$\begin{aligned} x_{i+1} &= Q_2^{-1}\tilde{x}_{i+1} = Q_2^{-1}(\tilde{x}_{i-1} + \tilde{\omega}_{i+1}(\tilde{r}_i + \tilde{x}_i - \tilde{x}_{i-1})) = \\ &= Q_2^{-1}(Q_2x_{i-1} + \omega_{i+1}(Q_1^{-1}r_i + Q_2x_i - Q_2x_{i-1})) = \\ &= x_{i-1} + \omega_{i+1}(Q_2^{-1}Q_1^{-1}r_i + x_i - x_{i-1}) \end{aligned}$$

Sestavíme-li to dohromady, dostaneme algoritmus s předpodmíněním.

Algoritmus 1.15
Položíme $x_{-1} = 0$.
Zvolíme x_0.
Pro $i = 0, 1, 2, ...$ **provedeme**
$r_i = f - Ax_i$.
$\eta_i = (Q_1^{-1}r_i, Q_1^{-1}r_i)$
$\omega_{i+1} = \left[1 + \frac{\eta_i}{\eta_{i-1}} \cdot \frac{1}{\omega_i}\right]^{-1}$ pro $i \geq 1$, přičemž pro $i = 0$ je $\omega_1 = 1$
$x_{i+1} = x_{i-1} + \omega_{i+1}(Q_2^{-1}Q_1^{-1}r_i + x_i - x_{i-1})$
`Konec cyklu pro` i.
Konec Algoritmu.

Metodu danou tímto algoritmem nazveme „**zobecněná metoda sdružených gradientů (GCG) s předpodmíněním**“.
Podmínka ortogonality residuí se převede stejným způsobem. Platí tedy

$$(\tilde{r}_i, \tilde{r}_j) = (Q_1^{-1}r_i, Q_1^{-1}r_j) = 0 \quad \text{pro} \quad i \neq j,$$

a tedy vektory $\{Q_1^{-1}r_i\}$ jsou lineáně nezávislé.
Dále poznamenejme, že

$$sp(Q_1^{-1}r_0, ..., Q_1^{-1}r_i) \subset sp(Q_1^{-1}r_0, (Q_1^{-1}AQ_2^{-1})Q_1^{-1}r_0, ..., (Q_1^{-1}AQ_2^{-1})^iQ_1^{-1}r_0),$$

což plyne z konstrukce residuí (1.14). A protože jsou vektory $\{Q_1^{-1}r_i\}$ lineárně nezávislé, je inkluse vlastně rovností:

$$sp(Q_1^{-1}r_0, ..., Q_1^{-1}r_i) = sp(Q_1^{-1}r_0, (Q_1^{-1}AQ_2^{-1})Q_1^{-1}r_0, ..., (Q_1^{-1}AQ_2^{-1})^i Q_1^{-1}r_0).$$

Z toho plyne, že podmínka

$$(Q_1^{-1}r_i, Q_1^{-1}r_j) = 0 \quad \text{pro} \quad i \neq j$$

je ekvivalentní Galerkinově podmínce

$$(Q_1^{-1}Ax_i, v) = (Q_1^{-1}f, v)$$

$$\forall v \in \mathcal{K}_i = sp(Q_1^{-1}r_0, (Q_1^{-1}AQ_2^{-1})Q_1^{-1}r_0, ..., (Q_1^{-1}AQ_2^{-1})^{i-1}Q_1^{-1}r_0).$$

Poznámka: Jak již bylo řečeno, zobecněná metoda sdružených gradientů předpokládá, že pro matici koeficientů $\tilde{A}$ platí $\tilde{A} = I - \tilde{R}$, tj., že symetrická část $\tilde{M}$ matice $\tilde{A}$ je jednotková matice. To je velice omezující předpoklad. Podívejme se, jaký musí mít pro tento případ matice $\tilde{A}$ tvar.

$$I = \tilde{M} = \frac{\tilde{A} + \tilde{A}^T}{2} \quad \Rightarrow \quad 2I = \tilde{A} + \tilde{A}^T.$$

Odtud vidíme, že matice $\tilde{A}$ má na diagonále jedničky a mimodiagonální protilehlé prvky jsou navzájem opačné. Má tedy tuto strukturu:

$$\tilde{a}_{i,i} = 1, \quad \tilde{a}_{i,j} = -\tilde{a}_{j,i}, \quad i, j = 1, ..., N, \quad i \neq j.$$

1.2.2 Orthores

Budeme-li uvažovat obecnější tvar nesymetrické matice A, tj. $A = M - R$, resp. $\tilde{A} = \tilde{M} - \tilde{R}$, pak nevystačíme s tříkrokovou rekurencí. Young a Jea (viz [Young]) navrhli zobecnění metody sdružených gradientů, kde k výpočtu iterace x_{i+1} je užito residuum r_i a $i+1$ předchozích iterací $\{x_j\}_{j=0}^i$, tj.

$$x_{i+1} = \beta_i r_i + \sum_{j=0}^{i} \gamma_j^{(i)} x_j,$$

neboli, budeme-li rovnou uvažovat soustavu v předpodmíněném tvaru:

$$\tilde{x}_{i+1} = \beta_i \tilde{r}_i + \sum_{j=0}^{i} \gamma_j^{(i)} \tilde{x}_j,$$

tedy

$$x_{i+1} = \beta_i Q_2^{-1}Q_1^{-1}r_i + \sum_{j=0}^{i} \gamma_j^{(i)} x_j. \tag{1.15}$$

Počítejme residua:

$$\begin{aligned}\tilde{r}_{i+1} &= \tilde{f} - \tilde{A}\tilde{x}_{i+1} = \tilde{f} - \tilde{A}(\beta_i\tilde{r}_i + \sum_{j=0}^{i}\gamma_j^{(i)}\tilde{x}_j) = -\beta_i\tilde{A}\tilde{r}_i + \tilde{f} - \sum_{j=0}^{i}\gamma_j^{(i)}\tilde{A}\tilde{x}_j = \\ &= -\beta_i\tilde{A}\tilde{r}_i + \tilde{f} - \gamma_0^{(i)}\tilde{f} + \gamma_0^{(i)}\tilde{f} - \gamma_0^{(i)}\tilde{A}\tilde{x}_0 - \gamma_1^{(i)}\tilde{f} + \gamma_1^{(i)}\tilde{f} - \gamma_1^{(i)}\tilde{A}\tilde{x}_1 - \dots \\ \dots &\quad -\gamma_i^{(i)}\tilde{f} + \gamma_i^{(i)}\tilde{f} - \gamma_i^{(i)}\tilde{A}\tilde{x}_i = -\beta_i\tilde{A}\tilde{r}_i + \tilde{f} - \sum_{j=0}^{i}\gamma_j^{(i)}\tilde{f} + \sum_{j=0}^{i}\gamma_j^{(i)}\tilde{r}_j.\end{aligned}$$

Budeme-li předpokládat, že

$$\sum_{j=0}^{i}\gamma_j^{(i)} = 1,$$

pak se poslední rovnost zjednoduší na tvar

$$\tilde{r}_{i+1} = -\beta_i\tilde{A}\tilde{r}_i + \sum_{j=0}^{i}\gamma_j^{(i)}\tilde{r}_j,$$

neboli

$$r_{i+1} = -\beta_i A Q_2^{-1}Q_1^{-1}r_i + \sum_{j=0}^{i}\gamma_j^{(i)}r_j. \tag{1.16}$$

Určíme koeficienty β_i a $\{\gamma_j^{(i)}\}_{j=0}^{i}$ tak, aby platilo $(\tilde{r}_i, \tilde{r}_j) = (Q_1^{-1}r_i, Q_1^{-1}r_j) = 0$ pro $i \neq j$. Uvažujme postupně $i = 1, 2, ...$

- $i = 1$: $\quad j = 0$

  $$(\tilde{r}_1, \tilde{r}_0) = (-\beta_0\tilde{A}\tilde{r}_0 + \gamma_0^{(0)}\tilde{r}_0, \tilde{r}_0) = -\beta_0(\tilde{A}\tilde{r}_0, \tilde{r}_0) + \gamma_0^{(0)}(\tilde{r}_0, \tilde{r}_0)$$

  Položíme-li $\gamma_0^{(0)} = 1$, pak z podmínky ortogonality residuí plyne, že $\beta_0 = \frac{(\tilde{r}_0, \tilde{r}_0)}{(\tilde{A}\tilde{r}_0, \tilde{r}_0)}$. Označíme-li

  $$\alpha_0^{(0)} = \frac{(\tilde{A}\tilde{r}_0, \tilde{r}_0)}{(\tilde{r}_0, \tilde{r}_0)}, \quad \text{pak} \quad \beta_0 = \frac{1}{\alpha_0^{(0)}} \quad \text{a} \quad \gamma_0^{(0)} = \beta_0 \cdot \alpha_0^{(0)}.$$

- $i = 2$: $\quad j = 0, 1$

  $$\begin{aligned}(\tilde{r}_2, \tilde{r}_0) &= (-\beta_1\tilde{A}\tilde{r}_1 + \gamma_0^{(1)}\tilde{r}_0 + \gamma_1^{(1)}\tilde{r}_1, \tilde{r}_0) = \\ &= -\beta_1(\tilde{A}\tilde{r}_1, \tilde{r}_0) + \gamma_0^{(1)}(\tilde{r}_0, \tilde{r}_0) + \gamma_1^{(1)}(\tilde{r}_1, \tilde{r}_0) = \\ &= -\beta_1(\tilde{A}\tilde{r}_1, \tilde{r}_0) + \gamma_0^{(1)}(\tilde{r}_0, \tilde{r}_0) \\ (\tilde{r}_2, \tilde{r}_1) &= (-\beta_1\tilde{A}\tilde{r}_1 + \gamma_0^{(1)}\tilde{r}_0 + \gamma_1^{(1)}\tilde{r}_1, \tilde{r}_1) = \\ &= -\beta_1(\tilde{A}\tilde{r}_1, \tilde{r}_1) + \gamma_0^{(1)}(\tilde{r}_0, \tilde{r}_1) + \gamma_1^{(1)}(\tilde{r}_1, \tilde{r}_1) = \\ &= -\beta_1(\tilde{A}\tilde{r}_1, \tilde{r}_1) + \gamma_1^{(1)}(\tilde{r}_1, \tilde{r}_1),\end{aligned}$$

neboť $(\tilde{r}_1, \tilde{r}_0)$ se již rovná nule.
Dostáváme 3 rovnice pro 3 neznámé:

$$\begin{aligned} -\beta_1(\tilde{A}\tilde{r}_1, \tilde{r}_0) + \gamma_0^{(1)}(\tilde{r}_0, \tilde{r}_0) &= 0 \\ -\beta_1(\tilde{A}\tilde{r}_1, \tilde{r}_1) + \gamma_1^{(1)}(\tilde{r}_1, \tilde{r}_1) &= 0 \\ \gamma_0^{(1)} + \gamma_1^{(1)} &= 1 \end{aligned}$$

Položíme-li

$$\alpha_0^{(1)} = \frac{(\tilde{A}\tilde{r}_1, \tilde{r}_0)}{(\tilde{r}_0, \tilde{r}_0)}, \quad \alpha_1^{(1)} = \frac{(\tilde{A}\tilde{r}_1, \tilde{r}_1)}{(\tilde{r}_1, \tilde{r}_1)},$$

pak

$$\beta_1 = \frac{1}{\alpha_0^{(1)} + \alpha_1^{(1)}} \quad \text{a} \quad \gamma_0^{(1)} = \beta_1 \cdot \alpha_0^{(1)}, \quad \gamma_1^{(1)} = \beta_1 \cdot \alpha_1^{(1)}.$$

- Podobně pro obecné i.

Závěr: Čísla $\{\alpha_j^{(i)}\}_{j=0}^i$, β_i a $\{\gamma_j^{(i)}\}_{j=0}^i$ jsou určeny tak, že

$$(\tilde{r}_i, \tilde{r}_j) = (Q_1^{-1}r_i, Q_1^{-1}r_j) = 0 \quad \text{pro} \quad i \neq j, \qquad \text{a} \qquad \sum_{j=0}^{i} \gamma_j^{(i)} = 1.$$

Zobecněním výše uvedených vzorců získáme následující algoritmus.

Algoritmus 1.16

Zvolíme x_0.
Spočteme $r_0 = f - Ax_0$.
Pro $i = 0, 1, 2, ...$ **provedeme**
$\alpha_j^{(i)} = \frac{(Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_i, Q_1^{-1}r_j)}{(Q_1^{-1}r_j, Q_1^{-1}r_j)}, \quad j = 0, ..., i$
$\beta_i = (\sum_{j=0}^i \alpha_j^{(i)})^{-1}$
$\gamma_j^{(i)} = \beta_i \alpha_j^{(i)}, \quad j = 0, ..., i$
$x_{i+1} = \beta_i Q_2^{-1}Q_1^{-1}r_i + \sum_{j=0}^i \gamma_j^{(i)} x_j$
$r_{i+1} = -\beta_i AQ_2^{-1}Q_1^{-1}r_i + \sum_{j=0}^i \gamma_j^{(i)} r_j$
`Konec cyklu pro` i.
Konec Algoritmu.

Metoda vycházející z tohoto algoritmu se nazývá „**Orthores**“.

Stejně jako v případě metody GCG, i zde u metody Orthores platí

$$sp(Q_1^{-1}r_0, ..., Q_1^{-1}r_i) = sp(Q_1^{-1}r_0, (Q_1^{-1}AQ_2^{-1})Q_1^{-1}r_0, ..., (Q_1^{-1}AQ_2^{-1})^i Q_1^{-1}r_0),$$

což plyne ze vztahu pro residua (1.16) a podmínka

$$(Q_1^{-1}r_i, Q_1^{-1}r_j) = 0 \quad \text{pro} \quad i \neq j$$

je ekvivalentní s Galerkinovou podmínkou

$$(Q_1^{-1}Ax_i, v) = (Q_1^{-1}f, v)$$
$$\forall v \in \mathcal{K}_i = sp(Q_1^{-1}r_0, (Q_1^{-1}AQ_2^{-1})Q_1^{-1}r_0, ..., (Q_1^{-1}AQ_2^{-1})^{i-1}Q_1^{-1}r_0).$$

Je možné též uvažovat useknutou verzi této metody, kde stejně jako např. u metody Orthomin(k), je dolní mez sumy nikoli 0, ale nějaké $i-k+1$. Useknutý algoritmus vypadá takto:

Algoritmus 1.17

Zvolíme x_0.
Spočteme $r_0 = f - Ax_0$.
Pro $i = 0, 1, 2, ...$ **provedeme**
$\alpha_j^{(i)} = \frac{(Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_i, Q_1^{-1}r_j)}{(Q_1^{-1}r_j, Q_1^{-1}r_j)}, \quad j = 0, ..., i,$
kde $\alpha_j^{(i)} = 0$ pro $0 \le j \le i - k$
$\beta_i = (\sum_{j=i-k+1}^{i} \alpha_j^{(i)})^{-1}$
$\gamma_j^{(i)} = \beta_i \alpha_j^{(i)}, \quad j = 0, ..., i$
$x_{i+1} = \beta_i Q_2^{-1}Q_1^{-1}r_i + \sum_{j=i-k+1}^{i} \gamma_j^{(i)} x_j$
$r_{i+1} = -\beta_i AQ_2^{-1}Q_1^{-1}r_i + \sum_{j=i-k+1}^{i} \gamma_j^{(i)} r_j$
`Konec cyklu pro` i.
Konec Algoritmu.

Metoda vycházející z tohoto useknutého algoritmu se nazývá „**Orthores(k)**“.

Metoda Orthores(2) odpovídá odvozené tříčlenné rekurenci pro iterace x_i a je-li $\tilde{A} = I - \tilde{R}$, pak je Orthores(2) ekvivalentní s metodou sdružených gradientů GCG.

Z Algoritmu 1.16 nebo 1.17 vidíme, že iterace x_{i+1} je lineární kombinací počáteční aproximace x_0 a vektorů $\{Q_2^{-1}Q_1^{-1}r_j\}_{j=0}^{i}$, neboť iterace x_j v sumě se dají rozepsat pomocí předchozího x_{j+1} a vektorů $\{Q_2^{-1}Q_1^{-1}r_k\}_{k=0}^{j}$, atd. Zkusíme tedy najít jiný vzorec pro x_{i+1}, ve kterém bude lineární kombinace x_0 a vektorů $\{Q_2^{-1}Q_1^{-1}r_j\}_{j=0}^{i}$.
Mějme dáno x_0. Pak nechť

$$x_j = x_0 + \sum_{k=0}^{j-1} \xi_k^{(j)} Q_2^{-1}Q_1^{-1}r_k \quad \text{pro} \quad j \ge 1,$$

toto vyjádření dosadíme do iterace x_{i+1} a najdeme koeficienty $\{\xi_k^{(i+1)}\}_{k=0}^{i}$.

$$\begin{aligned}
x_{i+1} &= \beta_i Q_2^{-1}Q_1^{-1}r_i + \sum_{j=0}^{i} \gamma_j^{(i)} x_j = \\
&= \beta_i Q_2^{-1}Q_1^{-1}r_i + \sum_{j=0}^{i} \gamma_j^{(i)} (x_0 + \sum_{k=0}^{j-1} \xi_k^{(j)} Q_2^{-1}Q_1^{-1}r_k) = \\
&= \beta_i Q_2^{-1}Q_1^{-1}r_i + \sum_{j=0}^{i} \gamma_j^{(i)} x_0 + \sum_{j=1}^{i} \gamma_j^{(i)} \sum_{k=0}^{j-1} \xi_k^{(j)} Q_2^{-1}Q_1^{-1}r_k = \\
&= x_0 + \beta_i Q_2^{-1}Q_1^{-1}r_i + \sum_{j=1}^{i} \sum_{k=0}^{j-1} \gamma_j^{(i)} \xi_k^{(j)} Q_2^{-1}Q_1^{-1}r_k \;=\; \heartsuit
\end{aligned}$$

Teď přeindexujeme sumy:

$$\begin{aligned}
j = 1 \quad &\Rightarrow \quad k = 0 \\
j = 2 \quad &\Rightarrow \quad k = 0, 1 \\
j = 3 \quad &\Rightarrow \quad k = 0, 1, 2 \\
&\dots \\
j = i \quad &\Rightarrow \quad k = 0, 1, 2, \ \dots, \ i - 1
\end{aligned}$$

To je ekvivalentní s tímto:

$$\begin{aligned} k=0 \quad &\Rightarrow \quad j=1,\ \ldots,\ i \\ k=1 \quad &\Rightarrow \quad j=2,\ \ldots,\ i \\ k=2 \quad &\Rightarrow \quad j=3,\ \ldots,\ i \\ &\ldots \\ k=i-1 \quad &\Rightarrow \quad j=i \end{aligned}$$

Tedy

$$\begin{aligned} \heartsuit &= x_0 + \beta_i Q_2^{-1}Q_1^{-1}r_i + \sum_{k=0}^{i-1}\sum_{j=k+1}^{i} \gamma_j^{(i)}\xi_k^{(j)}Q_2^{-1}Q_1^{-1}r_k = \\ &= x_0 + \beta_i Q_2^{-1}Q_1^{-1}r_i + \sum_{k=0}^{i-1}(\sum_{j=k+1}^{i} \gamma_j^{(i)}\xi_k^{(j)}Q_2^{-1}Q_1^{-1})r_k = \\ &= x_0 + \sum_{k=0}^{i} \xi_k^{(i+1)}Q_2^{-1}Q_1^{-1}r_k, \end{aligned}$$

kde

$$\xi_k^{(i+1)} = \sum_{j=k+1}^{i} \gamma_j^{(i)}\xi_k^{(j)} \quad \text{pro} \quad k = 0, ..., i-1$$

a

$$\xi_i^{(i+1)} = \beta_i.$$

Maticově vypadá zápis takto:

$$\begin{pmatrix} \xi_0^{(i+1)} \\ \xi_1^{(i+1)} \\ \vdots \\ \xi_{i-1}^{(i+1)} \\ \xi_i^{(i+1)} \end{pmatrix} = \begin{pmatrix} \xi_0^{(1)} & \xi_0^{(2)} & \ldots & \xi_0^{(i)} & 0 \\ 0 & \xi_1^{(2)} & \ldots & \xi_1^{(i)} & 0 \\ \vdots & & \ddots & \vdots & \vdots \\ \vdots & & & \xi_{i-1}^{(i)} & 0 \\ 0 & \ldots & \ldots & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \gamma_1^{(i)} \\ \gamma_2^{(i)} \\ \vdots \\ \gamma_i^{(i)} \\ \beta_i \end{pmatrix}$$

Iterace $\{x_j\}_{j=1}^{i}$ není v tomto případě potřeba počítat, residua $\{r_j\}_{j=1}^{i}$ se počítají stejně jako výše pomocí předchozích residuí. Bude-li norma residua r_{i+1} pro nějaké i v průběhu výpočtu dostatečně malá, lze spočítat přibližné řešení x_{i+1} podle vzorce

$$x_{i+1} = x_0 + \sum_{k=0}^{i} \xi_k^{(i+1)}Q_2^{-1}Q_1^{-1}r_k.$$

Výhodou tohoto postupu je to, že při programování této metody nemusíme mít k dispozici dvourozměrná pole velikosti $N \times N$ a ukládat do nich spočtené iterace $\{x_j\}_{j=0}^{i}$ a residua $\{r_j\}_{j=0}^{i}$, což potřebujeme k výpočtu iterace x_{i+1}, ale stačí uložit pouze počáteční vektor x_0, residua $\{r_j\}_{j=0}^{i}$ a koeficienty $\{\xi_k^{(j)}\}_{k=0,\ldots,i-1;j=k+1,\ldots,i}$ což je trojúhelníková matice. Tím se sníží nároky na paměť a dále se také sníží počet operací v každém kroku.
Algoritmus můžeme napsat tímto způsobem:

Algoritmus 1.18

Zvolíme x_0.
Spočteme $r_0 = f - Ax_0$.
Pro $i = 0, 1, 2, ...$ **provedeme**
$\alpha_j^{(i)} = \frac{(Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_i, Q_1^{-1}r_j)}{(Q_1^{-1}r_j, Q_1^{-1}r_j)}, \quad j = 0, ..., i$
$\beta_i = (\sum_{j=0}^{i} \alpha_j^{(i)})^{-1}$
$\gamma_j^{(i)} = \beta_i \alpha_j^{(i)}, \quad j = 0, ..., i$
$\xi_i^{(i+1)} = \beta_i$
$\xi_k^{(i+1)} = \sum_{j=k+1}^{i} \gamma_j^{(i)} \xi_k^{(j)}, \quad k = 0, ..., i-1$
$r_{i+1} = -\beta_i A Q_2^{-1} Q_1^{-1} r_i + \sum_{j=0}^{i} \gamma_j^{(i)} r_j$
Je-li $\| r_{i+1} \|_2 < \varepsilon$, pak
$x_{i+1} = x_0 + Q_2^{-1} Q_1^{-1} \sum_{k=0}^{i} \xi_k^{(i+1)} r_k$
`Konec cyklu pro` i.
Konec Algoritmu.

Metodu danou takovým algoritmem nazveme „TM–**Orthores** *(Triangle Method)*“.

Předpoklad, aby matice $\tilde{M}$ byla positivně definitní, je nutný. Metoda Orthores se může zhroutit pro problémy, ve kterých je symetrická část $\tilde{M}$ matice $\tilde{A}$ indefinitní.

Na závěr vyslovíme větu pro odhad chyby metody Orthores, která je obdobou věty 1.4 pro metodu GCR.

Věta 1.12:

1. Metoda Orthores počítá řešení soustavy $Ax = f$ po nejvýše N krocích.

2. Iterace generované metodou Orthores splňují:

$$\begin{aligned} \| x^\star - x_i \|_2 &\leq \kappa(Q_2) \cdot \frac{\| \tilde{A} \|_2}{\lambda_{\min}(\tilde{M})} \cdot \min_{q_i \in P_i} \| q_i(\tilde{A}) \|_2 \cdot \| x^\star - x_0 \|_2 \leq \\ &\leq \kappa(Q_2) \cdot \frac{\| \tilde{A} \|_2}{\lambda_{\min}(\tilde{M})} \cdot \Big[\sqrt{1 - \frac{\lambda_{\min}(\tilde{M})^2}{\lambda_{\max}(\tilde{A}^T\tilde{A})}}\Big]^i \cdot \| x^\star - x_0 \|_2, \end{aligned}$$

   kde P_i je množina všech polynomů stupně i, pro které platí $q(0) = 1$, $x^\star$ je přesné řešení soustavy $Ax = f$ a $\kappa(Q_2) = \| Q_2 \| \cdot \| Q_2^{-1} \|$ je spektrální číslo podmíněnosti matice Q_2.

Důkaz:

1. První tvrzení plyne bezprostředně ze vztahu ortogonality $(r_i, r_j) = 0 \quad \text{pro} \quad i \neq j$.

2. Triviálně platí nerovnost

$$\| x^\star - x_i \|_2 = \| Q_2^{-1}Q_2 x^\star - Q_2^{-1}Q_2 x_i \|_2 \leq \| Q_2^{-1} \| \cdot \| \tilde{x}^\star - \tilde{x}_i \|_2 .$$

   Dále pro každý vektor y platí

$$\lambda_{\min}(\tilde{M}) \leq \frac{(y, \tilde{M}y)}{(y, y)},$$

tedy pro $y := \tilde{x}^\star - \tilde{x}_i$ platí:

$$\begin{aligned} \| \tilde{x}^\star - \tilde{x}_i \|_2^2 \cdot \lambda_{\min}(\tilde{M}) &= (\tilde{x}^\star - \tilde{x}_i, \tilde{x}^\star - \tilde{x}_i) \cdot \lambda_{\min}(\tilde{M}) \leq \\ &\leq (\tilde{x}^\star - \tilde{x}_i, \tilde{M}(\tilde{x}^\star - \tilde{x}_i)) = \\ &= \frac{1}{2}(\tilde{x}^\star - \tilde{x}_i, (\tilde{A} + \tilde{A}^T)(\tilde{x}^\star - \tilde{x}_i)) = \\ &= (\tilde{A}(\tilde{x}^\star - \tilde{x}_i), \tilde{x}^\star - \tilde{x}_i) = (\tilde{f} - \tilde{A}\tilde{x}_i, \tilde{x}^\star - \tilde{x}_i) = \\ &= (\tilde{r}_i, \tilde{x}^\star - \tilde{x}_i) \end{aligned}$$

Odtud dostáváme nerovnost

$$\| \tilde{x}^\star - \tilde{x}_i \|_2^2 \leq \frac{1}{\lambda_{\min}(\tilde{M})} \cdot (\tilde{r}_i, \tilde{x}^\star - \tilde{x}_i).$$

Nechť

$$\mathcal{K}_i(\tilde{r_0}, \tilde{A}) := sp(\tilde{r}_0, \tilde{A}\tilde{r}_0, ..., \tilde{A}^{i-1}\tilde{r}_0).$$

Pro každé $v \in \mathcal{K}_i$ je

$$\tilde{x}^\star - \tilde{x}_i = \tilde{x}^\star - \tilde{x}_0 - v - (\tilde{x}_i - \tilde{x}_0 - v) = \tilde{x}^\star - \tilde{x}_0 - v + w,$$

kde $w \in \mathcal{K}_i$, protože $\tilde{x}_i$ je lineární kombinací $\tilde{r}_{i-1}$ a $\{\tilde{x}_j\}_{j=0}^{i-1}$, $\tilde{r}_{i-1}$ je lineární kombinací $\tilde{A}\tilde{r}_{i-2}$ a $\{\tilde{r}_j\}_{j=0}^{i-2}$, viz Algoritmus 1.16, tudíž $\tilde{x}_i \in \tilde{x}_0 + \mathcal{K}_i$.
Protože platí

$$(\tilde{A}\tilde{x}_i, v) = (\tilde{f}, v) \quad \forall v \in \mathcal{K}_i,$$

je

$$(\tilde{r}_i, w) = 0,$$

a tedy

$$\begin{aligned} (\tilde{r}_i, \tilde{x}^\star - \tilde{x}_i) &= (\tilde{r}_i, \tilde{x}^\star - \tilde{x}_0 - v) \leq \| \tilde{r}_i \|_2 \cdot \| \tilde{x}^\star - \tilde{x}_0 - v \|_2 \leq \\ &\leq \| \tilde{f} - \tilde{A}\tilde{x}_i \|_2 \cdot \min_{v \in \mathcal{K}_i} \| \tilde{x}^\star - \tilde{x}_0 - v \|_2 \leq \\ &\leq \| \tilde{A}\tilde{x}^\star - \tilde{A}\tilde{x}_i \|_2 \cdot \min_{v \in \mathcal{K}_i} \| \tilde{x}^\star - \tilde{x}_0 - v \|_2 \leq \\ &\leq \| \tilde{A} \|_2 \cdot \| \tilde{x}^\star - \tilde{x}_i \|_2 \cdot \min_{v \in \mathcal{K}_i} \| \tilde{x}^\star - \tilde{x}_0 - v \|_2 \end{aligned}$$

podle Schwarz-Cauchyho nerovnosti. Ale pro každé $v \in \mathcal{K}_i$ je

$$v = s_{i-1}(\tilde{A}) \cdot \tilde{r}_0 = \tilde{A}s_{i-1}(\tilde{A}) \cdot (\tilde{x}^\star - \tilde{x}_0)$$

pro nějaký reálný polynom s_{i-1} stupně $i-1$, takže

$$\tilde{x}^\star - \tilde{x}_0 - v = (I - \tilde{A}s_{i-1}(\tilde{A})) \cdot (\tilde{x}^\star - \tilde{x}_0) = q_i(\tilde{A}) \cdot (\tilde{x}^\star - \tilde{x}_0),$$

kde $q_i \in P_i$.
Dáme-li vše dohromady, dostaneme

$$\| \tilde{x}^\star - \tilde{x}_i \|_2^2 \leq \frac{1}{\lambda_{\min}(\tilde{M})} \cdot (\tilde{r}_i, \tilde{x}^\star - \tilde{x}_i) \leq$$

$$\leq \frac{\| \tilde{A} \|_2}{\lambda_{\min}(\tilde{M})} \cdot \| \tilde{x}^\star - \tilde{x}_i \|_2 \cdot \min_{v \in \mathcal{K}_i} \| \tilde{x}^\star - \tilde{x}_0 - v \|_2 \leq$$

$$\leq \frac{\| \tilde{A} \|_2}{\lambda_{\min}(\tilde{M})} \cdot \| \tilde{x}^\star - \tilde{x}_i \|_2 \cdot \min_{q_i \in P_i} \| q_i(\tilde{A}) \cdot (\tilde{x}^\star - \tilde{x}_0) \|_2 \leq$$

$$\leq \frac{\| \tilde{A} \|_2}{\lambda_{\min}(\tilde{M})} \cdot \| \tilde{x}^\star - \tilde{x}_i \|_2 \cdot \min_{q_i \in P_i} \| q_i(\tilde{A}) \|_2 \cdot \| \tilde{x}^\star - \tilde{x}_0 \|_2,$$

tedy

$$\| \tilde{x}^\star - \tilde{x}_i \|_2 \leq \frac{\| \tilde{A} \|_2}{\lambda_{\min}(\tilde{M})} \cdot \min_{q_i \in P_i} \| q_i(\tilde{A}) \|_2 \cdot \| \tilde{x}^\star - \tilde{x}_0 \|_2 \leq$$

$$\leq \frac{\| \tilde{A} \|_2}{\lambda_{\min}(\tilde{M})} \cdot \min_{q_i \in P_i} \| q_i(\tilde{A}) \|_2 \cdot \| Q_2 \|_2 \cdot \| x^\star - x_0 \|_2,$$

odkud již plyne první nerovnost druhého tvrzení věty. Druhá nerovnost se dokáže stejně jako u věty 1.4. □

Tato věta nám dává konvergenci plné verze metody Orthores. Doposud nejsou známy výsledky, které by ukázali, že useknutá metoda Orthores(k) je konvergentní. Aproximace $\tilde{x}_i$ jsou lineární kombinací $\tilde{r}_{i-1}$ a $\{\tilde{x}_j\}_{j=i-k}^{i-1}$ a neplatí tedy $\tilde{x}_i \in \tilde{x}_0 + \mathcal{K}_i$.

1.2.3 Metoda FOM a její modifikace

V tomto odstavci nejprve stručně uvedeme obecnější odvození metod pro řešení systému $Ax = f$. Uvidíme, že např. metoda GCR je speciálním případem tohoto postupu. Budeme se zabývat praktickou realizací, která spočívá v sestrojení ortonormální báze v uvažovaném Krylovově podprostoru. Tento postup pochází od Saada (viz [Saad 1]) a spočívá v tom, že při konstrukci i-té aproximace x_i se uvažují dva i-dimensionální podprostory $K_i, L_i \in \mathbb{R}^N$ takové, že $x_i \in x_0 + K_i$ a residuum r_i je kolmé na L_i. Např. metoda GCR při této formulaci pracuje tak, že $x_i \in x_0 + K_i$, kde $K_i = sp(p_0, ..., p_{i-1})$ a r_i je kolmé na L_i, kde $L_i = sp(Ap_0, ..., Ap_{i-1})$.

Při metodě FOM se volí $K_i = L_i$ a K_i je Krylovův podprostor $\mathcal{K}_i(r_0, A)$ generovaný residuem r_0 a maticí A a výsledná residua jsou na sebe navzájem kolmá.

Nejprve najdeme ortonormální bázi $(v_1, ..., v_i)$ pro prostor $\mathcal{K}_i(r_0, A)$, tj. bude

$$\mathcal{K}_i(A, r_0) = sp(r_0, Ar_0, ..., A^{i-1}r_0) = sp(v_1, v_2, ..., v_i).$$

Pro výpočet těchto vektorů se používá Arnoldiho algoritmus. Ukážeme, jak vypadá.

1. Zvolíme počáteční vektor v_1, pro který platí $\| v_1 \| = 1$.

2. Položíme
$$w = Av_1 - h_{1,1}v_1$$
a koeficient $h_{1,1}$ určíme tak, aby vektor w byl ortogonální na vektor v_1. Tj.
$$0 = (w, v_1) = (Av_1 - h_{1,1}v_1, v_1) = (Av_1, v_1) - h_{1,1}(v_1, v_1) \quad \Rightarrow \quad h_{1,1} = (Av_1, v_1).$$

Dále vezmeme normu vektoru w, tj. spočítáme

$$h_{2,1} = \| w \|$$

a nakonec položíme

$$v_2 = \frac{w}{h_{2,1}}.$$

3. Položíme

$$w = Av_2 - h_{1,2}v_1 \quad \text{a} \quad z = w - h_{2,2}v_2 = Av_2 - h_{1,2}v_1 - h_{2,2}v_2$$

a opět koeficienty $h_{1,1}$ a $h_{2,2}$ určíme tak, aby vektor w byl ortogonální na vektor v_1 a vektor z na v_2. Tedy

$$0 = (w, v_1) = (Av_2 - h_{1,2}v_1, v_1) = (Av_2, v_1) - h_{1,2}(v_1, v_1) \quad \Rightarrow \quad h_{1,2} = (Av_2, v_1) \ ;$$

$$\begin{aligned} 0 &= (z, v_2) = (Av_2 - h_{1,2}v_1 - h_{2,2}v_2, v_2) = \\ &= (Av_2, v_2) - h_{1,2}(v_1, v_2) - h_{2,2}(v_2, v_2) = \\ &= (Av_2, v_2) - h_{2,2}(v_2) \quad \Rightarrow \quad h_{2,2} = (Av_2, v_2). \end{aligned}$$

Vektor z je také ortogonální na vektor v_1:

$$(z, v_1) = (w - h_{2,2}v_2, v_1) = (w, v_1) - h_{2,2}(v_2, v_1) = 0.$$

Dále vezmeme normu vektoru z, tj.

$$h_{3,2} = \| z \|$$

a položíme

$$v_3 = \frac{z}{h_{2,1}}.$$

Poznamenáváme, že $h_{3,2} \neq 0$, neboť kdyby se to rovnalo nule, pak

$$0 = h_{3,2} = \| z \| \quad \Rightarrow \quad z = 0 \quad \Rightarrow \quad 0 = Av_2 - h_{1,2}v_1 - h_{2,2}v_2.$$

Vektory v_1, v_2, Av_2 tudíž nejsou lineárně nezávislé a to je spor.

Tímto způsobem získáme ortonormální vektory $v_1, ..., v_n$. Celý proces můžeme shrnout takto:

Algoritmus 1.19

Zvolíme v_1, kde $\| v_1 \| = 1$.
Pro $i = 1, 2, ...$ provedeme
$h_{j,i} = (Av_i, v_j), \quad j = 1, 2, ..., i$
$w = Av_i - \sum_{j=1}^{i} h_{j,i}v_j$
$h_{i+1,i} = \| w \|$
$v_{i+1} = \frac{w}{h_{i+1,i}}$
`Konec cyklu pro` i.
Konec Algoritmu.

Tento proces, který generuje ortonormální vektory, se nazývá „**Arnoldiho algoritmus**“. Označme $V_i = (v_1, ..., v_i)$ a

$$\bar{H}_i = \begin{pmatrix} h_{1,1} & h_{1,2} & \dots & \dots & h_{1,i} \\ h_{2,1} & h_{2,2} & \dots & \dots & h_{2,i} \\ & h_{3,2} & \dots & \dots & h_{3,i} \\ & & \ddots & & \vdots \\ & & & h_{i,i-1} & h_{i,i} \\ & & & & h_{i+1,i} \end{pmatrix}$$

Matice V_i je ortonormální a $\bar{H}_i \in \mathbb{R}^{(i+1)\times i}$ je horní Hessenbergova matice. Pak Arnoldiho proces lze napsat v maticovém tvaru

$$AV_i = V_{i+1}\bar{H}_i.$$

Důkaz provedeme indukcí podle i stejně jako u podobného vztahu metody Orthodir v části 1.1.4:

- $i = 0$:
$$AV_1 = Av_1 = h_{2,1}v_2 + h_{1,1}v_1 = (v_1, v_2) \cdot \begin{pmatrix} h_{1,1} \\ h_{2,1} \end{pmatrix} = V_2\bar{H}_1$$

- Přechod od i k $i+1$: Nechť $AV_{i-1} = V_i\bar{H}_{i-1}$. Platí

$$\begin{aligned} Av_i &= h_{i+1,i}v_{i+1} + \sum_{j=1}^{i} h_{j,i}v_j = h_{i+1,i}v_{i+1} + h_{i,i}v_i + ... + h_{1,i}v_1 = \\ &= (v_1, ..., v_i, v_{i+1}) \cdot \begin{pmatrix} h_{1,i} \\ \vdots \\ h_{i,i} \\ h_{i+1,i} \end{pmatrix} = V_{i+1} \cdot \begin{pmatrix} h_{1,i} \\ \vdots \\ h_{i,i} \\ h_{i+1,i} \end{pmatrix} \end{aligned}$$

  A tedy

$$\begin{aligned} AV_i &= A \cdot (v_1, ..., v_i) = (AV_{i-1}, Av_i) = (V_i\bar{H}_{i-1}, V_{i+1} \cdot \begin{pmatrix} h_{1,i} \\ \vdots \\ h_{i,i} \\ h_{i+1,i} \end{pmatrix}) = \\ &= (V_{i+1} \cdot \begin{pmatrix} \bar{H}_{i-1} \\ 0 \end{pmatrix}, V_{i+1} \cdot \begin{pmatrix} h_{1,i} \\ \vdots \\ h_{i,i} \\ h_{i+1,i} \end{pmatrix}) = \\ &= V_{i+1} \cdot (\begin{pmatrix} \bar{H}_{i-1} \\ 0 \end{pmatrix}, \begin{pmatrix} h_{1,i} \\ \vdots \\ h_{i,i} \\ h_{i+1,i} \end{pmatrix}) = V_{i+1}\bar{H}_i, \end{aligned}$$

  což dokončuje indukci.

Nyní přejdeme k případu předpodmínění. Uvažujme proto soustavu $\tilde{A}\tilde{x} = \tilde{f}$, kde $\tilde{A} = Q_1^{-1}AQ_2^{-1}$, $\tilde{x} = Q_2x$, $\tilde{f} = Q_1^{-1}f$ a residuum $\tilde{r} = Q_1^{-1}r$. Definujeme-li navíc

$$\tilde{v} = Q_2v,$$

pak dostaneme vztahy v předpodmíněném tvaru.

Algoritmus 1.20

Zvolíme v_1, kde $\| Q_2v_1 \| = 1$.
Pro $i = 1, 2, ...$ provedeme
$h_{j,i} = (Q_1^{-1}Av_i, Q_2v_j), \quad j = 1, 2, ..., i$
$w = Q_1^{-1}Av_i - \sum_{j=1}^{i} h_{j,i}Q_2v_j$
$h_{i+1,i} = \| w \|$
$v_{i+1} = \frac{Q_2^{-1}w}{h_{i+1,i}}$
`Konec cyklu pro` i.

Konec Algoritmu.

Tento proces se nazývá „**Arnoldiho algoritmus s předpodmíněním**“.
V tomto případě platí

$$(Q_2v_j, Q_2v_t) = \delta_{jt},$$

neboli, že vektory $\{Q_2v_i\}$ jsou ortonormální. Nevíme nic o vektorech v_i. Samozřejmě situace se nemění, pokud je $Q_2 = I$. Pak vektory $\{v_i\}$ budou ortonormální stejně jako v nepředpodmíněném případě.
Maticový zápis Arnoldiho procesu lze odvodit obdobně a platí

$$Q_1^{-1}AV_i = Q_2V_{i+1}\bar{H}_i, \quad \text{kde} \quad V_i = (v_1, ..., v_i).$$

Ekvivalentní je zápis

$$Q_1^{-1}AV_i = Q_2V_iH_i + Q_2v_{i+1}h_{i+1,i}e_i^T,$$

kde vektor e_i má na i-tém místě jedničku a jinde samé nuly a H_i je čtvercová horní Hessenbergova matice, která vznikne z matice $\bar{H}_i$ vynecháním posledního řádku.
Jelikož platí $(Q_2v_j, Q_2v_t) = \delta_{jt}$, což znamená, že vektory $\{Q_2v_j\}$ jsou ortonormální, tedy matice Q_2V_i je ortonormální, pak po vynásobení maticí $(Q_2V_i)^{-1} = (Q_2V_i)^T = V_i^{-1}Q_2^{-1}$ dostaneme:

$$H_i = V_i^{-1}Q_2^{-1}Q_1^{-1}AV_i.$$

Tato konstrukce je základem pro třídu metod pro řešení systému $Ax = f$. Je-li dána libovolná počáteční hodnota x_0 s residuem r_0, pak nechť

$$v_1 = \frac{Q_2^{-1}Q_1^{-1}r_0}{\| Q_1^{-1}r_0 \|_2}, \quad K_i := L_i := sp(\tilde{v}_1, ..., \tilde{v}_i) = sp(Q_2v_1, ..., Q_2v_i).$$

Na začátku této podkapitoly bylo definováno, že metoda FOM počítá aproximaci

$$x_i \in x_0 + K_i, \quad \text{kde residuum} \quad r_i \quad \text{je ortogonální na} \quad L_i.$$

To znamená, že

$$x_i = x_0 + V_i\underline{c}^{(i)} \quad \text{a} \quad (r_i, V_i) = 0,$$

což pro předpodmíněný případ dává tvar

$$\tilde{x}_i = \tilde{x}_0 + \tilde{V}_i \underline{c}^{(i)} \quad \text{a} \quad (\tilde{r}_i, \tilde{V}_i) = 0,$$

neboli

$$x_i = x_0 + V_i \underline{c}^{(i)} \quad \text{a} \quad (Q_1^{-1} r_i, Q_2 V_i) = (Q_2 V_i)^T Q_1^{-1} r_i = V_i^{-1} Q_2^{-1} Q_1^{-1} r_i = 0.$$

Vynásobme nyní první rovnici maticí Q_2, maticí $-\tilde{A} = -Q_1^{-1} A Q_2^{-1}$, přičtěme $\tilde{f} = Q_1^{-1} f$ a nakonec vynásobme maticí $(Q_2 V_i)^{-1} = V_i^{-1} Q_2^{-1}$. Dostaneme jednoduchou rovnici pro koeficienty $\underline{c}^{(i)}$.

$$\begin{aligned}
x_i &= x_0 + V_i \underline{c}^{(i)} \\
Q_2 x_i &= Q_2 x_0 + Q_2 V_i \underline{c}^{(i)} \\
Q_1^{-1} f - Q_1^{-1} A x_i &= Q_1^{-1} f - Q_1^{-1} A x_0 - Q_1^{-1} A V_i \underline{c}^{(i)} \\
Q_1^{-1} r_i &= Q_1^{-1} r_0 - Q_1^{-1} A V_i \underline{c}^{(i)} \\
V_i^{-1} Q_2^{-1} Q_1^{-1} r_i &= V_i^{-1} Q_2^{-1} Q_1^{-1} r_0 - V_i^{-1} Q_2^{-1} Q_1^{-1} A V_i \underline{c}^{(i)} \\
0 &= V_i^{-1} Q_2^{-1} Q_1^{-1} r_0 - V_i^{-1} Q_2^{-1} Q_1^{-1} A V_i \underline{c}^{(i)} \\
V_i^{-1} Q_2^{-1} Q_1^{-1} r_0 &= V_i^{-1} Q_2^{-1} Q_1^{-1} A V_i \underline{c}^{(i)}
\end{aligned}$$

Ale vztah $(Q_2 v_j, Q_2 v_t) = \delta_{jt}$ a definice $v_1 = \frac{Q_2^{-1} Q_1^{-1} r_0}{\| Q_1^{-1} r_0 \|_2}$ nám implikují, že

$$\begin{aligned}
V_i^{-1} Q_2^{-1} Q_1^{-1} r_0 &= (Q_2 V_i)^{-1} Q_1^{-1} r_0 = (Q_2 V_i)^T Q_1^{-1} r_0 = \\
&= (Q_2 v_1, ..., Q_2 v_i)^T \cdot Q_2 v_1 \cdot \| \, Q_1^{-1} r_0 \, \|_2 = \| \, Q_1^{-1} r_0 \, \|_2 \cdot e_1;
\end{aligned}$$

což spolu se vztahem $H_i = V_i^{-1} Q_2^{-1} Q_1^{-1} A V_i$ dává:

$$V_i^{-1} Q_2^{-1} Q_1^{-1} r_0 = V_i^{-1} Q_2^{-1} Q_1^{-1} A V_i \underline{c}^{(i)} \quad \Rightarrow \quad \| \, Q_1^{-1} r_0 \, \|_2 \cdot e_1 = H_i \underline{c}^{(i)}.$$

Výpočet aproximace x_i tedy vyžaduje řešení horního Hessenbergova systému rovnic řádu i. Metodu definovanou tímto výběrem x_i označíme jako úplná ortogonalizační metoda (*full orthogonalization method*).
Z rovností

$$x_i = x_0 + V_i \underline{c}^{(i)}, \quad Q_1^{-1} A V_i = Q_2 V_i H_i + Q_2 v_{i+1} h_{i+1,i} e_i^T,$$

$$\| \, Q_1^{-1} r_0 \, \|_2 \cdot e_1 = H_i \underline{c}^{(i)} \quad \text{a} \quad v_1 = \frac{Q_2^{-1} Q_1^{-1} r_0}{\| \, Q_1^{-1} r_0 \, \|_2}$$

lze spočítat residua r_i takto:

$$\begin{aligned}
Q_1^{-1} r_i &= Q_1^{-1} f - Q_1^{-1} A x_i = Q_1^{-1} f - Q_1^{-1} A (x_0 + V_i \underline{c}^{(i)}) = Q_1^{-1} r_0 - Q_1^{-1} A V_i \underline{c}^{(i)} = \\
&= Q_1^{-1} r_0 - (Q_2 V_i H_i + Q_2 v_{i+1} h_{i+1,i} e_i^T) \underline{c}^{(i)} = \\
&= Q_2 v_1 \cdot \| \, Q_1^{-1} r_0 \, \|_2 - Q_2 V_i \cdot \| \, Q_1^{-1} r_0 \, \|_2 \cdot e_1 - Q_2 v_{i+1} h_{i+1,i} e_i^T \underline{c}^{(i)} = \\
&= -Q_2 v_{i+1} h_{i+1,i} e_i^T \underline{c}^{(i)},
\end{aligned}$$

takže

$$\| \, r_i \, \|_2 = \| \, Q_1 Q_2 v_{i+1} h_{i+1,i} e_i^T \underline{c}^{(i)} \, \| = \, |h_{i+1,i} c_i^{(i)}| \cdot \| \, Q_1 Q_2 v_{i+1} \, \| \, .$$

Pro tato residua dále platí:

$$\begin{aligned}(Q_1^{-1}r_i, Q_1^{-1}r_j) &= (-Q_2 v_{i+1} h_{i+1,i} e_i^T \underline{c}^{(i)}, -Q_2 v_{j+1} h_{j+1,j} e_j^T \underline{c}^{(j)}) = \\ &= h_{i+1,i} h_{j+1,j} c_i^{(i)} c_j^{(j)} \cdot (Q_2 v_{i+1}, Q_2 v_{j+1}) = 0\end{aligned}$$

pro $i \neq j$, takže residua jsou (v předpodmíněném tvaru) ortogonální. Nyní dáme všechny vztahy dohromady a získáme algoritmus.

Algoritmus 1.21

Zvolíme x_0.
Spočteme $r_0 = f - Ax_0$.
Spočteme $v_1 = \frac{Q_2^{-1}Q_1^{-1}r_0}{\|Q_1^{-1}r_0\|_2}$
Pro $j = 0, 1, 2, ...$ **provedeme**
$h_{t,j} = (Q_2 v_t, Q_1^{-1} A v_j), \quad t = 1, ..., j$
$w = Q_1^{-1} A v_j - \sum_{t=1}^{j} h_{t,j} Q_2 v_t$
$h_{j+1,j} = \| w \|_2$
$v_{j+1} = \frac{Q_2^{-1} w}{h_{j+1,j}}$
$\underline{c}^{(j)} = \| Q_1^{-1} r_0 \|_2 \cdot H_j^{-1} e_1$
$\| r_j \|_2 = |h_{j+1,j} c_j^{(j)}| \cdot \| Q_1 Q_2 v_{j+1} \|$
`Konec cyklu pro` j.
Položíme $i = j$.
Spočteme $x_i = x_0 + \sum_{j=1}^{i} c_j^{(i)} v_j$.
Konec Algoritmu.

Metodu danou tímto algoritmem nazveme „**Úplná ortogonalizační metoda (FOM)** *(Full Orthogonalization Method)*“.

Useknutá verze Arnoldiho metody $h_{j+1,j} Q_2 v_{j+1} = Q_1^{-1} A v_j - \sum_{t=1}^{j} h_{t,j} Q_2 v_t$ (viz Algoritmus 1.20) je dána tak, že jako dolní mez sumy se bere číslo $j - k + 1$, pokud je větší než nula, tedy

$$h_{j+1,j} Q_2 v_{j+1} = Q_1^{-1} A v_j - \sum_{t=j-k+1}^{j} h_{t,j} Q_2 v_t,$$

kde se v sumě bere do úvahy pouze posledních k vektorů $\{Q_2 v_t\}_{t=j-k+1}^{j}$. Matice H_i je v tomto případě pásová horní Hessenbergova matice s šířkou pásu $k + 1$. Například pro $k = 3$ máme:

$$\begin{aligned}
Q_1^{-1}Av_1 &= h_{2,1}Q_2v_2 + h_{1,1}Q_2v_1 \\
Q_1^{-1}Av_2 &= h_{3,2}Q_2v_3 + h_{1,2}Q_2v_1 + h_{2,2}Q_2v_2 \\
Q_1^{-1}Av_3 &= h_{4,3}Q_2v_4 + h_{1,3}Q_2v_1 + h_{2,3}Q_2v_2 + h_{3,3}Q_2v_3 \\
Q_1^{-1}Av_4 &= h_{5,4}Q_2v_5 + h_{2,4}Q_2v_2 + h_{3,4}Q_2v_3 + h_{4,4}Q_2v_4 \\
Q_1^{-1}Av_5 &= h_{6,5}Q_2v_6 + h_{3,5}Q_2v_3 + h_{4,5}Q_2v_4 + h_{5,5}Q_2v_5,
\end{aligned}$$

maticově

$$Q_1^{-1} A V_5 = Q_2 V_6 \bar{H}_5,$$

kde $V_6 = (v_1, ..., v_6)$ a $\bar{H}_5$ je pásová obdélníková horní Hessenbergova matice 6×5 s šířkou pásu 4:

$$\bar{H}_5 = \begin{pmatrix} h_{1,1} & h_{1,2} & h_{1,3} & & \\ h_{2,1} & h_{2,2} & h_{2,3} & h_{2,4} & \\ & h_{3,2} & h_{3,3} & h_{3,4} & h_{3,5} \\ & & h_{4,3} & h_{4,4} & h_{4,5} \\ & & & h_{5,4} & h_{5,5} \\ & & & & h_{6,5} \end{pmatrix}$$

Maticová rovnost

$$Q_1^{-1} A V_i = Q_2 V_i H_i + Q_2 v_{i+1} h_{i+1,i} e_i^T$$

tedy stále ještě platí, ale matice $Q_2 V_i$ už není ortonormální. Ortonormální jsou pouze poslední k vektory $\{Q_2 v_t\}_{t=j-k+1}^{j}$.
Definujeme-li prostory K_i a L_i stejně jako výše, tj.

$$K_i := L_i := sp(\tilde{v}_1, ..., \tilde{v}_i) = sp(Q_2 v_1, ..., Q_2 v_i),$$

pak lze opět definovat podobnou metodu. Tuto metodu označil Saad jako neúplná ortogonalizační metoda (*incomplete orthogonalization method*). Residua r_i se spočtou stejným způsobem jako v případě metody FOM, tj.

$$\| r_i \|_2 = |h_{i+1,i} c_i^{(i)}| \cdot \| Q_1 Q_2 v_{i+1} \| .$$

Algoritmus 1.22

Zvolíme x_0.
Spočteme $r_0 = f - Ax_0$.
Spočteme $v_1 = \frac{Q_2^{-1} Q_1^{-1} r_0}{\|Q_1^{-1} r_0\|_2}$
Pro $j = 0, 1, 2, ...$ **provedeme**
$h_{t,j} = (Q_2 v_t, Q_1^{-1} A v_j), \quad t = t_j, ..., j$
$w = Q_1^{-1} A v_j - \sum_{t=t_j}^{j} h_{t,j} Q_2 v_t, \quad$ kde $\quad t_j = \max(1, j - k + 1)$
$h_{j+1,j} = \| w \|_2$
$v_{j+1} = \frac{Q_2^{-1} w}{h_{j+1,j}}$
$\underline{c}^{(j)} = \| Q_1^{-1} r_0 \|_2 \cdot H_j^{-1} e_1$
$\| r_j \|_2 = |h_{j+1,j} c_j^{(j)}| \cdot \| Q_1 Q_2 v_{j+1} \|$
`Konec cyklu pro` j.
Položíme $i = j$.
Spočteme $x_i = x_0 + \sum_{j=1}^{i} c_j^{(i)} v_j$.
Konec Algoritmu.

Jak jsme již řekli výše, metodu danou tímto useknutým algoritmem nazveme „**Neúplná ortogonalizační metoda (IOM(k))** *(Incomplete Orthogonalization Method)*“.

Je-li matice A symetrická a positivně definitní, pak se vzorec

$$h_{j+1,j} v_{j+1} = A v_j - \sum_{t=1}^{j} h_{t,j} v_t,$$

viz Algoritmus 1.19 po dosazení za w, redukuje na tříčlennou rekurenci, jak nyní ukážeme. K tomuto účelu si vezmeme vztah

$$A V_i = V_{i+1} \bar{H}_i$$

a budeme ho upravovat.

$$(AV_i)^T = (V_{i+1}\bar{H}_i)^T \quad \Rightarrow \quad V_i^T A^T = V_i^T A = \bar{H}_i^T V_{i+1}^T.$$

Odtud dostáváme, že zaprvé

$$V_i^T A V_i = V_i^T V_{i+1} \bar{H}_i = (I, o) \cdot \bar{H}_i = H_i,$$

kde matice (I, o) je velikosti $i \times (i+1)$, a zadruhé

$$V_i^T A V_i = \bar{H}_i^T V_{i+1}^T V_i = \bar{H}_i^T \cdot \begin{pmatrix} I \\ o^T \end{pmatrix} = H_i^T,$$

kde matice $\begin{pmatrix} I \\ o^T \end{pmatrix}$ je velikosti $(i+1) \times i$. Poznamenáváme, že o značí nulový vektor. Těmito úpravami jsme dospěli k závěru, že $H_i = H_i^T$ a tedy matice H_i je třídiagonální a symetrická, tedy $h_{i+1,i} = h_{i,i+1}\ \forall i$.

Obdobně pro obecný případ předpodmínění. Je-li matice $\tilde{A} = Q_1^{-1} A Q_2^{-1}$ symetrická a positivně definitní, pak se vzorec

$$h_{j+1,j} Q_2 v_{j+1} = Q_1^{-1} A v_j - \sum_{t=1}^{j} h_{t,j} Q_2 v_t$$

redukuje na tříčlennou rekurenci a tudíž metoda FOM je ekvivalentní s metodou IOM(2). V tomto případě je aproximace x_i rovna aproximaci vypočtené po i krocích metody sdružených gradientů.

Další možností je použít LU rozklad matice H_i. Nová aproximace x_i může pak být spočtena z každého nového vektoru v_i. Aby vzniklý algoritmus byl tvarově podobný s ostatními algoritmy, kde se začíná s $i = 0$, tak i zde začneme indexovat od nuly, tj. e_0 bude značit jednotkový vektor s nulovou složkou rovnou jedné.

Mějme Hessenbergovu matici H_i spočtenou po i krocích ze vzorce

$$h_{j+1,j} Q_2 v_{j+1} = Q_1^{-1} A v_j - \sum_{t=1}^{j} h_{t,j} Q_2 v_t$$

nebo

$$h_{j+1,j} Q_2 v_{j+1} = Q_1^{-1} A v_j - \sum_{t=j-k+1}^{j} h_{t,j} Q_2 v_t$$

a nechť existuje LU rozklad matice H_i, tj.

$$H_i = L_i U_i,$$

kde L_i je dolní trojúhelníková matice s jedinou nenulovou poddiagonálou
a U_i je horní trojúhelníková matice s jednotkovou diagonálou.
Rovnost $\| Q_1^{-1} r_0 \|_2 \cdot e_1 = H_i \underline{c}^{(i)}$ upravíme takto:

$$V_i \underline{c}^{(i)} = \| Q_1^{-1} r_0 \|_2 \cdot V_i H_i^{-1} e_0 = \| Q_1^{-1} r_0 \|_2 \cdot V_i U_i^{-1} L_i^{-1} e_0.$$

Vektor e_0 proto, protože začínáme od $i = 0$, tj. $V_i = (v_0, ..., v_i)$ a prvky $\{h_{k,l}\}$ matice H_i se též indexují od nuly.
Nechť dále

$$P_i := V_i U_i^{-1} \quad \text{a} \quad \underline{\alpha}^{(i)} := \| Q_1^{-1} r_0 \|_2 \cdot L_i^{-1} e_0.$$

Aproximace x_{i+1} je opět lineární kombinací posledních $i+1$ vektorů v_i, tj.

$$\begin{aligned} x_{i+1} &= x_0 + V_i \underline{c}^{(i)} = x_0 + \| Q_1^{-1} r_0 \|_2 \cdot V_i U_i^{-1} L_i^{-1} e_0 = \\ &= x_0 + P_i \underline{\alpha}^{(i)} = x_0 + \sum_{j=0}^{i} \alpha_j^{(i)} p_j = x_i + \alpha_i^{(i)} p_i, \end{aligned}$$

kde p_j je j-tý sloupec matice P_i, nazveme ho směr a $\alpha_j^{(i)}$ je j-tá složka vektoru $\underline{\alpha}^{(i)}$.
Hodnoty p_i a $\{\alpha_j^{(i)}\}_{j=0}^{i}$ spočítáme přímo z matic $V_i, \quad L_i, \quad U_i$:

$$P_i = V_i U_i^{-1} \quad \Rightarrow \quad V_i = P_i U_i,$$

tj.

$$\begin{pmatrix} | & | & & | \\ v_0 & v_1 & \dots & v_i \\ | & | & & | \end{pmatrix} = \begin{pmatrix} | & | & & | \\ p_0 & p_1 & \dots & p_i \\ | & | & & | \end{pmatrix} \cdot \begin{pmatrix} 1 & u_{0,1} & \dots & u_{0,i} \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & u_{i-1,i} \\ 0 & \dots & 0 & 1 \end{pmatrix}$$

Odtud plyne, že

$$\begin{aligned} v_0 &= p_0 \\ v_1 &= p_0 u_{0,1} + p_1 \\ &\vdots \\ v_i &= p_0 u_{0,i} + p_1 u_{1,i} + \ldots + p_{i-1} u_{i-1,i} + p_i, \end{aligned}$$

čímž získáváme vzorec pro p_i:

$$p_i = v_i - \sum_{j=0}^{i-1} u_{j,i} p_j.$$

Obdobně spočítáme koeficienty $\{\alpha_j^{(i)}\}_{j=0}^{i}$:

$$\underline{\alpha}^{(i)} = \| Q_1^{-1} r_0 \|_2 \cdot L_i^{-1} e_0 \quad \Rightarrow \quad L_i \cdot \underline{\alpha}^{(i)} = \| Q_1^{-1} r_0 \|_2 \cdot e_0,$$

tj.

$$\begin{pmatrix} l_{0,0} & 0 & \dots & \dots & 0 \\ l_{1,0} & \ddots & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & l_{i,i-1} & l_{i,i} \end{pmatrix} \cdot \begin{pmatrix} \alpha_0^{(i)} \\ \vdots \\ \vdots \\ \vdots \\ \alpha_i^{(i)} \end{pmatrix} = \| Q_1^{-1} r_0 \|_2 \cdot \begin{pmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix}$$

Odtud dostáváme, že

$$\begin{aligned} l_{0,0} \cdot \alpha_0^{(i)} &= \| Q_1^{-1} r_0 \|_2 \\ l_{1,0} \cdot \alpha_0^{(i)} + l_{1,1} \cdot \alpha_1^{(i)} &= 0 \\ &\vdots \\ l_{i,i-1} \cdot \alpha_{i-1}^{(i)} + l_{i,i} \cdot \alpha_i^{(i)} &= 0, \end{aligned}$$

a tím získáváme vzorec pro $\{\alpha_j^{(i)}\}_{j=0}^{i}$:

$$\alpha_j^{(i)} = -\frac{l_{i,i-1} \cdot \alpha_{j-1}^{(i)}}{l_{i,i}}; \quad \text{přičemž} \quad \alpha_0^{(i)} = \frac{\| Q_1^{-1} r_0 \|_2}{l_{0,0}}$$

Protože matice H_{i+1} vznikne z matice H_i přidáním
- $(i+1)$-ního řádku $(0, ..., 0, h_{i+1,i}, h_{i+1,i+1})$ a
- $(i+1)$-ního sloupce $(h_{0,i+1}, ..., h_{i+1,i+1})^T$,

pak matice L_{i+1} je také rovna matici L_i, ke které přidáme
- $(i+1)$-ní řádek $(0, ..., 0, l_{i+1,i}, l_{i+1,i+1})$ a
- $(i+1)$-ní sloupec $(0, ..., 0, l_{i+1,i+1})^T$;

a rovněž za matici U_{i+1} lze vzít matici U_i rozšířenou o
- $(i+1)$-ní řádek $(0, ..., 0, 1)$ a
- $(i+1)$-ní sloupec $(u_{0,i+1}, ..., u_{i,i+1}, 1)^T$.

Odtud plyne, že pro koeficienty α platí:

$$\alpha_j^{(i+1)} = \alpha_j^{(i)}, \quad j = 0, ..., i;$$

stačí tedy v každém kroku algoritmu uvedeného níže (kde je index i místo indexu $i+1$) spočítat pouze koeficient $\alpha_{i+1}^{(i+1)}$.

Nyní vyjádříme normu residuí. Platí:

$$\begin{aligned} H_i \underline{c}^{(i)} &= \| Q_1^{-1} r_0 \|_2 \cdot e_0 \quad \Rightarrow \\ \Rightarrow \quad \underline{c}^{(i)} &= \| Q_1^{-1} r_0 \|_2 \cdot H_i^{-1} \cdot e_0 = \| Q_1^{-1} r_0 \|_2 \cdot U_i^{-1} L_i^{-1} \cdot e_0 = U_i^{-1} \underline{\alpha}^{(i)} \quad \Rightarrow \\ \Rightarrow \quad U_i \underline{c}^{(i)} &= \underline{\alpha}^{(i)}, \end{aligned}$$

tj.

$$\begin{pmatrix} 1 & u_{0,1} & \dots & u_{0,i} \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & u_{i-1,i} \\ 0 & \dots & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} c_0^{(i)} \\ \vdots \\ \vdots \\ c_i^{(i)} \end{pmatrix} = \begin{pmatrix} \alpha_0^{(i)} \\ \vdots \\ \vdots \\ \alpha_i^{(i)} \end{pmatrix},$$

neboli

$$\begin{aligned} c_0^{(i)} + u_{0,1} c_1^{(i)} + \ldots + u_{0,i} c_i^{(i)} &= \alpha_0^{(i)} \\ c_1^{(i)} + \ldots + u_{1,i} c_i^{(i)} &= \alpha_1^{(i)} \\ &\vdots \\ c_i^{(i)} &= \alpha_i^{(i)} \end{aligned}$$

Pro poslední složku tedy platí

$$|c_i^{(i)}| = |\alpha_i^{(i)}|.$$

V normě residuí se vyskytuje právě poslední složka, proto dostáváme, že

$$\| r_{i+1} \| = |h_{i+1,i} c_i^{(i)}| \cdot \| Q_1 Q_2 v_{i+1} \| = |h_{i+1,i} \alpha_i^{(i)}| \cdot \| Q_1 Q_2 v_{i+1} \| .$$

Nakonec můžeme celý algoritmus předvést. Pro přehlednost vynecháme horní index u koeficientů α.

Algoritmus 1.23

Zvolíme x_0.
Spočteme $r_0 = f - Ax_0$.
Spočteme $v_0 = \frac{Q_2^{-1}Q_1^{-1}r_0}{\|Q_1^{-1}r_0\|_2}$
Pro $i = 0, 1, 2, ...$ **provedeme**
$h_{j,i} = (Q_2 v_j, Q_1^{-1} A v_i), \quad j = 0, ..., i$
Provedeme rozklad $H_i = L_i U_i$.
$p_i = v_i - \sum_{j=0}^{i-1} u_{j,i} p_j$
$\alpha_i = -\frac{l_{i,i-1}\alpha_{i-1}}{l_{i,i}}$ pro $i \geq 1$, přičemž $\alpha_0 = \frac{\|Q_1^{-1}r_0\|}{l_{0,0}}$
$x_{i+1} = x_i + \alpha_i p_i$
$w = Q_1^{-1} A v_i - \sum_{j=0}^{i} h_{j,i} Q_2 v_j$
$h_{i+1,i} = \| w \|_2$
$v_{i+1} = \frac{Q_2^{-1} w}{h_{i+1,i}}$
$\| r_{i+1} \|_2 = |h_{i+1,i}\alpha_i| \cdot \| Q_1 Q_2 v_{i+1} \|$
`Konec cyklu pro` i.
Konec Algoritmu.

Metoda daná tímto algoritmem se nazývá „**Úplná ortogonalizační metoda se směry (DFOM)** *(Directions Full Orthogonalization Method)*“.

Obdobně lze sestavit odseknutou variantu této metody, obdobně jako výše.

Algoritmus 1.24

Zvolíme x_0.
Spočteme $r_0 = f - Ax_0$.
Spočteme $v_0 = \frac{Q_2^{-1}Q_1^{-1}r_0}{\|Q_1^{-1}r_0\|_2}$
Pro $i = 0, 1, 2, ...$ **provedeme**
$h_{j,i} = (Q_2 v_j, Q_1^{-1} A v_i), \quad j = j_i, ..., i$
Provedeme rozklad $H_i = L_i U_i$.
$p_i = Q_2 v_i - \sum_{j=j_i}^{i-1} u_{j,i} p_j, \quad$ kde $j_i = \max(0, i-k+1)$
$\alpha_i = -\frac{l_{i,i-1}\alpha_{i-1}}{l_{i,i}}, \quad$ pro $i \geq 1, \quad$ přičemž $\alpha_0 = \frac{1}{l_{0,0}}$
$x_{i+1} = x_i + \alpha_i p_i$
$w = Q_1^{-1} A v_i - \sum_{j=j_i}^{i} h_{j,i} Q_2 v_j$
$h_{i+1,i} = \| w \|_2$
$v_{i+1} = \frac{Q_2^{-1} w}{h_{i+1,i}}$
$\| r_{i+1} \|_2 = |h_{i+1,i}\alpha_i| \cdot \| Q_1 Q_2 v_{i+1} \|$
`Konec cyklu pro` i.
Konec Algoritmu.

Metoda daná tímto useknutým algoritmem se tedy nazývá „**Neúplná ortogonalizační metoda se směry (DIOM(k))** *(Directions Incomplete Orthogonalization Method)*“.

Ze vzorců

$$h_{j+1,j} Q_2 v_{j+1} = Q_1^{-1} A v_j - \sum_{t=1}^{j} h_{t,j} Q_2 v_t, \quad (Q_2 v_j, Q_2 v_t) = \delta_{jt}$$

a

$$Q_1^{-1} r_i = -Q_2 v_{i+1} h_{i+1,i} e_i^T \underline{c}^{(i)}$$

plyne, že metody FOM a DFOM splňují podmínky ortogonality (viz výše) a Galerkinovy podmínky analogické s těmi u metody Orthores:

$$(Q_1^{-1}r_i, Q_1^{-1}r_j) = 0 \quad \text{a} \quad (Q_1^{-1}Ax_i, z) = (Q_1^{-1}f, z)$$

$$\forall z \in sp(Q_1^{-1}r_0, (Q_1^{-1}AQ_2^{-1})Q_1^{-1}r_0, ..., (Q_1^{-1}AQ_2^{-1})^{i-1}Q_1^{-1}r_0);$$

tedy v našem případě metody FOM

$$(Q_1^{-1}r_i, Q_1^{-1}r_j) = 0 \quad \text{a} \quad (Q_1^{-1}Ax_i, z) = (Q_1^{-1}f, z) \ \ \forall z \in sp(Q_2v_1, ..., Q_2v_i)$$

a u metody DFOM

$$(Q_1^{-1}r_i, Q_1^{-1}r_j) = 0 \quad \text{a} \quad (Q_1^{-1}Ax_i, z) = (Q_1^{-1}f, z) \ \ \forall z \in sp(Q_2v_0, ..., Q_2v_i).$$

Metody FOM a DFOM jsou tedy ekvivalentní s metodou Orthores a odhad chyby ve větě 1.12 platí pro obě dvě metody FOM a DFOM.

Věta 1.13: Iterace generované metodami FOM nebo DFOM splňují:

$$\begin{aligned} \| x^\star - x_i \|_2 &\leq \kappa(Q_2) \cdot \frac{\| \tilde{A} \|_2}{\lambda_{\min}(\tilde{M})} \cdot \min_{q_i \in P_i} \| q_i(\tilde{A}) \|_2 \cdot \| x^\star - x_0 \|_2 \leq \\ &\leq \kappa(Q_2) \cdot \frac{\| \tilde{A} \|_2}{\lambda_{\min}(\tilde{M})} \cdot \Big[\sqrt{1 - \frac{\lambda_{\min}(\tilde{M})^2}{\lambda_{\max}(\tilde{A}^T\tilde{A})}}\Big]^i \cdot \| x^\star - x_0 \|_2, \end{aligned}$$

kde P_i je množina všech polynomů stupně i, pro které platí $q(0) = 1$, $x^\star$ je přesné řešení soustavy $Ax = f$ a $\kappa(Q_2) = \| Q_2 \| \cdot \| Q_2^{-1} \|$ je spektrální číslo podmíněnosti matice Q_2.

Důkaz: Podobný jako ve větě 1.12. □

Kapitola 2

Axelssonova metoda GCG-LS

V části 1.1.3 jsme se zmínili o Axelssonově zobecnění metody sdružených residuí, kde se řeší problém nejmenších čtverců. V této kapitole se budeme podrobněji věnovat tomuto zobecnění a představíme metodu, kterou Axelsson nazval metoda GCG-LS, neboli *Generalized Conjugate Gradient Least Square method*, zobecněná metoda sdružených gradientů ve smyslu nejmenších čtverců, viz [Axel 2]. (Tuto metodu budeme také někdy značit jako Axelssonovo zobecnění, nebo zkráceně Axelsson.)

Budeme uvažovat verzi plnou (tj. takovou, že využijeme všechny předchozí spočtené vektory) i useknutou (tj. takovou, kde bereme do úvahy pouze posledních s spočtených vektorů) a popíšeme situaci, kdy jsou obě verze identické. Dále ukážeme výhody se zachováním tzv. kontrolního členu v useknutém případě a předvedeme algoritmus založený na speciálním skalárním součinu.

Uvidíme také, že algoritmus, který představíme v této kapitole, je se speciální volbou parametrů totožný s algoritmy 1.8 a 1.9 v sekci 1.1.3.

2.1 Úvod

V nedávné době bylo navrženo několik zobecněných verzí metody sdružených gradientů pro řešení lineárního systému $Ax = f$. V těchto metodách se rekursivně konstruuje posloupnost hledaných směrů $p_{k-j}, \quad j = k, k-1, ..., 0$ a počítá nová aproximace řešení x_{k+1} jako lineární kombinace předchozích hledaných směrů.

Pro výpočet hledaných směrů se navrhlo mnoho metod. Například vznikla myšlenka počítat je rekursivně jako lineární kombinace nejnovějšího residua a předchozích hledaných směrů. Zvláště se doporučilo užít useknutý tvar této metody, kde se x_{k+1} počítá jako součet počátečního x_0 a lineární kombinace směrů p_k, p_{k-1}, p_{k-2}. Vložením vhodné ortogonální podmínky se navíc vyloučil koeficient u p_{k-1} a člen s p_{k-2} se použil jako kontrolní člen, jehož velikost mohla říct, zda je takové useknutí výhodné.

V této kapitole vyložíme metodu, ve které se p_k počítá pomocí předchozích $(s+1)$ směrů a x_{k+1} je rovno součtu x_0 a lineární kombinace p_k a p_{k-s-2} a kde člen s p_{k-s-2} slouží jako kontrolní člen. Bude-li jeho relativní hodnota malá, pak můžeme říct, že jsme nalezli správnou hodnotu s.

Ve druhé části předvedeme algoritmus a řekneme něco o konvergenci, ve třetí části budeme zkoumat, kdy dostaneme přesné řešení a ve čtvrté části stanovíme podmínky, pro které je $(s+1)$-členná verze identická s plnou verzí, tj. takovou, ve které uvažujeme všechny dosud spočtené hledané směry p_k.

Uvažujme tedy lineární systém $Ax = f$, kde A je regulární, nesymetrická a reálná matice řádu N, $x \in \mathbb{R}^N$ a $f \in \mathbb{R}^N$.

2.2 Algoritmus

Uvažujme minimalizaci kvadratického funkcionálu

$$E(x) = \frac{1}{2}(r, r)_0 = \frac{1}{2}(f - Ax, f - Ax)_0. \tag{2.1}$$

Zde $(.,.)_0$ je skalární součin v $\mathbb{R}^N$ definovaný $(u, v)_0 = (u, M_0 v)$, kde M_0 je symetrická a positivně definitní matice. Odpovídající norma je $\| u \|_0 = (u, u)_0^{\frac{1}{2}}$. Nechť $\hat{x}$ je takový vektor, že platí $E(\hat{x}) = \inf_{x \in \mathbb{R}^N} E(x)$.

Předpokládejme, že matice $M_0 A + A^T M_0$ je positivně definitní. To znamená, že pro každé $x \neq 0$ platí

$$\begin{aligned} 0 < (x, (M_0 A + A^T M_0)x) &= (x, M_0 Ax) + (x, A^T M_0 x) = (x, Ax)_0 + (Ax, x)_0 = \\ &= (x, Ax)_0 + \overline{(x, Ax)}_0 = (x, \lambda x)_0 + \overline{(x, \lambda x)}_0 = \lambda + \overline{\lambda}, \end{aligned}$$

kde λ je vlastní číslo matice A příslušné vlastnímu vektoru x, jehož „nulková“ norma je rovna jedné. Odtud je vidět, že spektrum matice $M_0 A + A^T M_0$ leží vpravo od imaginární osy.

Nyní odvodíme zobecněnou metodu sdružených gradientů ve smyslu nejmenších čtverců pro řešení soustavy $Ax = f$, která počítá právě vektor $\hat{x}$. Nechť je dáno přirozené číslo $t \geq 1$, vektor x_k, což je nějaká aproximace přesného řešení $\hat{x}$ a $(t_k + 1)$ vektorů p_{k-j}, $j = 0, ..., t_k$, které nazveme hledané směry, kde t_k leží v intervalu

$$\langle 1, ..., \min\{t_{k-1} + 1, t\} \rangle, \quad t_0 = 0$$

a kde množina $\{Ap_{k-j}\}_{j=0}^{t_k}$ je lineárně nezávislá. Určíme nový vektor x_{k+1} následujícím způsobem.

Najdeme $t_k + 1$ parametrů

$$\alpha_{k-j}^{(k)} \in (-\infty, \infty), \quad j = 0, ..., t_k$$

tak, aby funkcionál $E(x)$ nabýval svého minima v bodě

$$x_{k+1} = x_k + \sum_{j=0}^{t_k} \alpha_{k-j}^{(k)} p_{k-j}. \tag{2.2}$$

Z důvodu zjednodušení uvažujme na chvíli $\alpha_{k-j}^{(k)}$ jako proměnné.

Odvoďme, co plyne z nutné podmínky $\frac{\partial E}{\partial \alpha_{k-l}^{(k)}} = 0, \quad l = 0, ..., t_k$.

$$\begin{aligned} E(x_{k+1}) &= \frac{1}{2}(f - Ax_{k+1}, f - Ax_{k+1})_0 = \\ &= \frac{1}{2}(f - A(x_k + \sum_{j=0}^{t_k} \alpha_{k-j}^{(k)} p_{k-j}), f - A(x_k + \sum_{j=0}^{t_k} \alpha_{k-j}^{(k)} p_{k-j}))_0 = \end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2}(f - Ax_k, f - Ax_k)_0 - \sum_{j=0}^{t_k} \alpha_{k-j}^{(k)} (f - Ax_k, Ap_{k-j})_0 + \\
&+ \frac{1}{2} \sum_{j=0}^{t_k} \sum_{l=0}^{t_k} \alpha_{k-j}^{(k)} \alpha_{k-l}^{(k)} (Ap_{k-j}, Ap_{k-l})_0 = \\
&= \frac{1}{2}(r_k, r_k)_0 - \sum_{j=0}^{t_k} \alpha_{k-j}^{(k)} (r_k, Ap_{k-j})_0 + \frac{1}{2} \sum_{j=0}^{t_k} \sum_{l=0}^{t_k} \alpha_{k-j}^{(k)} \alpha_{k-l}^{(k)} (Ap_{k-j}, Ap_{k-l})_0.
\end{aligned}
$$

Provedeme Gateauxovu derivaci podle $\alpha_{k-j}^{(k)}$ a pro $l = 0, ..., t_k$ dostaneme:

$$
\begin{aligned}
0 &= -(r_k, Ap_{k-l})_0 + \sum_{j=0}^{t_k} \alpha_{k-j}^{(k)} (Ap_{k-j}, Ap_{k-l})_0 = (-r_k + \sum_{j=0}^{t_k} \alpha_{k-j}^{(k)} Ap_{k-j}, Ap_{k-l})_0 = \\
&= (Ax_k - f + \sum_{j=0}^{t_k} \alpha_{k-j}^{(k)} Ap_{k-j}, Ap_{k-l})_0 = (Ax_{k+1} - f, Ap_{k-l})_0 = -(r_{k+1}, Ap_{k-l})_0.
\end{aligned}
$$

Zjistili jsme tedy, že

$$
(r_k, Ap_{k-l})_0 = \sum_{j=0}^{t_k} \alpha_{k-j}^{(k)} (Ap_{k-j}, Ap_{k-l})_0, \quad \text{pro} \quad l = 0, ..., t_k, \quad \text{kde} \tag{2.3}
$$

$$
r_k = f - Ax_k. \tag{2.4}
$$

Ze vztahu (2.2) plyne vhodné užití rekursivní formule pro výpočet residua r_{k+1}, kromě definovaného výrazu (2.4):

$$
r_{k+1} = r_k - \sum_{j=0}^{t_k} \alpha_{k-j}^{(k)} Ap_{k-j}. \tag{2.5}
$$

Podle výše dokázaného vztahu

$$
(r_{k+1}, Ap_{k-l})_0 = 0, \quad l = 0, ..., t_k \tag{2.6}
$$

uvažovaného ve tvaru $(r_k, Ap_{k-1-l})_0 = 0, \quad l = 0, ..., t_{k-1}$ a využitím předpokladu $t_k \leq t_{k-1} + 1$ lze rovnost (2.3) psát maticově takto:

$$
\Lambda_{t_k}^{(k)} \underline{\alpha}^{(k)} = \underline{\gamma}^{(k)}, \tag{2.7}
$$

kde

$$
\begin{aligned}
\Lambda^{(k)} := \Lambda_{t_k}^{(k)} &= [(Ap_{k-j}, Ap_{k-l})_0], \quad j, l = 0, ..., t_k \\
(\underline{\alpha}^{(k)})_j &= \alpha_{k-j}^{(k)}, \quad j = 0, ..., t_k \\
(\underline{\gamma}^{(k)})_j &= 0, \quad j = 1, ..., t_k \\
(\underline{\gamma}^{(k)})_0 &= (r_k, Ap_k)_0.
\end{aligned}
$$

Matice $\Lambda^{(k)}$ je regulární právě když je množina $\{Ap_{k-j}\}_{j=0}^{t_k}$ lineárně nezávislá, což je tehdy, když je množina $\{p_{k-j}\}_{j=0}^{t_k}$ lineárně nezávislá, neboť matice A je regulární.

Nyní budeme konstruovat hledané směry tak, aby byly lineárně nezávislé. V k-tém kroku budeme počítat nový hledaný směr jako lineární kombinaci residua r_{k+1}, které odpovídá nejnovější aproximaci x_{k+1}, a $(s_k + 1)$ předchozích hledaných směrů, tj.

$$p_{k+1} = r_{k+1} + \sum_{j=0}^{s_k} \beta_{k-j}^{(k)} p_{k-j}, \tag{2.8}$$

kde $s_k = \min\{k, s\}$, $s \geq 0$ a $s_0 = 0$ a kde parametry $\beta_{k-j}^{(k)}$, $j = 0, ..., s_k$ jsou určeny tak, že hledané směry splňují ortogonální podmínku

$$(Ap_{k+1}, Ap_{k-l})_1 = 0, \quad l = 0, ..., s_k. \tag{2.9}$$

Zde $(.,.)_1$ je skalární součin v $\mathbb{R}^N$ definovaný $(u, v)_1 = (u, M_1 v)$, kde M_1 je symetrická a positivně definitní matice.

Nový hledaný směr p_{k+1} se přidá do množiny hledaných směrů a je-li $s_k = s_{k-1}$ a $t_k = t_{k-1}$, pak nejstarší směr, tj. ten, který byl v této množině spočten jako první, z této množiny vyškrtneme, protože ho už nebudeme potřebovat.
(V praxi se obvykle volí $t_k = s_{k-1} + 1$ nebo $t_k = s_{k-1} + 2$.)

Vztah (2.9) platí také pro předchozí kroky a protože platí $s_k \leq s_{k-1} + 1$, pak platí obecně $(Ap_i, Ap_j)_1 = 0, \quad i \neq j, \quad i, j = k+1, k, ..., k - s_k$. Vztahy (2.8) a (2.9) implikují, že

$$\beta_{k-l}^{(k)} = -\frac{(Ar_{k+1}, Ap_{k-l})_1}{(Ap_{k-l}, Ap_{k-l})_1}, \quad l = 0, ..., s_k. \tag{2.10}$$

Jestliže zvolíme x_0 a p_0, pak vztahy (2.2), (2.7), (2.8) a (2.10) kompletně definují algoritmus. Počáteční vektor x_0 se volí libovolně a v souhlase se vztahem (2.8) pro $k = -1$ volíme $p_0 = r_0$.

Algoritmus 2.1

Zvolíme x_0 libovolně, $t \geq 1$, $s \geq 0$.
Spočteme $r_0 = f - Ax_0$.
Položíme $p_0 = r_0$.
Pro $k = 0, 1, 2, ...$ **provedeme**
$\Lambda_{t_k}^{(k)} \underline{\alpha}^{(k)} = \underline{\gamma}^{(k)}$, kde
$\Lambda_{t_k}^{(k)} = [(Ap_{k-j}, Ap_{k-l})_0], \quad j, l = 0, ..., t_k$
$(\alpha^{(k)})_j = \alpha_{k-j}^{(k)}, \quad j = 0, ..., t_k$
$(\gamma^{(k)})_j = 0, \quad j = 1, ..., t_k$
$(\gamma^{(k)})_0 = (r_k, Ap_k)_0$
$x_{k+1} = x_k + \sum_{j=0}^{t_k} \alpha_{k-j}^{(k)} p_{k-j}$, kde $1 \leq t_k \leq \min\{t_{k-1} + 1, t\}$, $t_0 = 0$
$r_{k+1} = r_k - \sum_{j=0}^{t_k} \alpha_{k-j}^{(k)} Ap_{k-j}$
$\beta_{k-l}^{(k)} = -\frac{(Ar_{k+1}, Ap_{k-l})_1}{(Ap_{k-l}, Ap_{k-l})_1}, \quad l = 0, ..., s_k$
$p_{k+1} = r_{k+1} + \sum_{j=0}^{s_k} \beta_{k-j}^{(k)} p_{k-j}$, kde $s_k = \min\{k, s\}$
`Konec cyklu pro` k.
Konec Algoritmu.

Metoda vycházející z tohoto algoritmu se nazývá „**zobecněná metoda sdružených gradientů ve smyslu nejmenších čtverců (GCG-LS(s))** *(Generalized Conjugate Gradient Least Square method)*“.

Obdobně jako v kapitole 1. lze odtud odvodit algoritmus s předpodmíněním, kde místo soustavy $Ax = f$ řešíme soustavu $\tilde{A}\tilde{x} = \tilde{f}$, kde $\tilde{A} = Q_1^{-1}AQ_2^{-1}$, $\tilde{x} = Q_2x$ a $\tilde{f} = Q_1^{-1}f$. Označení pro Λ, α, β a γ ponecháme. Pak po dodatečném dodefinování $\tilde{p} = Q_2p$ dostaneme tento algoritmus:

Algoritmus 2.2

Zvolíme x_0 libovolně, $t \geq 1$, $s \geq 0$.
Spočteme $r_0 = f - Ax_0$.
Položíme $p_0 = Q_2^{-1}Q_1^{-1}r_0$.
Pro $k = 0, 1, 2, ...$ **provedeme**
$\Lambda_{t_k}^{(k)}\underline{\alpha}^{(k)} = \underline{\gamma}^{(k)}$, kde
$\Lambda_{t_k}^{(k)} = [(Q_1^{-1}Ap_{k-j}, Q_1^{-1}Ap_{k-l})_0]$, $j, l = 0, ..., t_k$
$(\alpha^{(k)})_j = \alpha_{k-j}^{(k)}$, $j = 0, ..., t_k$
$(\gamma^{(k)})_j = 0$, $j = 1, ..., t_k$
$(\gamma^{(k)})_0 = (Q_1^{-1}r_k, Q_1^{-1}Ap_k)_0$
$x_{k+1} = x_k + \sum_{j=0}^{t_k} \alpha_{k-j}^{(k)}p_{k-j}$, kde $1 \leq t_k \leq \min\{t_{k-1} + 1, t\}$, $t_0 = 0$
$r_{k+1} = r_k - \sum_{j=0}^{t_k} \alpha_{k-j}^{(k)}Ap_{k-j}$
$\beta_{k-l}^{(k)} = -\frac{(Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_{k+1}, Q_1^{-1}Ap_{k-l})_1}{(Q_1^{-1}Ap_{k-l}, Q_1^{-1}Ap_{k-l})_1}$, $l = 0, ..., s_k$
$p_{k+1} = Q_2^{-1}Q_1^{-1}r_{k+1} + \sum_{j=0}^{s_k} \beta_{k-j}^{(k)}p_{k-j}$, kde $s_k = \min\{k, s\}$
`Konec cyklu pro` k.
Konec Algoritmu.

Metodu vycházející z tohoto algoritmu nazveme „**zobecněná metoda sdružených gradientů ve smyslu nejmenších čtverců (GCG-LS(s)) s předpodmíněním**“.

Samozřejmě můžeme tuto useknutou verzi algoritmu restartovat po každých m krocích. To znamená, že iterace x_m se položí jako nová počáteční hodnota x_0, příslušné residuum r_m se bude rovnat residuu r_0 a položením $k = 0$ začneme celý proces znovu od začátku položením $p_0 = Q_2^{-1}Q_1^{-1}r_0$.

Algoritmus 1.8 metody LSGCR je speciálním případem algoritmu 2.2 pro $M_0 = M_1 = I$, $s = 0$ a $t_k = k\ \forall k$, tedy pro $t = \infty$. Obdobně useknutý algoritmus 1.9 metody Axel(k) je speciálním případem algoritmu 2.2 opět pro $M_0 = M_1 = I$, $s = 0$ a t_k zvolené tak, aby pro aproximaci x_{k+1} byl potřeba stejný počet posledních vektorů p_{k-j}, jako pro aproximaci x_{i+1} metody Axel(k). V obou případech získáme stejné posloupnosti iterací.

Jediná možnost zhroucení algoritmu je, když bude matice $\Lambda^{(k)}$ singulární. Dokážeme, že to může nastat pouze v případě, že x_k je již řešením, tj. že x_k již minimalizuje funkcionál E.

Věta 2.1: Je-li $r_k \neq 0$, pak matice

$$\Lambda^{(k)} = [(Q_1^{-1}Ap_{k-j}, Q_1^{-1}Ap_{k-l})_0], \quad j, l = 0, ..., t_k$$

je regulární, jestliže $t_k \geq s_{k-1} + 1$.

Důkaz: Napišme si vztah (2.9) v předpodmíněném tvaru a k nahrazeným $k - 1$:

$$(Q_1^{-1}Ap_k, Q_1^{-1}Ap_{k-1-l})_1 = 0.$$

Vidíme, že vektor $Q_1^{-1}Ap_k$ je ortogonální na vektory $Q_1^{-1}Ap_{k-1-l}$, $l = 0, ..., s_{k-1}$.
Matice $\Lambda^{(k)}$ je singulární právě když je množina $\{Q_1^{-1}Ap_{k-l}\}_{l=0}^{t_k}$ lineárně závislá. Sporem

tedy předpokládejme, že tato množina je lineárně závislá, tj., že např. vektor $Q_1^{-1}Ap_k$ je lineární kombinací zbylých vektorů této množiny.
Dále napišme vztah (2.8) opět v předpodmíněném tvaru a k nahrazeným $k-1$:

$$p_k = Q_2^{-1}Q_1^{-1}r_k + \sum_{j=0}^{s_{k-1}} \beta_{k-1-j}^{(k-1)} p_{k-1-j}.$$

Odtud plyne, že $Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_k$ je lineární kombinací vektorů $\{Q_1^{-1}Ap_{k-l}\}_{l=0}^{s_{k-1}+1}$ a tedy také vektorů $\{Q_1^{-1}Ap_{k-l}\}_{l=1}^{t_k}$, neboť $t_k \geq s_{k-1}+1$ a $Q_1^{-1}Ap_k$ je lineární kombinací vektorů $Q_1^{-1}Ap_{k-1}$,...,$Q_1^{-1}Ap_{k-t_k}$ z předpokladu.
Existují tedy skaláry $\mu_{k-j}^{(k)}$, že

$$(Q_1^{-1}r_k, Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_k)_0 = (Q_1^{-1}r_k, \sum_{j=1}^{t_k} \mu_{k-j}^{(k)} Q_1^{-1}Ap_{k-j})_0.$$

Tedy (protože platí $t_k \leq t_{k-1}+1$) podle vztahu (2.6) napsaným pro $k-1$ platí:

$$(Q_1^{-1}r_k, Q_1^{-1}AQ_2^{-1}Q_1^{-1}r_k)_0 = 0, \quad \text{neboli} \quad (Q_1^{-1}r_k, (M_0Q_1^{-1}AQ_2^{-1})Q_1^{-1}r_k) = 0, \quad \text{neboli}$$

$$(Q_1^{-1}r_k, (M_0Q_1^{-1}AQ_2^{-1} + (Q_1^{-1}AQ_2^{-1})^T M_0)Q_1^{-1}r_k) = 0. \tag{2.11}$$

K důkazu posledního „neboli“: obecně

$$0 = (y, By) = (B^T y, y) = (y, B^T y) = 0 \quad \Rightarrow \quad (y, (B+B^T)y) = 0.$$

Podle předpokladu ze začátku této podkapitoly je $M_0A + A^TM_0$ positivně definitní, tj. v předpodmíněném tvaru je matice $M_0\tilde{A} + \tilde{A}^TM_0 = M_0Q_1^{-1}AQ_2^{-1} + (Q_1^{-1}AQ_2^{-1})^TM_0$ positivně definitní. Tedy vztah (2.11) implikuje, že $Q_1^{-1}r_k = 0$, a protože matice Q_1^{-1} je regulární, pak $r_k = 0$ a to je spor. Proto je množina $\{Q_1^{-1}Ap_{k-l}\}_{l=0}^{t_k}$ lineárně nezávislá. □

Pro lepší přehlednost a zjednodušení nebudeme psát další tvrzení v předpodmíněných tvarech, ale pouze v základním, tzn. nepředpodmíněném tvaru. Z postupů uvedených výše umíme uvedená tvrzení převést do předpodmíněného tvaru.

Lemma 2.1: Platí:

1. $(r_{k+1}, Ar_{l+1})_0 = 0 \quad \text{pro} \quad l = s_l + k - t_k, ..., k-1$
2. Je-li $t_k = k$, pak $(r_{k+1}, Ar_{l+1})_0 = 0 \quad \text{pro} \quad l = -1, ..., k-1$

Důkaz:

1. Podle (2.8) a (2.6) máme:

$$\begin{aligned}(r_{k+1}, Ar_{l+1})_0 &= (r_{k+1}, Ap_{l+1} - \sum_{j=0}^{s_l} \beta_{l-j}^{(k)} Ap_{l-j}) = \\ &= (r_{k+1}, Ap_{l+1}) - \sum_{j=0}^{s_l} \beta_{l-j}^{(k)} (r_{k+1}, Ap_{l-j}) = 0\end{aligned}$$

pro $l = s_l + k - t_k, ..., k-1$, neboť podle (2.6):

- První sčítanec je nulový pro $l+1 = k-t_k, ..., k$.
- Výrazy pod sumou jsou nulové pro $l = k-t_k, ..., k$; ... ; $l-s_l = k-t_k, ..., k$.
- Celkem $k-t_k \le l-s_l \le l \le k-1$ a tedy $l = s_l + k - t_k, ..., k-1$,

což dokazuje první tvrzení.

2. Je-li $t_k = k$, pak druhé tvrzení platí pro $l = 0, ..., k-1$, neboť platí $s_l \le l$.
Pro $l = -1$ máme: $(r_{k+1}, Ar_0)_0 = (r_{k+1}, Ap_0)_0 = 0$ podle (2.6) pro $l = t_k = k$. □

Protože platí

(2.12) $$(r_{k+1}, Ar_{l+1})_0 = 0 \quad \text{pro} \quad l = -1, ..., k-1,$$

je metoda GCG-LS(s) typu sdružených residuí (A-ortogonalita residuí).

Lemma 2.2:

1. Jestliže $t_k \ge s_{k-1} + 1$, pak platí

(2.13) $$\alpha_k^{(k)} = (\det \Lambda^{(k)})^{-1} \cdot \det(\Lambda_{k,0}^{(k)}) \cdot (r_k, Ar_k)_0,$$

kde $\Lambda_{k,0}^{(k)}$ je algebraický doplněk prvku na pozici (0,0) v matici $\Lambda^{(k)}$, tj. matice $\Lambda_{k,0}^{(k)}$ vznikne z matice $\Lambda^{(k)}$ vynecháním prvního řádku a sloupce. Tuto matici nazveme submaticí matice $\Lambda^{(k)}$.

2. Je-li $r_k \ne 0$, pak $\alpha_k^{(k)} > 0$.

Důkaz:

1. Z Cramerova pravidla plyne, že pro první složku ($j = 0$) řešení soustavy (2.7) platí

$$\alpha_k^{(k)} = (\det \Lambda^{(k)})^{-1} \cdot \sum_{i=0}^{t_k} (-1)^i \cdot \det(\Lambda_{k,i}^{(k)}) \cdot (\underline{\gamma}^{(k)})_i,$$

kde $\Lambda_{k,i}^{(k)}$ je submatice odpovídajícího prvku matice $\Lambda^{(k)}$ na pozici (0,i).
Podle (2.7) je

$$(\underline{\gamma}^{(k)})_i = 0 \quad \text{pro} \quad j = 1, ..., t_k$$

a

$$(\underline{\gamma}^{(k)})_0 = (r_k, Ap_k)_0.$$

Tedy

$$\alpha_k^{(k)} = (\det \Lambda^{(k)})^{-1} \cdot \det(\Lambda_{k,0}^{(k)}) \cdot (r_k, Ap_k)_0.$$

Ze vztahu (2.8) plyne, že

$$(r_k, Ap_k)_0 = (r_k, Ar_k)_0 - \sum_{j=0}^{s_{k-1}} \beta_{k-1-j}^{(k-1)} (r_k, Ap_{k-1-j})_0$$

a ze vztahu (2.6), že

$$(r_k, Ap_{k-1-j})_0 = 0 \quad \text{pro} \quad j = 0, ..., t_{k-1} \ge t_k - 1 \ge s_{k-1}$$

podle předpokladu. Z toho tedy plyne vztah

$$(r_k, Ap_k)_0 = (r_k, Ar_k)_0. \tag{2.14}$$

Odtud nakonec dostáváme

$$\alpha_k^{(k)} = (\det \Lambda^{(k)})^{-1} \cdot \det(\Lambda_{k,0}^{(k)}) \cdot (r_k, Ar_k)_0,$$

což dokončuje důkaz prvního tvrzení.

2. Matice $\Lambda^{(k)}$ je pro každé k symetrická a positivně definitní, tudíž $\det(\Lambda^{(k)}) > 0$. Uvažujme rozložení matice A na symetrickou a antisymetrickou část M a R. Víme, že platí $(u, Ru) = 0 \ \ \forall u$, což platí i v našem skalárním součinu

$$(u, Ru)_0 = 0 \ \ \forall u.$$

Toho nyní využijme:

$$\begin{aligned}(r_k, Ar_k)_0 &= (r_k, (M-R)r_k)_0 = \frac{1}{2}(r_k, (A+A^T)r_k)_0 = \\ &= \frac{1}{2}(r_k, Ar_k)_0 + \frac{1}{2}(Ar_k, r_k)_0 = \frac{1}{2}(r_k, M_0 Ar_k) + \frac{1}{2}(Ar_k, M_0 r_k) = \\ &= \frac{1}{2}(r_k, M_0 Ar_k) + \frac{1}{2}(r_k, A^T M_0 r_k) = \frac{1}{2}(r_k, (M_0 A + A^T M_0)r_k).\end{aligned}$$

Podle základního předpokladu je matice $M_0 A + A^T M_0$ positivně definitní a dále $r_k \neq 0$. Z toho plyne, že $(r_k, Ar_k)_0 > 0$.
Dosadíme-li zjištěné kladné výrazy do (2.13), zjistíme, že $\alpha_k^{(k)} > 0$. □

Na závěr této části vyslovíme větu o konvergenci.
Věta 2.2: Nechť opět $t_k \geq s_{k-1} + 1$. Pak platí:

1.

$$(r_{k+1}, r_{k+1})_0 = (r_k, r_k)_0 - \det(\Lambda^{(k)})^{-1} \cdot \det(\Lambda_{k,0}^{(k)}) \cdot (r_k, Ar_k)_0^2, \quad k = 0, 1, 2, ...$$

a je-li $r_k \neq 0$, pak metoda GCG-LS(s) konverguje monotonně, tj.

$$E(x_{k+1}) < E(x_k).$$

2.

$$\begin{aligned}(r_{k+1}, r_{k+1})_0 &= (r_k, r_k)_0 - \frac{(r_k, Ar_k)_0^2}{\min\limits_{g \in W_{k-1}} \| Ar_k - g \|_0^2} \ \leq \\ &\leq \left(1 - \frac{1}{\| [\frac{1}{2}(\mathcal{A}^{-1} + \mathcal{A}^{-T})]^{-1} \| \cdot \| [\frac{1}{2}(\mathcal{A} + \mathcal{A}^T)]^{-1} \|}\right) \cdot (r_k, r_k)_0,\end{aligned}$$

kde $W_{k-1} = sp(Ap_{k-1}, ..., Ap_{k-t_k})$, $\mathcal{A} = M_0^{\frac{1}{2}} A M_0^{-\frac{1}{2}}$ a $\| . \|$ odpovídá skalárnímu součinu $(.,.)^{\frac{1}{2}}$.
Odtud plyne, že $(r_k, r_k)_0 \to 0$ pro $k \to \infty$.

Důkaz:

1. Podle vztahu (2.5) vyjádříme residuum r_{k+1} a použijeme vztah (2.6):

$$\begin{aligned}(r_{k+1}, r_{k+1})_0 &= (r_{k+1}, r_k - \sum_{j=0}^{t_k} \alpha_{k-j}^{(k)} A p_{k-j})_0 = \\ &= (r_{k+1}, r_k)_0 = (r_k - \sum_{j=0}^{t_k} \alpha_{k-j}^{(k)} A p_{k-j}, r_k)_0.\end{aligned}$$

Protože víme, že $t_k - 1 \leq t_{k-1}$, použijeme opět (2.6) a vidíme, že skalární součin pod sumou bude nenulový pouze pro $j = 0$. Tedy

$$(r_{k+1}, r_{k+1})_0 = (r_k, r_k)_0 - \alpha_k^{(k)} (A p_k, r_k)_0. \tag{2.15}$$

Jestliže dosadíme za $(Ap_k, r_k)_0$ vztah (2.14) a za $\alpha_k^{(k)}$ vztah (2.13) v lemmatu 2.2, dostaneme první tvrzení věty.
Dále víme, že $\det(\Lambda^{(k)}) > 0$, neboť matice $\Lambda^{(k)}$ je symetrická a positivně definitní. Odtud platí, že

$$(r_{k+1}, r_{k+1})_0 < (r_k, r_k)_0, \quad \text{neboli} \quad E(x_{k+1}) < E(x_k).$$

2. Vyjdeme z první části této věty a dokážeme nejprve, že

$$\frac{\det \Lambda_{k,0}^{(k)}}{\det \Lambda^{(k)}} = \frac{1}{\min\limits_{g \in W_{k-1}} \| Ar_k - g \|_0^2} \tag{2.16}$$

Začneme **levou** stranou této rovnosti.
Ze vztahu (2.8) plyne, že

$$Ar_k = Ap_k - \sum_{j=0}^{s_{k-1}} \beta_{k-1-j}^{(k-1)} p_{k-1-j}$$

Nyní rozepíšeme $\det \Lambda^{(k)}$ a využijeme předpokladu $t_k \geq s_{k-1} + 1$:

$$\begin{aligned}\det \Lambda^{(k)} &= 1 \cdot \det \Lambda^{(k)} \cdot 1 = \\ &= \det \begin{pmatrix} 1 & -\beta_{k-1}^{(k-1)} & -\beta_{k-2}^{(k-1)} & \dots & -\beta_{k-1-s_{k-1}}^{(k-1)} & 0 & \dots & 0 \\ & 1 & 0 & \dots & 0 & 0 & \dots & 0 \\ & & 1 & & & & & \vdots \\ & & & \ddots & & & & \vdots \\ & & & & & 1 & & \vdots \\ & & & & & & 1 & \vdots \\ & & & & & & \ddots & 0 \\ & & & & & & & 1 \end{pmatrix} \cdot \\ &\cdot \quad \det \begin{pmatrix} (Ap_k, Ap_k)_0 & (Ap_{k-1}, Ap_k)_0 & \dots & (Ap_{k-t_k}, Ap_k)_0 \\ (Ap_k, Ap_{k-1})_0 & (Ap_{k-1}, Ap_{k-1})_0 & \dots & (Ap_{k-t_k}, Ap_{k-1})_0 \\ \vdots & \vdots & \ddots & \vdots \\ (Ap_k, Ap_{k-t_k})_0 & (Ap_{k-1}, Ap_{k-t_k})_0 & \dots & (Ap_{k-t_k}, Ap_{k-t_k})_0 \end{pmatrix} \cdot\end{aligned}$$

$$
\begin{aligned}
\cdot \quad & \det \begin{pmatrix}
1 & & & & & & & \\
-\beta_{k-1}^{(k-1)} & 1 & & & & & & \\
-\beta_{k-2}^{(k-1)} & 0 & \ddots & & & & & \\
\vdots & \vdots & & \ddots & & & & \\
-\beta_{k-1-s_{k-1}}^{(k-1)} & \vdots & & & \ddots & & & \\
0 & \vdots & & & & \ddots & & \\
\vdots & \vdots & & & & & 1 & \\
0 & 0 & \dots & \dots & \dots & \dots & 0 & 1
\end{pmatrix} = \\
= \quad & \det[(\%)\cdot(\%)\cdot(\%)] = \\
= \quad & \det \begin{pmatrix}
(Ar_k, Ar_k)_0 & (Ar_k, Ap_{k-1})_0 & \dots & (Ap_{k-t_k}, Ar_k)_0 \\
(Ar_k, Ap_{k-1})_0 & (Ap_{k-1}, Ap_{k-1})_0 & \dots & (Ap_{k-t_k}, Ap_{k-1})_0 \\
\vdots & \vdots & \ddots & \vdots \\
(Ar_k, Ap_{k-t_k})_0 & (Ap_{k-1}, Ap_{k-t_k})_0 & \dots & (Ap_{k-t_k}, Ap_{k-t_k})_0
\end{pmatrix} =: \\
=: \quad & \det \Omega^{(k)}.
\end{aligned}
$$

Protože součin determinantů je roven determinantu součinu matic, použili jsme pro matice symbol %, který značí „to samé“.
Nyní rozvineme $\det \Omega^{(k)}$ podle 1. řádku a protože platí $\Lambda_{k,0}^{(k)} = \Omega_{k,0}^{(k)}$, dostaneme:

$$
\det \Lambda^{(k)} = \det \Omega^{(k)} = (Ar_k, Ar_k)_0 \cdot \det \Lambda_{k,0}^{(k)} + \sum_{j=1}^{t_k} (-1)^j \cdot (Ar_k, Ap_{k-j}) \cdot \det \Omega_{k,j}^{(k)}
$$

Odtud plyne, že

$$
\frac{\det \Lambda^{(k)}}{\det \Lambda_{k,0}^{(k)}} = (Ar_k, Ar_k)_0 + \sum_{j=1}^{t_k} (-1)^j \cdot (Ar_k, Ap_{k-j}) \cdot \frac{\det \Omega_{k,j}^{(k)}}{\det \Lambda_{k,0}^{(k)}}
$$

Teď přejdeme k **pravé** straně rovnosti výše.
Nechť $g \in W_{k-1} = sp(Ap_{k-1}, ..., Ap_{k-t_k})$. Pak $g = \sum_{j=1}^{t_k} \mu_j Ap_{k-j}$.
Označme $F(\mu) = \| Ar_k - g \|_0^2$ a upravujme.

$$
\begin{aligned}
F(\mu) \quad &= \quad (Ar_k - \sum_{j=1}^{t_k} \mu_j Ap_{k-j}, Ar_k - \sum_{j=1}^{t_k} \mu_j Ap_{k-j})_0 = \\
&= \quad (Ar_k, Ar_k)_0 - 2 \cdot \sum_{j=1}^{t_k} \mu_j (Ar_k, Ap_{k-j})_0 + \sum_{j=1}^{t_k} \sum_{l=1}^{t_k} \mu_j \mu_l (Ap_{k-j}, Ap_{k-l})_0.
\end{aligned}
$$

Protože hledáme minimum $F(\mu)$, provedeme Gateauxovu derivaci $F'(\mu) = 0$ a pro $l = 1, ..., t_k$ dostaneme:

$$
(Ar_k, Ap_{k-l})_0 = \sum_{j=1}^{t_k} \mu_j (Ap_{k-j}, Ap_{k-l})_0.
$$

Z této rovnosti vypočítáme μ_j pomocí Cramerova pravidla a řešení označíme jako $\mu_j^\star$. Tedy

$$
\mu_1 (Ap_{k-1}, Ap_{k-1})_0 + \mu_2 (Ap_{k-2}, Ap_{k-1})_0 + \ldots + \mu_{t_k} (Ap_{k-t_k}, Ap_{k-1})_0 \quad =
$$

$$
\begin{aligned}
&= (Ar_k, Ap_{k-1})_0; \\
\mu_1(Ap_{k-1}, Ap_{k-2})_0 + \mu_2(Ap_{k-2}, Ap_{k-2})_0 + \ldots + \mu_{t_k}(Ap_{k-t_k}, Ap_{k-2})_0 &= \\
&= (Ar_k, Ap_{k-2})_0; \\
&\vdots \\
\mu_1(Ap_{k-1}, Ap_{k-t_k})_0 + \mu_2(Ap_{k-2}, Ap_{k-t_k})_0 + \ldots + \mu_{t_k}(Ap_{k-t_k}, Ap_{k-t_k})_0 &= \\
&= (Ar_k, Ap_{k-t_k})_0,
\end{aligned}
$$

neboli

$$
\begin{pmatrix}
(Ap_{k-1}, Ap_{k-1})_0 & (Ap_{k-2}, Ap_{k-1})_0 & \ldots & (Ap_{k-t_k}, Ap_{k-1})_0 \\
(Ap_{k-1}, Ap_{k-2})_0 & (Ap_{k-2}, Ap_{k-2})_0 & \ldots & (Ap_{k-t_k}, Ap_{k-2})_0 \\
\vdots & \vdots & \ddots & \vdots \\
(Ap_{k-1}, Ap_{k-t_k})_0 & (Ap_{k-2}, Ap_{k-t_k})_0 & \ldots & (Ap_{k-t_k}, Ap_{k-t_k})_0
\end{pmatrix} \cdot \begin{pmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_{t_k} \end{pmatrix} =
$$

$$
= \begin{pmatrix} (Ar_k, Ap_{k-1})_0 \\ (Ar_k, Ap_{k-2})_0 \\ \vdots \\ (Ar_k, Ap_{k-t_k})_0 \end{pmatrix}
$$

Z Cramerova pravidla a ze srovnání s maticemi $\Omega_{k,j}^{(k)}$ plyne, že

$$
\mu_j^\star = (-1)^j \cdot \frac{-\det \Omega_{k,j}^{(k)}}{\det \Lambda_{k,0}^{(k)}}, \quad j = 1, ..., t_k.
$$

Toto řešení $\mu_j^\star$, které splňuje rovnost

$$
(Ar_k, Ap_{k-l})_0 = \sum_{j=1}^{t_k} \mu_j^\star (Ap_{k-j}, Ap_{k-l})_0,
$$

dosadíme do $F(\mu)$ a dostaneme:

$$
\begin{aligned}
F(\mu) &= (Ar_k, Ar_k)_0 - 2 \cdot \sum_{j=1}^{t_k} \mu_j (Ar_k, Ap_{k-j})_0 + \sum_{j=1}^{t_k} \sum_{l=1}^{t_k} \mu_j \mu_l (Ap_{k-j}, Ap_{k-l})_0 = \\
&= (Ar_k, Ar_k)_0 - 2 \cdot \sum_{j=1}^{t_k} (-1)^j \cdot \frac{-\det \Omega_{k,j}^{(k)}}{\det \Lambda_{k,0}^{(k)}} \cdot (Ar_k, Ap_{k-j})_0 + \\
&+ \sum_{j=1}^{t_k} (-1)^j \cdot \frac{-\det \Omega_{k,j}^{(k)}}{\det \Lambda_{k,0}^{(k)}} \cdot (Ar_k, Ap_{k-j})_0 = \\
&= (Ar_k, Ar_k)_0 + \sum_{j=1}^{t_k} (-1)^j \cdot \frac{\det \Omega_{k,j}^{(k)}}{\det \Lambda_{k,0}^{(k)}} \cdot (Ar_k, Ap_{k-j})_0.
\end{aligned}
$$

Porovnáme-li nyní vztahy, které jsme dostali po úpravách **levé** a **pravé** strany rovnosti (2.16), zjistíme, že skutečně platí

$$
\frac{\det \Lambda_{k,0}^{(k)}}{\det \Lambda^{(k)}} = \frac{1}{\min\limits_{g \in W_{k-1}} \| \, Ar_k - g \, \|_0^2}
$$

Odtud plyne

$$\begin{aligned}(r_{k+1},r_{k+1})_0 &= (r_k,r_k)_0 - \det(\Lambda^{(k)})^{-1}\cdot\det(\Lambda^{(k)}_{k,0})\cdot(r_k,Ar_k)_0^2 = \\ &= (r_k,r_k)_0 - \frac{1}{\min\limits_{g\in W_{k-1}} \parallel Ar_k - g \parallel_0^2}\cdot(r_k,Ar_k)_0^2.\end{aligned}$$

Vezmeme-li $g=0$, pak dostaneme vztah

$$(2.17)\qquad (r_{k+1},r_{k+1})_0 \le (r_k,r_k)_0 - \frac{(r_k,Ar_k)_0^2}{\parallel Ar_k \parallel_0^2} = (r_k,r_k)_0 - \frac{(r_k,Ar_k)_0^2}{(Ar_k,Ar_k)_0}$$

Nyní budeme upravovat a odhadovat zlomek vpravo. Připomínáme, že M a R značí symetrickou a antisymetrickou část matice A.
Nejprve vezmeme **čitatel**.

$$\begin{aligned}(r_k,Ar_k)_0 &= (r_k,(M-R)r_k)_0 = (r_k,Mr_k)_0 = \frac{1}{2}(r_k,(A+A^T)r_k)_0 = \\ &= \frac{1}{2}(r_k,Ar_k)_0 + \frac{1}{2}(Ar_k,r_k)_0 = \frac{1}{2}(r_k,M_0Ar_k) + \frac{1}{2}(r_k,A^TM_0r_k) = \\ &= \frac{1}{2}(r_k,M_0^{\frac12}M_0^{\frac12}AM_0^{-\frac12}M_0^{\frac12}r_k) + \frac{1}{2}(r_k,M_0^{\frac12}M_0^{-\frac12}A^TM_0^{\frac12}M_0^{\frac12}r_k) = \\ &= \frac{1}{2}(r_k,M_0^{\frac12}(M_0^{\frac12}AM_0^{-\frac12} + M_0^{-\frac12}A^TM_0^{\frac12})M_0^{\frac12}r_k) = \\ &= \frac{1}{2}(r_k,M_0^{\frac12}(\mathcal{A}+\mathcal{A}^T)M_0^{\frac12}r_k) = (M_0^{\frac12}r_k,\frac{1}{2}(\mathcal{A}+\mathcal{A}^T)M_0^{\frac12}r_k) \ge \\ &\ge \lambda_{\min}(\frac{1}{2}(\mathcal{A}+\mathcal{A}^T))\cdot(M_0^{\frac12}r_k,M_0^{\frac12}r_k) = \\ &= \parallel(\frac{1}{2}(\mathcal{A}+\mathcal{A}^T))^{-1}\parallel^{-1}\cdot(M_0^{\frac12}r_k,M_0^{\frac12}r_k) = \\ &= \parallel(\frac{1}{2}(\mathcal{A}+\mathcal{A}^T))^{-1}\parallel^{-1}\cdot(r_k,r_k)_0,\end{aligned}$$

neboť obecně pro symetrickou matici B platí

$$\parallel B \parallel = \lambda_{\max}(B) \quad\Rightarrow\quad \parallel B^{-1}\parallel^{-1} = \lambda_{\min}(B).$$

Dále upravíme **jmenovatel**.

$$\begin{aligned}(Ar_k,Ar_k)_0 &= (r_k,A^TM_0Ar_k) = (r_k,M_0^{\frac12}(M_0^{-\frac12}A^TM_0^{\frac12})(M_0^{\frac12}AM_0^{-\frac12})M_0^{\frac12}r_k) = \\ &= (r_k,M_0^{\frac12}\mathcal{A}^T\mathcal{A}M_0^{\frac12}r_k)\end{aligned}$$

Nyní budeme potřebovat jinou úpravu výrazu (r_k,Ar_k). Vyjdeme z toho, co už o tomto skalárním součinu víme.

$$\begin{aligned}(r_k,Ar_k)_0 &= \frac{1}{2}(r_k,M_0^{\frac12}(\mathcal{A}+\mathcal{A}^T)M_0^{\frac12}r_k) = \frac{1}{2}(r_k,M_0^{\frac12}\mathcal{A}^T(\mathcal{A}^{-1}+\mathcal{A}^{-T})\mathcal{A}M_0^{\frac12}r_k) \ge \\ &\ge \parallel(\frac{1}{2}(\mathcal{A}^{-1}+\mathcal{A}^{-T}))^{-1}\parallel^{-1}\cdot(r_k,M_0^{\frac12}\mathcal{A}^T\mathcal{A}M_0^{\frac12}r_k).\end{aligned}$$

Odtud je

$$(Ar_k,Ar_k)_0 = (r_k,M_0^{\frac12}\mathcal{A}^T\mathcal{A}M_0^{\frac12}r_k) \le \parallel(\frac{1}{2}(\mathcal{A}^{-1}+\mathcal{A}^{-T}))^{-1}\parallel\cdot(r_k,Ar_k)_0.$$

Teď již můžeme upravit celý výraz (2.17).

$$\begin{aligned}
(r_{k+1}, r_{k+1})_0 &\leq (r_k, r_k)_0 - \frac{(r_k, Ar_k)_0^2}{(Ar_k, Ar_k)_0} \leq \\
&\leq (r_k, r_k)_0 - \frac{(r_k, Ar_k)_0 \cdot (r_k, Ar_k)_0}{\| (\frac{1}{2}(\mathcal{A}^{-1} + \mathcal{A}^{-T}))^{-1} \| \cdot (r_k, Ar_k)_0} \leq \\
&\leq (r_k, r_k)_0 - \frac{\| (\frac{1}{2}(\mathcal{A} + \mathcal{A}^T))^{-1} \|^{-1} \cdot (r_k, r_k)_0}{\| (\frac{1}{2}(\mathcal{A}^{-1} + \mathcal{A}^{-T}))^{-1} \|} = \\
&= \left(1 - \frac{1}{\| [\frac{1}{2}(\mathcal{A}^{-1} + \mathcal{A}^{-T})]^{-1} \| \cdot \| [\frac{1}{2}(\mathcal{A} + \mathcal{A}^T)]^{-1} \|}\right) \cdot (r_k, r_k)_0,
\end{aligned}$$

a tím jsme u konce důkazu. □

Poznámka: Uvažujeme-li plnou verzi algoritmu, tj. $t_k = k$, pak platí

$$\det \Lambda_{k,0}^{(k)} = \det \Lambda^{(k-1)}.$$

2.3 Ukončení procesu

V této části budeme zkoumat, kdy $\| r_k \|_0 \to 0$ pro plnou verzi algoritmu, kde $t_k = k$. Z rovnosti $p_0 = r_0$ a z konstrukce směrů p_k a r_k podle vzorců (2.8) a (2.5) plyne, že směr p_k je lineární kombinací vektorů $A^j r_0, \quad j = 0, ..., k$. Definujeme proto Krylovovu množinu

$$\mathcal{K}_k = \mathcal{K}_k(r_0) = sp(r_0, Ar_0, ..., A^k r_0).$$

Podobně je residuum r_k lineární kombinací r_0 a vektorů z množiny $A\mathcal{K}_{k-1}$, tedy r_k je direktní součet $r_k = r_0 \oplus A\mathcal{K}_{k-1}$. Je tedy

$$r_k = (I + q_k(A))r_0, \quad k = 0, 1, 2, ... \tag{2.18}$$

pro nějaký polynom q_k splňující $q_k(0) = 0$. Pro chybový funkcionál E tedy platí:

$$E(x_k) = \frac{1}{2}(r_k, r_k)_0 = \frac{1}{2} \| (I + q_k(A))r_0 \|_0^2 .$$

Různé volby posloupností t_k a s_k dávají různé hodnoty $\alpha_{k-j}^{(k)}$ a $\beta_{k-j}^{(k)}$ a tedy i různé polynomy q_k. Protože je $t_k = k, \quad k = 0, 1, ...,$ pak

$$(r_k, r_k)_0 = \min_{q_k \in P_k^0} \| (I + q_k(A))r_0 \|_0^2, , \tag{2.19}$$

kde P_k^0 značí množinu polynomů q_k stupně nejvýše k takových, že $q_k(0) = 0$.

Nyní vyslovíme nějaké definice z maticové algebry a na závěr vyslovíme hlavní větu této podkapitoly.

Definice 2.1: Polynom $g(\lambda)$ stupně ≥ 1 se nazývá anihilační polynom (*annihilating polynomial*) matice B, jestliže $g(B) = 0$.

Budeme uvažovat pouze reálné polynomy. Zde poznamenáváme, že

$$g(\lambda) = \det(B - \lambda I)$$

je anihilační reálný polynom stupně n, jestliže B je reálná matice řádu n.
Definice 2.2: Anihilační polynom matice B nejnižšího stupně se nazývá minimální polynom k B (*minimal polynomial to* B). Jeho stupeň označíme číslem $m(B)$.

Platí, že $m(B) \leq n$. Je-li B regulární a je-li koeficient nejnižšího řádu minimálního polynomu označeného jako $\mu_m(\lambda)$ nulový, pak

$$B^{-1}\mu_m(B) = 0 \quad \text{a} \quad g_{m-1}(\lambda) = \lambda^{-1}\mu_m(\lambda)$$

by byl anihilační polynom stupně $m-1$. Proto je koeficient nejnižšího řádu nenulový. Navíc budeme předpokládat, že tento koeficient je roven jedné, čehož lze dosáhnout normalizací minimálního polynomu. Minimální polynom je pak jediný.
Definice 2.3: Polynom $g(\lambda)$ takový, že $g(B)r_0 = 0$, se nazývá minimální polynom k B a r_0. Jeho stupeň označíme $m(B, r_0)$.

Platí $m(B, r_0) \leq m(B)$, neboť: Platí-li $g(B) = 0$, kde g je polynom nejnižšího stupně n, pro který to platí, pak platí také $g(B)r_0 = 0$ pro libovolný vektor r_0. Ale rovnost $g(B)r_0 = 0$ může nastat pro jiný polynom $\hat{g}$ stupně menšího než n. Pokud ne, pak to určitě nastane pro polynom g, jehož stupeň je n. Je tedy $m(B) = n$ a platí $m(B, r_0) \leq m(B)$.

Je-li matice B regulární, pak opět minimální polynom k B a r_0 normujeme a tento polynom bude jediný.
Nyní se vrátíme k řešení systému $Ax = f$ a vyslovíme větu.
Věta 2.3: Jestliže g_m je minimální polynom k A a r_0 a má minimální stupeň $m = m(A, r_0)$, pak plná verze metody GCG-LS, tj. pro $t_k = k$, dává nulové residuum po m krocích.
Důkaz: Podle definice 2.3 platí

$$g_m(A)r_0 = 0. \tag{2.20}$$

Protože je tento polynom normalizovaný, pak je $g_m(0) = 1$. Ze vztahu (2.20) plyne, že pro nějaký polynom $\hat{q}_m$ stupně m takový, že $\hat{q}_m(0) = 0$, tj. pro nějaký polynom $\hat{q}_m$, který má nulový člen nejnižšího řádu, platí $r_0 + \hat{q}_m(A)r_0 = 0$, neboť $I + \hat{q}_m(A) = g_m(A)$. Tedy podle (2.19) dostáváme:

$$0 \leq (r_m, r_m)_0 = \min_{q_m \in P_m^0} \| (I + q_m(A))r_0 \|_0^2 = 0, \tag{2.21}$$

kde minimální hodnota nastává pro $q_m = \hat{q}_m$. Je proto $r_m = 0$ a m je nejmenší číslo, pro které se to může stát. Algoritmus tedy končí s residuem rovným nule po $m = m(A, r_0)$ krocích. □

2.4 Useknutá verze

Zvolíme-li v praxi $t_k = k \ \forall k$, dostaneme příliš nákladnou metodu co se týče požadavku na uložení vektorů i výpočetní složitosti. Uvažujme proto nyní následující useknutou verzi algoritmu. Volíme

$$t_k = \min(k, t), \quad s_k = \min(k, s) \ \text{ a } \ t = s + 2.$$

Předpokládejme rovněž, že $(.,.)_0 = (.,.)_1$, tj. že $M_0 = M_1$. Pak ze vztahů (2.7) a (2.9), tj. ze vztahů

$$\Lambda^{(k)} = (Ap_{k-j}, Ap_{k-l})_0, \quad j, l = 0, 1, ..., t_k, \quad \text{kde} \quad t_k = \min(k, t) = \min(k, s + 2)$$

a

$$(Ap_{k+1}, Ap_{k-l})_1 = (Ap_{k+1}, Ap_{k-l})_0 = 0, \quad l = 0, 1, ..., s_k, \quad \text{kde} \quad s_k = \min(k, s)$$

plyne, že matice $\Lambda^{(k)}$ má takovouto strukturu:

$$\begin{pmatrix} x & 0 & \dots & 0 & 0 \\ 0 & x & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & x & 0 \\ 0 & 0 & \dots & 0 & x \end{pmatrix} \quad \text{velikosti} \quad (k+1) \times (k+1) \quad \text{pro} \quad k = 0, 1, ..., s+1$$

a

$$\begin{pmatrix} x & 0 & \dots & 0 & x \\ 0 & x & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & x & 0 \\ x & 0 & \dots & 0 & x \end{pmatrix} \quad \text{velikosti} \quad (s+3) \times (s+3) \quad \text{pro} \quad k = s+2, s+3, ...$$

Symbol x značí nenulové prvky matice. Protože pravá strana $\underline{\gamma}^{(k)}$ rovnice (2.7) je nulová kromě první složky, plyne odtud, že

$$\alpha_{k-j}^{(k)} = 0, \quad j = 1, ..., t_k - 1.$$

Dále pro $t_k = \min(k, t) = \min(k, s+2)$ *a* $s_{k-1} = \min(k-1, s)$ platí nerovnost

$$\min(k, s+2) \ \geq \ \min(k-1, s) + 1$$

a je tedy splněn předpoklad lemmatu 2.2, $t_k \geq s_{k-1} + 1$. Odtud plyne, že $\alpha_k^{(k)} > 0$. O zbývajícím koeficientu $\alpha_{k-t_k}^{(k)}$ nic nevíme, může být buď nulový nebo nenulový. Člen $\alpha_{k-t_k}^{(k)} p_{k-t_k}$ lze proto ve vztahu (2.2) použít jako „kontrolní člen“. Jeho relativní velikost ve srovnání se členem $\alpha_k^{(k)} p_k$ může naznačit, zda se vyplatí useknutí a tím zanedbání členů $\alpha_{k-j}^{(k)} p_{k-j}$, $j = t_k + 1, t_k + 2, ...$ nebo ne. Jestliže relativní norma kontrolního členu zůstává malá během iterací, lze očekávat, že zanedbané členy mají malý vliv. Platí tedy toto lemma:

Lemma 2.3: Nechť $M_0 = M_1$. Pak pro algoritmus metody GCG-LS(s), kde jsou koeficienty $\beta_{k-j}^{(k)}$ určeny podle vztahu (2.10), platí

$$\alpha_{k-j}^{(k)} = 0, \quad j = 1, ..., t_k - 1 \quad \text{a} \quad \alpha_k^{(k)} > 0.$$

V této podkapitole dále najdeme podmínky, pro které se plná verze metody GCG-LS rovná useknuté verzi metody GCG-LS(s), tj. bude nás hlavně zajímat správná hodnota s, pro kterou obě verze dávají stejnou posloupnost aproximací. Začneme nějakými vlastnostmi z maticové algebry (viz [Gant]).

Lemma 2.4: Následující tvrzení jsou ekvivalentní:

1. Pro matici A a její hermitovskou A^H platí: $A^H A = A A^H$.
2. Matici A lze diagonalizovat maticí U, tj. $U^H A U = D$, kde D je diagonální matice a U je unitární matice, tj. $U^H U = I$.

3. Existuje polynom q takový, že platí: $A^H = q(A)$. Tento polynom se nazývá normální polynom.

Matice, pro kterou platí alespoň jedno z těchto tvrzení, se nazývá normální matice.

Nyní to zobecníme tímto způsobem. Nechť A je diagonalizovatelná regulární, ale ne nutně unitární maticí S, tj. $S^{-1}AS = D$, kde D je diagonální matice. Označme $H = S^{-H}S^{-1}$, což je hermitovská a positivně definitní matice, neboť:

- $H^H = (S^{-H}S^{-1})^H = S^{-H}S^{-1} = H$
- $(x, Hx) = (x, S^{-H}S^{-1}x) = (S^{-1}x, S^{-1}x) \geq 0.$

Platí

$$HAH^{-1} = S^{-H}DS^H \quad \text{a} \quad H^{-1}A^HH = (HAH^{-1})^H = SD^HS^{-1}. \tag{2.22}$$

Protože D je diagonální, je také normální, $D^HD = DD^H$. Podle lemmatu 2.4 tedy existuje polynom q takový, že $D^H = q(D)$. Dostáváme, že

$$H^{-1}A^HH = Sq(D)S^{-1} = Sq(S^{-1}AS)S^{-1} = q(SS^{-1}ASS^{-1}) = q(A) \tag{2.23}$$

dosazením do vztahu (2.22).

Lemma 2.5: Je-li matice reálná, pak normální polynom je reálný.

Definice 2.4: Matici $A' = H^{-1}A^HH$ nazveme H-adjungovaná matice k matici A.

Je-li matice A diagonalizovatelná, pak H-adjungovanou matici A' matice A lze vyjádřit jako polynom v matici A, což plyne ze vztahu (2.23).

Lemma 2.6: Buď H hermitovská a positivně definitní matice. Pak následující tvrzení jsou ekvivalentní:

1. Pro matici A' platí: $A'A = AA'$.
2. Matice A je diagonalizovatelná.
3. Existuje polynom q takový, že platí: $A' = q(A)$.

Nyní zavedeme skalární součin definovaný hermitovskou a positivně definitní maticí H: $< u, v > = (u, Hv)$.

Definice 2.5: Jestliže matice A komutuje se svou H-adjungovanou maticí A' vzhledem ke skalárnímu součinu $< ., . >$, pak řekneme, že A je H-normální matice.

Je-li matice A H-normální, pak podle lemmatu 2.4 existuje polynom q takový, že $A' = q(A)$.

Definice 2.6: Nechť A je H-normální matice. Pak takový polynom $\hat{q}$, pro který platí $A' = \hat{q}(A)$ a má nejnižší možný stupeň, nazveme H-normální polynom k matici A. Jeho stupeň označíme $n(A, H)$ a nazveme H-normální stupeň.

Definice 2.7: Buď g libovolný polynom. Pak polynom $\hat{q}$, pro který platí

$$A'g(A)r_0 = \hat{q}(A)g(A)r_0 \tag{2.24}$$

a má nejnižší možný stupeň $n(A, H, r_0)$, nazveme H-normální polynom k matici A vzhledem k r_0. Číslo $n(A, H, r_0)$ nazveme H-normální stupeň vzhledem k r_0 a matici A, pro kterou platí rovnost (2.24), nazveme H-normální matice vzhledem k r_0.

Platí $n(A, H, r_0) \leq n(A, H)$, neboť je-li matice A H-normální a je řádu N, pak je H-normální vzhledem k libovolnému vektoru r_0 dimense N.

Nyní se již dostáváme k hlavní větě této podkapitoly.

Věta 2.4: Nechť platí:

1. $M_0 = M_1$ je symetrická a positivně definitní matice;
2. A je M_0-normální matice;
3. $n = n(A, M_0, r_0)$ je M_0-normální stupeň vzhledem k r_0;
4. $s \geq n(A, M_0, r_0) - 1$;
5. „kontrolní člen“ $\alpha_{k-t_k}^{(k)} p_{k-t_k}$ je roven nule.

Pak useknutý algoritmus metody GCG-LS(s) je identický s plnou verzí.
Důkaz: Budeme uvažovat plnou verzi algoritmu, kde koeficienty $\beta_{k-l}^{(k)}$ jsou určeny ze vztahu (2.10) a zjistíme, od kterého indexu jsou již rovny nule. Totéž provedeme s koeficienty $\alpha_{k-l}^{(k)}$.
Podle vztahů (2.8) a (2.5) je p_{k-l} lineární kombinací vektorů $A^j r_0$, $j = 0, ..., k$, platí tedy

$$p_{k-l} \in K_{k-l} = sp(r_0, Ar_0, ..., A^{k-l} r_0) \quad \Rightarrow \quad p_{k-l} = q_{k-l}(A) r_0,$$

kde q_{k-l} je nějaký polynom stupně $k - l$.
Protože matice A je M_0-normální, platí $A'A = AA'$, kde $A' = M_0^{-1} A^H M_0$. Odtud dostáváme, že

$$\begin{aligned}(Ar_{k+1}, Ap_{k-l})_1 &= (Ar_{k+1}, Ap_{k-l})_0 = (Ar_{k+1}, M_0 Ap_{k-l}) = (r_{k+1}, A^H M_0 Ap_{k-l}) = \\ &= (r_{k+1}, M_0 A' Ap_{k-l}) = (r_{k+1}, A' Ap_{k-l})_0 = (r_{k+1}, A' Ap_{k-l})_1 = \\ &= (r_{k+1}, A' Aq_{k-l}(A) r_0)_1 = (r_{k+1}, AA' q_{k-l}(A) r_0)_1\end{aligned}$$

a protože matice A je M_0-normální, tedy M_1-normální podle předpokladu, je proto M_1-normální vzhledem k r_0, pak

$$(r_{k+1}, AA' q_{k-l}(A) r_0)_1 = (r_{k+1}, A\hat{q}_n(A) q_{k-l}(A) r_0)_1$$

podle (2.24), kde $\hat{q}_n$ je M_0-normální polynom k A vzhledem k r_0 a má M_0-normální stupeň $n = n(A, M_0, r_0)$. Odtud je tedy

$$(Ar_{k+1}, Ap_{k-l})_1 = (r_{k+1}, A\hat{q}_n(A) q_{k-l}(A) r_0)_1. \tag{2.25}$$

Podle vztahu (2.6) platí pro plnou verzi algoritmu, tj. pro $t_k = k$, že

$$(r_{k+1}, Ap_{k-l})_0 = 0, \quad l = 0, ..., k,$$

kde však stejně jako výše $p_{k-l} = q_{k-l}(A) r_0$, $l = 0, ..., k$. Platí tedy

$$(r_{k+1}, Aq_0(A) r_0)_0 = 0, \ (r_{k+1}, Aq_1(A) r_0)_0 = 0, ..., (r_{k+1}, Aq_k(A) r_0)_0 = 0,$$

neboli jednoduše

$$(r_{k+1}, Aq_k(A) r_0)_0 = 0 \tag{2.26}$$

pro libovolný polynom q_k stupně k či menšího.
Protože $M_0 = M_1$, tj. $(.,.)_0 = (.,.)_1$, plyne ze vztahů (2.25) a (2.26), že

$$(Ar_{k+1}, Ap_{k-l})_1 = (r_{k+1}, A(\hat{q}_n(A) q_{k-l}(A)) r_0)_1 = 0, \tag{2.27}$$

jestliže polynom $\hat{q}_n(A)q_{k-l}(A)$ je stupně k nebo menšího, tedy rovnost (2.27) platí pro

$$n + k - l \leq k, \quad \text{tj. pro} \quad l \geq n.$$

Ze vztahu (2.10) pro koeficienty $\beta_{k-l}^{(k)}$ plyne, že

$$\beta_{k-l}^{(k)} = 0 \quad \text{pro} \quad l \geq n, \quad \text{tj.}$$

$$\text{obecně} \quad \beta_k^{(k)} \neq 0, \quad \beta_{k-1}^{(k)} \neq 0, \ \ldots, \beta_{k-n+1}^{(k)} \neq 0; \quad \text{ale} \quad \beta_{k-n}^{(k)} = 0, \quad \beta_{k-n-1}^{(k)} = 0, \ \ldots$$

To tedy znamená, že v rovnici (2.8) napsané ve tvaru

$$p_{k+1} = r_{k+1} + \sum_{j=0}^{s} \beta_{k-j}^{(k)} p_{k-j}$$

se pod sumou sčítají všechny obecně nenulové členy, jakmile $s \geq n - 1$.
Z předpokladu $M_0 = M_1$ dále plyne, že matice $\Lambda^{(k)}$ definovaná vztahem (2.7) má strukturu uvedenou na začátku této podkapitoly. Odtud proto plyne, že

$$\alpha_{k-j}^{(k)} = 0, \quad j = 1, 2, ..., t_k - 1 \quad \text{a} \quad \alpha_k^{(k)} > 0.$$

Podle předpokladu nulovosti „kontrolního členu“ je dále $\alpha_{k-t_k}^{(k)} = 0$, neboť směry p_{k-t_k} se konstruují $A^T A$-ortogonálně, vztah (2.9), a nemohou tedy být nulové. Odtud rovnice (2.2) získá tvar

$$x_{k+1} = x_k + \alpha_k^{(k)} p_k.$$

Tím jsme dokázali, že algoritmus metody GCG-LS je identický s useknutým algoritmem metody GCG-LS(s) pro $s \geq n - 1$. □
Na závěr můžeme useknutý algoritmus za předpokladů věty 2.4 zkompletovat.

Algoritmus 2.3

Zvolíme x_0 libovolně, $s \geq 0$.
Spočteme $r_0 = f - Ax_0$.
Položíme $p_0 = Q_2^{-1} Q_1^{-1} r_0$.
Pro $k = 0, 1, 2, \ldots$ **provedeme**
$\alpha_k^{(k)} = \frac{(Q_1^{-1} r_k, Q_1^{-1} A p_k)_0}{(Q_1^{-1} A p_k, Q_1^{-1} A p_k)_0}$
$x_{k+1} = x_k + \alpha_k^{(k)} p_k$
$r_{k+1} = r_k - \alpha_k^{(k)} A p_k$
$\beta_{k-l}^{(k)} = -\frac{(Q_1^{-1} A Q_2^{-1} Q_1^{-1} r_{k+1}, Q_1^{-1} A p_{k-l})_0}{(Q_1^{-1} A p_{k-l}, Q_1^{-1} A p_{k-l})_0}, \quad l = 0, ..., s$
$p_{k+1} = Q_2^{-1} Q_1^{-1} r_{k+1} + \sum_{j=0}^{s} \beta_{k-j}^{(k)} p_{k-j}$
`Konec cyklu pro` k.
Konec Algoritmu.

Metodu danou tímto algoritmem nazveme „**zobecněná metoda sdružených gradientů ve smyslu nejmenších čtverců (GCG-LS(s)) s předpodmíněním**“, která je pro $M_0 = M_1$, M_0-normální matici A a pro $s \geq n - 1$ identická s plnou verzí této metody, tj. pro $t_k = k$ v algoritmu 2.2, pokud je „kontrolní člen“ $\alpha_{k-t_k}^{(k)} p_{k-t_k}$ nulový.

Kapitola 3

Metoda GMRES

V této kapitole předvedeme iterační metodu pro řešení lineárních systémů

$$Ax = f, \tag{3.1}$$

která v každém kroku minimalizuje normu residua přes nějaký Krylovův podprostor $\mathcal{K}_i$. Zajímavé je to, že touto metodou můžeme dostat řešení, i když má matice A indefinitní symetrickou část M.

3.1 Úvod

Metoda GMRES využívá Arnoldiho proces generování ortonormálních vektorů. Stejně jako např. u metod GCR nebo Orthodir si vezmeme Krylovův podprostor $\mathcal{K}_i(r_0, A)$ a pomocí Arnoldiho algoritmu najdeme ortonormální bázi prostoru $\mathcal{K}_i$. Tento proces vytváří matici koeficientů H_i, která je v horním Hessenbergově tvaru. Saad a Schultz (viz [Saad 2]) položili novou aproximaci x_i jako lineární kombinaci vygenerované ortonormální báze plus počáteční přiblížení x_0 a ukázali, že minimalizace normy residua závisí na matici koeficientů H_i. Použitím Givensových matic elementárních rotací se matice H_i převede na trojúhelníkovou matici, ze které již snadným způsobem získáme takovou lineární kombinaci Arnoldiho ortonormálních vektorů, že norma residua příslušné aproximace x_i je minimalizována přes Krylovův prostor $\mathcal{K}_i$.

Právě popsaná metoda GMRES (*Generalized Minimal Residual method*, tj. zobecněná metoda minimálních residuí, poznamenáváme, že nejde o zobecnění Algoritmu 1.6, jedná se pouze o čistou podobnost názvů) počítá řešení i v případě indefinitní symetrické části M a je teoreticky ekvivalentní s metodami GCR a Orthodir, ale na rozdíl od nich se nemusí zhroutit právě pro systémy, ve kterých má matice A indefinitní symetrickou část. Metoda Orthodir se sice také nemusí zhroutit, ale je numericky méně stabilní než např. metoda GCR, u které je velké nebezpečí zhroucení.

Příklad: Mějme systém dvou lineárních rovnic pro dvě neznámé $Ax = f$, kde

$$A = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad f = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

a položme $x_0 = (1, 1)^T$. Přesné řešení je $x^\star = (0, 0)^T$. Zkusme ho najít metodou GCR. Vezměme si proto Algoritmus 1.2 a počítejme:

- 1.krok: $i = 0$:

$$x_0 = (1,1)^T, \quad r_0 = f - Ax_0 = (-1,1)^T, \quad p_0 = r_0 = (-1,1)^T$$

$$\alpha_0 = \frac{(r_0, Ap_0)}{(Ap_0, Ap_0)} = 0, \quad x_1 = x_0 = (1,1)^T, \quad r_1 = r_0 = (-1,1)^T$$

$$\beta_0^{(0)} = -\frac{(Ar_1, Ap_0)}{(Ap_0, Ap_0)} = -1, \quad p_1 = r_1 + \beta_0^{(0)} p_0 = (0,0)^T$$

- 2.krok: $i = 1$: Při výpočtu $\alpha_1 = \frac{(r_1, Ap_1)}{(Ap_1, Ap_1)}$ dochází k dělení nulou a tudíž ke zhroucení algoritmu. Symetrická část $M = \frac{(A+A^T)}{2}$ matice A je nulová matice, není tedy positivně definitní.

Na tomto příkladu systému s indefinitní symetrickou částí vidíme, že se metoda GCR zhroutila. Jak uvidíme později, metoda GMRES však najde přesné řešení.

3.2 Algoritmus

Nejprve vytvoříme ortonormální bázi $(v_1, ..., v_i)$ v Krylovově podprostoru $\mathcal{K}_i(A, r_0) = sp(r_0, Ar_0, ..., A^{i-1}r_0)$. Aproximace x_i se pak volí tak, aby platilo

$$x_i = \arg\min_{x \in \mathcal{K}_i} \| f - Ax \|, \tag{3.2}$$

tj. x_i bude lineární kombinací vektorů $v_1, ..., v_i$. Pro výpočet těchto vektorů se používá Arnoldiho algoritmus. V části 1.2.3 jsme ukázali, jak tento algoritmus vypadá.

Algoritmus 3.1
Zvolíme v_1, kde $\| v_1 \| = 1$.
Pro $i = 1, 2, ...$ provedeme
$h_{j,i} = (Av_i, v_j), \quad j = 1, 2, ..., i$
$w = Av_i - \sum_{j=1}^{i} h_{j,i} v_j$
$h_{i+1,i} = \| w \|$
$v_{i+1} = \frac{w}{h_{i+1,i}}$
`Konec cyklu pro` i.
Konec Algoritmu.

Tento proces, který generuje ortonormální vektory, se nazývá „**Arnoldiho algoritmus**“. Maticový zápis Arnoldiho procesu vypadá takto:

$$AV_i = V_{i+1} \bar{H}_i, \tag{3.3}$$

kde

$$\bar{H}_i = \begin{pmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,i} \\ h_{2,1} & h_{2,2} & \dots & h_{2,i} \\ & h_{3,2} & \dots & h_{3,i} \\ & & \ddots & \vdots \\ & & & h_{i+1,i} \end{pmatrix} \quad \text{a} \quad V_i = (v_1, ..., v_i).$$

Důkaz je proveden v části 1.2.3.

Aproximaci x_i položíme jako lineární kombinaci vektorů $v_1, .., v_i$, tedy

$$x_i = x_0 + V_i y_i,$$

kde $y_i \in \mathbb{R}^i$ a položíme-li

$$v_1 = \frac{r_0}{\| r_0 \|}, \quad \xi := \| r_0 \|, \quad e_1 = (1, 0, ..., 0)^T,$$

pak můžeme upravit minimalizační podmínku (3.2) takto:

$$\begin{aligned} \| f - Ax_i \| &= \| f - A(x_0 + V_i y_i) \| = \| r_0 - AV_i y_i \| = \\ &= \| (\| r_0 \| \cdot v_1) - V_{i+1} \bar{H}_i y_i \| = \| \xi V_{i+1} e_1 - V_{i+1} \bar{H}_i y_i \| = \\ &= \| V_{i+1}(\xi e_1 - \bar{H}_i y_i) \|, \end{aligned}$$

neboli zkráceně

$$\| f - Ax_i \| = \| V_{i+1}(\xi e_1 - \bar{H}_i y_i) \| = \| \xi e_1 - \bar{H}_i y_i \|, \tag{3.4}$$

protože obecně platí

$$\| Vu \|^2 = (Vu, Vu) = (V^T Vu, u) = (u, u) = \| u \|^2 \quad \Rightarrow \quad \| Vu \| = \| u \|$$

pro ortonormální matici V a libovolný vektor u. Vidíme, že aproximaci x_i metody GMRES, která splňuje (3.2), lze spočítat ze soustavy

$$x_i = x_0 + V_i y_i, \tag{3.5}$$

kde pro y_i platí

$$y_i = \arg \min_{y \in \mathbb{R}^i} \| \xi e_1 - \bar{H}_i y \| . \tag{3.6}$$

Algoritmus metody GMRES je tedy sestaven z Arnoldiho metody, tj. z algoritmu 3.1 a vztahů (3.5) a (3.6).

Nyní použijeme předpodmínění. Řešíme lineární systém

$$\tilde{A}\tilde{x} = \tilde{f}, \quad \text{kde} \quad \tilde{A} = Q_1^{-1} A Q_2^{-1}, \quad \tilde{x} = Q_2 x, \quad \tilde{f} = Q_1^{-1} f.$$

Z původního algoritmu metody GMRES lze stejně jako v kapitole 1. u metody GCR odvodit algoritmus s předpodmíněním, kde definujeme $\eta = \tilde{\xi} = \| Q_1^{-1} r_0 \|$.

Položíme-li $Q_1 = Q_2 = I$, dostaneme algoritmus bez předpodmínění, položíme-li $Q_1 = I$, dostaneme algoritmus s pravým předpodmíněním a konečně pro $Q_2 = I$ máme algoritmus s levým předpodmíněním.

Algoritmus 3.2

Zvolíme x_0.
Spočteme $r_0 = f - Ax_0$.
Položíme $v_1 = \frac{Q_1^{-1} r_0}{\| Q_1^{-1} r_0 \|}$.
Pro $i = 1, 2, ...$ provedeme
$h_{j,i} = (Q_1^{-1} A Q_2^{-1} v_i, v_j), \quad j = 1, 2, ..., i$

$w = Q_1^{-1} A Q_2^{-1} v_i - \sum_{j=1}^{i} h_{j,i} v_j$
$h_{i+1,i} = \| w \|$
$v_{i+1} = \frac{w}{h_{i+1,i}}$
$x_i = x_0 + Q_2^{-1} V_i y_i$, kde
$y_i = \arg_{y \in \mathbb{R}^i} \min \| \eta e_1 - \bar{H}_i y \|$
$r_i = f - A x_i$
`Konec cyklu pro` i.
Konec Algoritmu.

Metoda vycházející z tohoto algoritmu se nazývá „**zobecněná metoda minimálních residuí - GMRES** *(Generalized Minimal Residual method)* s předpodmíněním“.

Aproximace x_i nemusíme počítat v každém kroku, ale například po každých l krocích, kde $l > 0$. To znamená, že spočítáme l vektorů $v_i,\ i = 1, ..., l$ a teprve potom příslušnou aproximaci x_i a residuum r_i. Bude-li norma tohoto residua $\| r_i \| < \varepsilon$, pak můžeme proces ukončit, v opačném případě pokračujeme ve výpočtu, tj. opět napočítáme l vektorů $v_i,\ i = l+1, ..., 2l$ a poté zkoumáme residuum.

Stejně jako u jiných metod, také zde můžeme algoritmus restartovat po každých k krocích, kde k je nějaký pevný celočíselný parametr. U nerestartované verze s rostoucím i roste počet vektorů potřebných k uložení do paměti. Proto lze použít restartovanou verzi, která tuto potíž odstraňuje.

Algoritmus 3.3
Zvolíme x_0.
Spočteme $r_0 = f - A x_0$.
Položíme $v_1 = \frac{Q_1^{-1} r_0}{\|Q_1^{-1} r_0\|}$.
100: Pro $i = 1, 2, ..., k$ provedeme
$h_{j,i} = (Q_1^{-1} A Q_2^{-1} v_i, Q_2 v_j), \quad j = 1, 2, ..., i$
$w = Q_1^{-1} A Q_2^{-1} v_i - \sum_{j=1}^{i} h_{j,i} v_j$
$h_{i+1,i} = \| w \|$
$v_{i+1} = \frac{w}{h_{i+1,i}}$
$x_i = x_0 + V_i y_i$, kde
$y_i = \arg_{y \in \mathbb{R}^i} \min \| \eta e_1 - \bar{H}_i y \|$.
$r_i = f - A x_i$
`Konec cyklu pro` i.
Je-li $i = k$, pak
$x_0 := x_k$
$v_1 := \frac{Q_1^{-1} r_i}{\|Q_1^{-1} r_i\|}$
$i = 0$
Návrat na 100.
Konec Algoritmu.

Metodu vycházející z tohoto restartovaného algoritmu nazýveme „**GMRES(k)**“.

Opět můžeme počítat residuum pouze po každých l krocích, stejně jako u nerestartované verze.

Nyní se budeme zabývat řešením problému (3.6). Abychom mohli takový vektor y_i určit, převedeme matici $\bar{H}_i$ na trojúhelníkovou matici pomocí matic rotací. Buď tedy

$$\bar{H}_i = \begin{pmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,i} \\ h_{2,1} & h_{2,2} & \dots & h_{2,i} \\ & h_{3,2} & \dots & h_{3,i} \\ & & \ddots & \vdots \\ & & & h_{i+1,i} \end{pmatrix}$$

Abychom dostali horní trojúhelníkovou matici, budeme postupně anulovat poddiagonálu matice $\bar{H}_i$. Začneme anulací prvku na pozici (2,1). Nechť

$$\bar{P}_1 = \begin{pmatrix} c_1 & s_1 & & & \\ -s_1 & c_1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix}$$

je první matice rotace, kde obecně

$$c_i = \cos(\theta_i), \quad s_i = \sin(\theta_i) \quad \text{a} \quad \theta_i \quad \text{je úhel rotace.}$$

Násobme

$$\begin{aligned} \bar{P}_1 \cdot \bar{H}_i &= \begin{pmatrix} c_1 & s_1 & & & \\ -s_1 & c_1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix} \cdot \begin{pmatrix} h_{1,1} & h_{1,2} & \dots & h_{1,i} \\ h_{2,1} & h_{2,2} & \dots & h_{2,i} \\ & h_{3,2} & \dots & h_{3,i} \\ & & \ddots & \vdots \\ & & & h_{i+1,i} \end{pmatrix} = \\ &= \begin{pmatrix} c_1 h_{1,1} + s_1 h_{2,1} & c_1 h_{1,2} + s_1 h_{2,2} & \dots \\ -s_1 h_{1,1} + c_1 h_{2,1} & -s_1 h_{1,2} + c_1 h_{2,2} & \dots \\ & h_{3,2} & \dots \\ & & \ddots \end{pmatrix} \end{aligned}$$

Teď chceme, aby $-s_1 h_{1,1} + c_1 h_{2,1} = 0$, a protože rovněž chceme, aby matice $\bar{P}_i$ byla ortonormální, bude platit $s_1^2 + c_1^2 = 1$. Získáváme soustavu dvou rovnic o dvou neznámých s_1 a c_1, kterou vyřešíme a dostaneme řešení:

$$c_1 = \frac{h_{1,1}}{\sqrt{h_{1,1}^2 + h_{2,1}^2}}, \quad s_1 = \frac{h_{2,1}}{\sqrt{h_{1,1}^2 + h_{2,1}^2}}.$$

Dosadíme do matice $\bar{P}_i$ a získáme obecně takovouto matici:

$$\bar{P}_1 \cdot \bar{H}_i = \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & \dots \\ 0 & \hat{r}_{2,2} & \hat{r}_{2,3} & \dots \\ & h_{3,2} & h_{3,3} & \dots \\ & & h_{4,3} & \dots \\ & & & \ddots \end{pmatrix}$$

kde

$$r_{1,1} = c_1 h_{1,1} + s_1 h_{2,1}, \quad r_{1,2} = c_1 h_{1,2} + s_1 h_{2,2}, \quad r_{1,3} = \ldots, \quad \text{atd.};$$

$$\hat{r}_{2,2} = -s_1 h_{1,2} + c_1 h_{2,2}, \quad \hat{r}_{2,3} = \ldots, \quad \text{atd.}$$

Dále budeme anulovat prvek na pozici (3,2). Nechť

$$\bar{P}_2 = \begin{pmatrix} 1 & & & & & \\ & c_1 & s_1 & & & \\ & -s_1 & c_1 & & & \\ & & & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \end{pmatrix}$$

je další, druhá matice rotace. Pak

$$\bar{P}_2 \cdot (\bar{P}_1 \cdot \bar{H}_i) = \begin{pmatrix} 1 & & & & & \\ & c_1 & s_1 & & & \\ & -s_1 & c_1 & & & \\ & & & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \end{pmatrix} \cdot \begin{pmatrix} r_{1,1} & r_{1,2} & \ldots & \ldots & \ldots \\ 0 & \hat{r}_{2,2} & \ldots & \ldots & \ldots \\ & h_{3,2} & \ldots & \ldots & \ldots \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \ddots \end{pmatrix} =$$

$$= \begin{pmatrix} r_{1,1} & r_{1,2} & \ldots \\ 0 & c_2\hat{r}_{2,2} + s_2 h_{3,2} & \ldots \\ & -s_2\hat{r}_{2,2} + c_2 h_{3,2} & \ldots \\ & & \ddots \end{pmatrix}$$

Vyřešíme soustavu dvou rovnic o dvou neznámých $-s_2\hat{r}_{2,2} + c_2 h_{3,2} = 0$ a $s_2^2 + c_2^2 = 1$ a dostaneme řešení

$$c_2 = \frac{\hat{r}_{2,2}}{\sqrt{\hat{r}_{2,2}^2 + h_{3,2}^2}}, \quad s_2 = \frac{h_{3,2}}{\sqrt{\hat{r}_{2,2}^2 + h_{3,2}^2}}.$$

Dosadíme a získáme takovouto matici:

$$\bar{P}_2 \cdot (\bar{P}_1 \cdot \bar{H}_i) = \begin{pmatrix} r_{1,1} & r_{1,2} & r_{1,3} & \ldots \\ 0 & r_{2,2} & r_{2,2} & \ldots \\ & 0 & \hat{r}_{3,3} & \ldots \\ & & h_{4,3} & \ldots \\ & & & \ddots \end{pmatrix}$$

kde první řádek zůstává beze změny a změní se jen druhý řádek:

$$r_{2,2} = c_2\hat{r}_{2,2} + s_2 h_{3,2}, \quad r_{2,3} = \ldots, \quad \text{atd.}$$

Takto pokračujeme dále až do indexu i. Označíme

$$P_i = \bar{P}_i \cdot \bar{P}_{i-1} \cdot \ldots \cdot \bar{P}_2 \cdot \bar{P}_1$$

jako součin Givensových matic elementárních rotací. Pak

$$P_i \cdot (\bar{H}_i y_i) = \begin{pmatrix} r_{1,1} & r_{1,2} & \dots & r_{1,i} \\ 0 & r_{2,2} & \dots & r_{2,i} \\ \vdots & & \ddots & \vdots \\ \vdots & & & r_{i,i} \\ 0 & \dots & \dots & 0 \end{pmatrix} \cdot y_i = \begin{pmatrix} R_i \\ 0 \end{pmatrix} \cdot y_i, \tag{3.7}$$

kde R_i je zmíněná horní trojúhelníková matice s prvky $r_{k,l}$.
Pravou stranu výrazu (3.6) musíme rovněž vynásobit maticí P_i a dostaneme

$$\begin{aligned} P_i \cdot (\xi e_1) &= \xi \cdot (\bar{P}_i \cdot \ldots \cdot \bar{P}_1) \cdot e_1 = \\ &= \xi \cdot \bar{P}_i \cdot \ldots \cdot \bar{P}_2 \cdot \begin{pmatrix} c_1 & s_1 & & & \\ -s_1 & c_1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} = \\ &= \xi \cdot \bar{P}_i \cdot \ldots \cdot \bar{P}_2 \cdot \begin{pmatrix} c_1 \\ -s_1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \\ &= \xi \cdot \bar{P}_i \cdot \ldots \cdot \bar{P}_3 \cdot \begin{pmatrix} 1 & & & & \\ & c_2 & s_2 & & \\ & -s_2 & c_2 & & \\ & & & 1 & \\ & & & & \ddots \\ & & & & & 1 \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ -s_1 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} = \\ &= \xi \cdot \bar{P}_i \cdot \ldots \cdot \bar{P}_3 \cdot \begin{pmatrix} c_1 \\ -s_1 c_2 \\ s_1 s_2 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \ldots = \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_i \\ \bar{g}_{i+1} \end{pmatrix} = \begin{pmatrix} g^{(i)} \\ \bar{g}_{i+1} \end{pmatrix}, \end{aligned}$$

tj.

$$P_i \cdot (\xi e_1) = \begin{pmatrix} g^{(i)} \\ \bar{g}_{i+1} \end{pmatrix}, \tag{3.8}$$

kde $g^{(i)} \in \mathbb{R}^i$ je vektor o složkách $(g_1, ..., g_i)$, které získáme po celkovém vynásobení. Tedy minimalizační podmínka (3.6) vypadá takto:

$$\begin{aligned} y_i &= \arg\min_{y \in \mathbb{R}^i} \| \xi e_1 - \bar{H}_i y \| = \arg\min_{y \in \mathbb{R}^i} \| P_i(\xi e_1 - \bar{H}_i y) \| = \\ &= \arg\min_{y \in \mathbb{R}^i} \| \begin{pmatrix} g^{(i)} \\ \bar{g}_{i+1} \end{pmatrix} - \begin{pmatrix} R_i \\ 0 \end{pmatrix} \cdot y \|, \end{aligned}$$

tedy zkráceně

$$y_i = \arg\min_{y \in \mathbb{R}^i} \| \begin{pmatrix} g^{(i)} \\ \bar{g}_{i+1} \end{pmatrix} - \begin{pmatrix} R_i \\ 0 \end{pmatrix} \cdot y \| . \tag{3.9}$$

Matice P_i je ortonormální, lze ji tedy v normě přidat. Protože R_i je trojúhelníková regulární matice, platí

$$g^{(i)} - R_i y_i = 0 \tag{3.10}$$

a použijeme-li vztah (3.4) pro vyjádření normy residua, dostaneme

$$\| r_i \| = \| f - Ax_i \| = \| \xi e_1 - \bar{H}_i y_i \| = \| \begin{pmatrix} g^{(i)} - R_i y_i \\ \bar{g}_{i+1} \end{pmatrix} \| = |\bar{g}_{i+1}|.$$

Čísla $|\bar{g}_{i+1}|$ jsou tedy normy residuí r_i pro $i = 1, 2, ...$ Jakmile metoda GMRES získá dostatečně malé residuum, tj. $|\bar{g}_{i+1}| \leq \varepsilon$, můžeme proces ukončit a položit

$$x_i = x_0 + V_i y_i, \quad \text{kde} \quad R_i y_i = g^{(i)}.$$

Podívejme se na předpodmínění. Ovlnkujeme-li výše uvedené vztahy, dostaneme algoritmus s předpodmíněním. V případě předpodmínění zprava nedochází k ničemu zvláštnímu, ale uvažujeme-li předpodmínění zleva, pak čísla $|\bar{g}_{i+1}|$ nejsou normy residuí $\| r_i \|$, nýbrž normy $\| Q_1^{-1} r_i \|$. Chceme-li tedy testovat, zda norma residua $\| r_i \| < \varepsilon$, musíme provést operaci navíc. Buď tuto normu odhadneme, tj. testujeme

$$\| Q_1 \| \cdot |\bar{g}_{i+1}| \leq \varepsilon,$$

protože

$$\| r_i \| = \| Q_1 Q_1^{-1} r_i \| \leq \| Q_1 \| \cdot \| Q_1^{-1} r_i \| \leq \| Q_1 \| \cdot |\bar{g}_{i+1}| \ \leq \ \varepsilon;$$

nebo v každém kroku vypočítáme aproximaci x_i a testujeme normu příslušného residua r_i podle vztahu $\| r_i \| = \| f - Ax_i \|$.

Můžeme tedy předvést jiný zápis algoritmu metody GMRES.

Algoritmus 3.4

Zvolíme x_0.
Spočteme $r_0 = f - Ax_0$.
Položíme $v_1 = \frac{Q_1^{-1} r_0}{\| Q_1^{-1} r_0 \|}$.
Položíme $\bar{g}_1 = \| Q_1^{-1} r_0 \|$.
Pro $i = 1, 2, ..., N$ provedeme
  $\zeta_{1,i} = (Q_1^{-1} A Q_2^{-1} v_i, v_1)$
  $h_{j,i} = (Q_1^{-1} A Q_2^{-1} v_i, v_j), \quad j = 1, 2, ..., i$
  $w = Q_1^{-1} A Q_2^{-1} v_i - \sum_{j=1}^{i} h_{j,i} v_j$
  $h_{i+1,i} = \| w \|$
  $v_{i+1} = \frac{w}{h_{i+1,i}}$
  Pro $j = 2, ..., i$ provedeme
    $r_{j-1,i} = c_{j-1} \zeta_{j-1,i} + s_{j-1} h_{j,i}$
    $\zeta_{j,i} = -s_{j-1} \zeta_{j-1,i} + c_{j-1} h_{j,i}$
  $r_{i,i} = \sqrt{\zeta_{i,i}^2 + h_{i+1,i}^2}$
  $c_i = \frac{\zeta_{i,i}}{r_{i,i}}$

$s_i = \frac{h_{i+1,i}}{r_{i,i}}$
$g_i = c_i \bar{g}_i$
$\bar{g}_{i+1} = -s_i \bar{g}_i$
Případ $Q_1 = I$:
Je-li $|\bar{g}_{i+1}| \leq \varepsilon$, pak
$R_i y_i = g^{(i)}$
$x_i = x_0 + Q_2^{-1} V_i y_i$ a STOP
Případ $Q_1 \neq I$:
Buď
Je-li $\| Q_1 \| \cdot |\bar{g}_{i+1}| \leq \varepsilon$, pak
$R_i y_i = g^{(i)}$
$x_i = x_0 + Q_2^{-1} V_i y_i$ a STOP
Nebo
$R_i y_i = g^{(i)}$
$x_i = x_0 + Q_2^{-1} V_i y_i$
$r_i = f - A x_i$
Je-li $\| r_i \| < \varepsilon$, pak STOP.
`Konec cyklu pro` i.
Konec Algoritmu.

Algoritmus vycházející z této metody se nazývá „**GMRES**“ určený maticí A a vektorem f.

Na závěr uděláme malou poznámku, jak lze alternativně spočítat nové koeficienty $h_{j,i}$, $j = 1, ..., i+1$ a vektory residuí r_i. Vyjdeme z Arnoldiho metody, tedy z Algoritmu 3.1.

- První i koeficienty $h_{j,i}$ spočítáme podle definice:

$$h_{j,i} = (Av_i, v_j), \quad j = 1, 2, ..., i;$$

respektive

$$h_{j,i} = (Q_1^{-1} A Q_2^{-1} v_i, v_j), \quad j = 1, 2, ..., i.$$

- Poslední koeficient $h_{i+1,i}$ spočítáme podle definice a ortogonality vektorů v_i:

$$\begin{aligned} h_{i+1,i}^2 &= \| Av_i - \sum_{j=1}^{i} h_{j,i} v_j \|^2 = (Av_i - \sum_{j=1}^{i} h_{j,i} v_j, Av_i - \sum_{j=1}^{i} h_{j,i} v_j) = \\ &= \| Av_i \|^2 - 2 \cdot (Av_i, \sum_{j=1}^{i} h_{j,i} v_j) + \| \sum_{j=1}^{i} h_{j,i} v_j \|^2 = \\ &= \| Av_i \|^2 - 2 \cdot \sum_{j=1}^{i} h_{j,i}^2 + \| \sum_{j=1}^{i} h_{j,i} v_j \|^2 = \| Av_i \|^2 - \sum_{j=1}^{i} h_{j,i}^2, \end{aligned}$$

neboť $\| v_j \| = 1 \ \ \forall j$;
respektive

$$h_{i+1,i}^2 = \| Q_1^{-1} A Q_2^{-1} v_i \|^2 - \sum_{j=1}^{i} h_{j,i}^2.$$

- Vektory residuí r_i spočteme z vektorů v_j, $j = 1, ..., i$ a z Av_i. Ze vztahu (3.4) lze residua vyjádřit jako

$$r_i = V_{i+1}(\xi e_1 - \bar{H}_i y_i).$$

Definujeme-li

$$t = (t_1, t_2, ..., t_{i+1})^T = \xi e_1 - \bar{H}_i y_i, \quad \text{kde} \quad \xi = \| \, r_0 \, \|,$$

pak

$$\begin{aligned} r_i &= (\sum_{j=1}^{i} t_j v_j) + t_{i+1} v_{i+1} = (\sum_{j=1}^{i} t_j v_j) + t_{i+1} \cdot \frac{Av_i - \sum_{j=1}^{i} h_{j,i} v_j}{h_{i+1,i}} = \\ &= \frac{t_{i+1}}{h_{i+1,i}} Av_i + \sum_{j=1}^{i} (t_j - t_{i+1} \cdot \frac{h_{j,i}}{h_{i+1,i}}) v_j. \end{aligned}$$

Obdobně definujeme-li

$$T = (T_1, T_2, ..., T_{i+1})^T = \eta e_1 - \bar{H}_i y_i, \quad \text{kde} \quad \eta = \| \, Q_1^{-1} r_0 \, \|,$$

pak

$$r_i = \frac{T_{i+1}}{h_{i+1,i}} Q_1^{-1} A Q_2^{-1} v_i + \sum_{j=1}^{i} (T_j - T_{i+1} \cdot \frac{h_{j,i}}{h_{i+1,i}}) v_j.$$

- Residua r_i lze také počítat rekurentně. Ze vztahů (3.5) a (3.3) plyne, že

$$r_i = r_0 - AV_i y_i = \xi v_1 - V_{i+1} \bar{H}_i y_i = V_{i+1}(\xi e_1 - \bar{H}_i y_i).$$

Dále ze vztahu (3.10) vidíme, že

$$y_i = R_i^{-1} g^{(i)}$$

a ze vztahu (3.7), že

$$\bar{H}_i = P_i^T \cdot \begin{pmatrix} R_i \\ 0 \end{pmatrix},$$

neboť matice P_i je ortonormální a tudíž $P_i^{-1} = P_i^T$. Odtud

$$r_i = V_{i+1}(\xi e_1 - P_i^T \cdot \begin{pmatrix} R_i \\ 0 \end{pmatrix} \cdot R_i^{-1} g^{(i)}) = V_{i+1}(\xi e_1 - P_i^T \cdot \begin{pmatrix} g^{(i)} \\ 0 \end{pmatrix}).$$

Vztah (3.8) upravíme takto:

$$\xi e_1 = P_i^T \cdot \begin{pmatrix} g^{(i)} \\ \bar{g}_{i+1} \end{pmatrix} = P_i^T \cdot \begin{pmatrix} g^{(i)} \\ 0 \end{pmatrix} + P_i^T \cdot e_{i+1} \cdot \bar{g}_{i+1}.$$

Z toho plyne, že

(3.11) $$r_i = V_{i+1} \cdot P_i^T \cdot e_{i+1} \cdot \bar{g}_{i+1}.$$

Pro matice P_i platí následující vztah:

$$P_i = \bar{P}_i \cdot \bar{P}_{i-1} \cdot \ldots \cdot \bar{P}_1 = \bar{P}_i \cdot \begin{pmatrix} P_{i-1} & o \\ o^T & 1 \end{pmatrix} = \begin{pmatrix} I_{i-1} & o & o \\ o^T & c_i & s_i \\ o^T & -s_i & c_i \end{pmatrix} \cdot \begin{pmatrix} P_{i-1} & o \\ o^T & 1 \end{pmatrix}$$

$\Longrightarrow$

$$P_i^T = \begin{pmatrix} P_{i-1}^T & o \\ o^T & 1 \end{pmatrix} \cdot \begin{pmatrix} I_{i-1} & o & o \\ o^T & c_i & -s_i \\ o^T & s_i & c_i \end{pmatrix}$$

Proto

$$\begin{aligned} r_i &= (V_i, v_{i+1}) \cdot \begin{pmatrix} P_{i-1}^T & o \\ o^T & 1 \end{pmatrix} \cdot \begin{pmatrix} I_{i-1} & o & o \\ o^T & c_i & -s_i \\ o^T & s_i & c_i \end{pmatrix} \cdot \begin{pmatrix} o \\ \bar{g}_{i+1} \end{pmatrix} = \\ &= -V_i P_{i-1}^T e_i s_i \bar{g}_{i+1} + v_{i+1} c_i \bar{g}_{i+1} = V_i P_{i-1}^T e_i s_i^2 \bar{g}_i + v_{i+1} c_i \bar{g}_{i+1}, \end{aligned}$$

protože $\bar{g}_{i+1} = -s_i \bar{g}_i$.
Podle rovnosti (3.11) nakonec dostaneme rovnost

$$r_i = r_{i-1} s_i^2 + v_{i+1} c_i \bar{g}_{i+1}, \quad \text{kde} \quad \bar{g}_1 = \| r_0 \|,$$

která v předpodmíněném případě získá tvar

$$r_i = r_{i-1} s_i^2 + Q_1 v_{i+1} c_i \bar{g}_{i+1}, \quad \text{kde} \quad \bar{g}_1 = \| Q_1^{-1} r_0 \|, \tag{3.12}$$

což je hledaná rekurence.

3.3 Teoretická analýza

U iteračních algoritmů vyvstává často otázka, zda se mohou zhroutit. Jak jsme mohli vidět, metoda GCR se může zhroutit pro problémy, ve kterých není matice A kladná reálná, tzn. že její symetrická část M není positivně definitní. V této části ukážeme, že metoda GMRES se nemůže zhroutit bez ohledu na positivnost matice M, protože předpoklad pozitivnosti nebudeme nikde potřebovat.

Nejprve předpokládejme, že lze zkonstruovat prvních i Arnoldiho vektorů. To nastane v případě, že $h_{j,j-1} \neq 0$, $j = 2, ..., i$, jak je vidět z algoritmu 3.1. Nechť tedy $h_{j,j-1} \neq 0$. To znamená, že z konstrukce matice R_{j-1} plyne, že diagonální prvek $r_{j-1,j-1}$ splňuje (viz Algoritmus 3.4):

$$r_{j-1,j-1} = \sqrt{\zeta_{j-1,j-1}^2 + h_{j,j-1}^2} > 0.$$

Diagonální prvky výsledné matice R_i se tedy neanulují, což znamená, že problém nejmenších čtverců (3.6), který je roven problému (3.9), lze vždy vyřešit. Algoritmus metody GMRES se tedy nemůže zhroutit, jakmile $h_{j,j-1} \neq 0$, $j = 2, ..., i$.

Nyní předpokládejme, že $h_{i+1,i} = 0$. Arnoldiho vektor v_{i+1} tedy nelze zkonstruovat. Víme, že platí vztah (3.3), který v případě $h_{i+1,i} = 0$ dostane tvar $AV_i = V_i H_i$. Vidíme, že matice A a H_i jsou podobné matice. Proto jestliže je matice A regulární, je regulární také matice H_i. Upravíme vztah (3.2). Víme, že $x_i = x_0 + V_i y_i$, kde y_i splňuje podmínku (3.6). Odtud

$$\begin{aligned} \| f - Ax_i \| &= \| f - A(x_0 + V_i y_i) \| = \| r_0 - AV_i y_i \| = \| \xi v_1 - V_i H_i y_i \| = \\ &= \| V_i(\xi e_1 - H_i y_i) \| = \| \xi e_1 - H_i y_i \| . \end{aligned}$$

Poněvadž je matice H_i čtvercová regulární, vypadá podmínka (3.6) jako $y_i = H_i^{-1} \xi e_1$. Platí tedy $\| r_i \| = 0$ a tedy x_i je přesné řešení. Odtud vidíme, že je-li $h_{i+1,i} = 0$, pak x_i je již přesné řešení.

Dále předpokládejme, že x_i je přesné řešení a předchozí x_j, $j = 1, ..., i-1$ nejsou. Pak norma residuí $\| r_j \| \neq 0$, $j = 1, ..., i-1$, ale norma residua $\| r_i \| = |\bar{g}_{i+1}| = 0$. Podle sestavení algoritmu však platí (viz Algoritmus 3.4) $\bar{g}_{i+1} = -s_i \bar{g}_i = 0$ a protože $|\bar{g}_i| = \| r_{i-1} \| \neq 0$, musí nutně platit $s_i = 0$. Ale

$$0 = s_i = \frac{h_{i+1,i}}{r_{i,i}} \quad \Rightarrow \quad h_{i+1,i} = 0$$

a z úvahy výše plyne, že algoritmus se zhroutí, neboť při výpočtu v_{i+1} dojde k dělení nulou.

Vraťme se ještě k Arnoldiho procesu (Algoritmus 3.1) a označme

$$\hat{v}_{i+1} = Av_i - \sum_{j=1}^{i} h_{j,i} v_j, \quad i = 1, 2, ...; \quad \hat{v}_1 = v_1.$$

Ukážeme, že podmínka $\hat{v}_j \neq 0$, $j = 1, ..., i$ a $\hat{v}_{i+1} = 0$ je ekvivalentní vlastnosti, že stupeň minimálního polynomu počátečního vektoru residua $r_0 = v_1 \cdot \| r_0 \|$ je roven i.
Předpokládejme nejprve, že tento stupeň je roven i. Pak tedy existuje polynom p_i stupně i takový, že

$$p_i(A)v_1 = 0$$

a i je nejnižší stupeň, pro který to platí. Proto jsou v Krylovově podprostoru

$$\mathcal{K}_{i+1}(A, r_0) = sp(r_0, Ar_0, ..., A^i r_0) = sp(v_1, v_2 ..., v_{i+1}) = sp(v_1, Av_1, ..., A^i v_1) = \mathcal{K}_{i+1}(A, v_1)$$

vektory $v_1, Av_1, ..., A^i v_1$ lineárně závislé a tudíž $\mathcal{K}_{i+1} = \mathcal{K}_i$. Vektor $\hat{v}_{i+1}$, který patří do množiny $\mathcal{K}_{i+1}$, neboť to je neznormovaný vektor v_{i+1}, a je ortogonální na množinu $\mathcal{K}_i$, protože tak se tvoří Arnoldiho vektory v_j, je proto nulový vektor.
Kromě toho předpokládejme, že $\hat{v}_j = 0$ pro nějaké $j = 1, ..., i$. Potom existuje polynom p_{j-1} stupně $j-1$ takový, že $\hat{v}_j = p_{j-1}(A)v_1 = 0$ a to je ve sporu s definicí minimálního polynomu p_i vektoru v_1, neboť i není nejmenší číslo takové, že platí $p_i(A)v_1 = 0$.
K důkazu opačné implikace předpokládejme, že

$$\hat{v}_j \neq 0, \quad j = 1, ..., i \quad \text{a} \quad \hat{v}_{i+1} = 0.$$

Pak existuje polynom p_i stupně i takový, že $p_i(A)v_1 = 0$ a je to polynom nejnižšího stupně, pro který toto platí. Opravdu, kdyby existovalo $j < i$ takové, že $p_j(A)v_1 = 0$, pak již víme (viz úvaha výše), že $\hat{v}_{j+1} = 0$ a to je spor, protože $j + 1 \leq i$.

Dále poznamenáváme, že $\hat{v}_{i+1} = 0$ implikuje, že (viz Algoritmus 3.1)

$$h_{i+1,i} = \| \hat{v}_{i+1} \| = 0$$

a naopak.

Tím jsme dokázali tuto větu:
Věta 3.1: Následující podmínky jsou ekvivalentní:

1. Aproximace x_i je přesné řešení.
2. Algoritmus se zhroutí v kroku i.
3. Číslo $h_{i+1,i} = 0$.

4. Vektor $\hat{v}_{i+1} = 0$.

5. Stupeň minimálního polynomu počátečního residua r_0 je roven i.

Protože pro N-dimensionální systém $Ax = f$ můžeme zkonstruovat maximálně N ortogonálních vektorů v_i, plyne odsud ještě tato věta:
Věta 3.2: Metoda GMRES dává přesné řešení po nejvýše N iteracích.

Podle věty 3.1 vidíme, že se restartovaný algoritmus GMRES(k) nezhroutí, avšak ne vždy konverguje. V případě, že matice A má symetrickou část M positivně definitní, pak konverguje vždy, stejně jako metoda GCR(k), což jsou ekvivalentní metody. Není-li však matice M positivně definitní, pak metoda GMRES(k) nemusí konvergovat, zatímco metoda GCR(k) se může dokonce zhroutit, jak jsme viděli výše.
Příklad: Uvažujme opět jako v úvodu problém $Ax = f$, kde

$$A = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad f = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad x_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

a vezměme si metodu GMRES(1). Algoritmus 3.3 dává toto:

- $r_0 = f - Ax_0 = (-1, 1)^T$, $v_1 = \frac{r_0}{\|r_0\|} = \frac{1}{\sqrt{2}}(-1, 1)^T$
- $h_{1,1} = (Av_1, v_1) = (\frac{1}{\sqrt{2}}(1, 1)^T, \frac{1}{\sqrt{2}}(-1, 1)^T) = 0, \quad h_{2,1} = \| w \| = \| Av_1 - h_{1,1}v_1 \| = 1$
- $y_1 = \arg_{y \in \mathbb{R}_1} \min \| (e_1 \| r_0 \|) - (h_{1,1}, h_{2,1})^T y \| = \arg_{y \in \mathbb{R}_1} \min \| (\sqrt{2}, 0)^T - (0, 1)^T y \| = \arg_{y \in \mathbb{R}_1} \min \sqrt{2 + y^2} = 0$
- $x_1 = x_0 + v_1 y_1 = x_0 = (1, 1)^T$ a restartujeme.

Vidíme, že GMRES(1) dává konstantní posloupnost aproximací a proto nemůže konvergovat. Uvažujeme-li však metodu GMRES($\geq$2), dostaneme přesné řešení ve druhé iteraci:

- $v_2 = \frac{w}{h_{2,1}} = Av_1 = \frac{1}{\sqrt{2}}(1, 1)^T$
- $h_{1,2} = (Av_2, v_1) = -1, \quad h_{2,2} = (Av_2, v_2) = 0, \quad h_{3,2} = \| w \| = \| Av_2 - h_{1,2}v_1 \| = 0$
- $y_2 = \arg_{y \in \mathbb{R}_2} \min \| (e_1 \| r_0 \|) - \begin{pmatrix} h_{1,1} & h_{1,2} \\ h_{2,1} & h_{2,2} \\ & h_{3,2} \end{pmatrix} \cdot y \| = \arg_{y \in \mathbb{R}_2} \min \| \begin{pmatrix} \sqrt{2} \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 & -1 \\ 1 & 0 \\ & 0 \end{pmatrix} \cdot \begin{pmatrix} y^{(1)} \\ y^{(2)} \end{pmatrix} \| = \arg_{y \in \mathbb{R}_2} \min \| (\sqrt{2} + y^{(2)}, -y^{(1)}, 0)^T \| = (0, -\sqrt{2})^T$
- $x_2 = x_0 + (v_1, v_2) \cdot y_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \frac{1}{\sqrt{2}} \cdot \begin{pmatrix} -1 & 1 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ -\sqrt{2} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$,

čímž získáváme přesné řešení. A protože je systém velikosti 2, dostali jsme ho skutečně po nejvýše dvou iteracích. Vidíme tedy rozdíl mezi metodami GMRES a GCR v případě jednoho konkrétního systému s indefinitní symetrickou částí M. Zatímco metoda GMRES našla přesné řešení, tak metoda GCR se zhroutila, jak jsme ukázali v úvodu této kapitoly.

Nyní předpokládejme, že P_k je prostor všech polynomů stupně nejvýše k a $\sigma(A)$ je spektrum matice A. Větu 1.4 pro metodu GCR lze přenést na metodu GMRES(k).

Věta 3.3: Nechť A je diagonalizovatelná tak, že $A = XDX^{-1}$, kde X je regulární a D je diagonální matice. Nechť dále symetrická část M matice A je positivně definitní a nechť

$$m_i \;=\; \min_{p_k \in P_k,\; p_k(0)=1} \; \max_{\lambda \in \sigma(A)} |p_k(\lambda)|. \tag{3.13}$$

Pak norma residua získaná v k-tém kroku metody GMRES splňuje:

1.
$$\| r_k \| \leq \; \kappa(X) \cdot m_i \cdot \| r_0 \|, \quad \text{kde} \quad \kappa(X) = \| X \| \cdot \| X^{-1} \| \,. \tag{3.14}$$

2.
$$\| r_k \| \leq \; \left[1 - \frac{\lambda_{\min}(M)^2}{\lambda_{\max}(A^T A)} \right] \cdot \| r_0 \| \,. \tag{3.15}$$

Pro lepší přehlednost budeme psát všechny vztahy v nepředpodmíněném tvaru, protože je umíme převést do tvarů s maticemi Q_1 a Q_2.

Věta 3.3 dokazuje konvergenci metody GMRES(k) pro každé k, pokud je M positivně definitní matice. Není-li však M positivně definitní, tj. matice A není kladná reálná, má tedy vlastní čísla vlevo od imaginární osy, pak upravíme vztah (3.13) a zavedeme horní odhad pro m_i.

Věta 3.4: Nechť má matice A právě l vlastních čísel $\lambda_1, \lambda_2, ...\lambda_l$, které mají reálnou část nekladnou. Nechť zbylá vlastní čísla jsou uzavřena v kruhu se středem C a poloměrem R, kde $C > R \geq 0$. Pak platí:

$$m_i \;\leq\; \left(\frac{R}{C}\right)^{k-l} \cdot \max_{j=l+1,..,N} \prod_{i=1}^{l} \frac{|\lambda_i - \lambda_j|}{|\lambda_i|} \;\leq\; \left(\frac{D}{d}\right) \cdot \left(\frac{R}{C}\right)^{k-l}, \tag{3.16}$$

kde

$$D \;= \max_{i=1,...,l; j=l+1,...,N} |\lambda_i - \lambda_j| \quad \text{a} \quad d \;= \min_{i=1,...,l} |\lambda_i|.$$

Důkaz: Uvažujme třídu polynomů definovanou jako $p(z) = r(z)q(z)$, kde

$$r(z) = \left(1 - \frac{z}{\lambda_1}\right) \cdot \left(1 - \frac{z}{\lambda_2}\right) \cdot \ldots \cdot \left(1 - \frac{z}{\lambda_l}\right)$$

a $q(z)$ je libovolný polynom stupně nejvýše $k - l$ takový, že $q(0) = 1$.
Protože platí $p(0) = 1$, $p(\lambda_i) = 0$, $i = 1, ..., l$ a $p(z)$ je polynom stupně nejvýše k, pak dostáváme ze vztahu (3.13)

$$m_i \;= \max_{j=l+1,...,N} |p(\lambda_j)| \;\leq \max_{j=l+1,...,N} |r(\lambda_j)| \;\cdot \max_{j=l+1,...,N} |q(\lambda_j)|$$

Dále vidíme, že

$$\max_{j=l+1,...,N} |r(\lambda_j)| \;= \max_{j=l+1,...,N} \prod_{i=1}^{l} \frac{|\lambda_i - \lambda_j|}{|\lambda_i|} \;\leq\; \left(\frac{D}{d}\right)^{l}$$

Kromě toho maximum $|q(z)|$, kde $z \in U = \{\lambda_j\}_{j=l+1,...,N}$ není větší, než maximum $|q(z)|$ pro z patřící do kruhu, který množinu U uzavírá. Vezmeme proto polynom

$$q(z) = \left(\frac{C - z}{C}\right)^{k-l},$$

který nabývá maxima v absolutní hodnotě pro z libovolně na hranici kruhu se středem C a poloměrem R, tedy např. pro $z = R + C$ a toto maximum je $(\frac{R}{C})^{k-l}$. Tedy

$$m_i \leq \left(\frac{D}{d}\right)^l \cdot \left(\frac{R}{C}\right)^{k-l},$$

což dává žádaný výsledek. □

Jsou-li všechna vlastní čísla matice A reálná, pak lze nerovnost (3.16) upravit takto:

$$m_i \;\leq\; \left(\frac{R}{C}\right)^{k-l} \cdot \max_{j=l+1,..,N} \prod_{i=1}^{l} \frac{|\lambda_i - \lambda_j|}{|\lambda_i|} \;=\; \left(\frac{R}{C}\right)^{k-l} \cdot \prod_{i=1}^{l} \frac{|\lambda_i - \lambda_N|}{|\lambda_i|},$$

kde λ_N je největší vlastní číslo matice A.

Na závěr vyslovíme důsledek předchozí věty, který nám dává správnou hodnotu k pro restartovanou metodu GMRES(k), aby konvergovala.

Lemma 3.1: Nechť jsou splněny předpoklady Vět 3.3 a 3.4. Pak restartovaná metoda GMRES(k) konverguje pro libovolný počáteční vektor x_0, jestliže

$$k \;>\; l \cdot \frac{\log\left(\frac{D \cdot C}{d \cdot R} \cdot \kappa(X)^{\frac{1}{l}}\right)}{\log\left(\frac{C}{R}\right)} \tag{3.17}$$

Důkaz: Vezmeme vztahy (3.14) a (3.16), dáme je dohromady a upravíme je.

$$\left(\frac{D}{d}\right)^l \cdot \left(\frac{R}{C}\right)^{k-l} \;\geq\; m_i \;\geq\; \frac{\| r_k \|}{\| r_0 \|} \cdot \frac{1}{\kappa(X)} \quad \Rightarrow \quad \frac{\| r_k \|}{\| r_0 \|} \;\leq\; \left(\frac{D}{d}\right)^l \cdot \left(\frac{R}{C}\right)^{k-l} \cdot \kappa(X)$$

Chceme, aby norma residuí klesala. Proto bude-li výraz vpravo menší než 1, pak bude menší než 1 i podíl norem residuí. Odtud spočteme k logaritmováním celého výrazu a úpravou logaritmů. Tedy

$$\begin{aligned}
\log\{\left(\frac{D}{d}\right)^l \cdot \left(\frac{R}{C}\right)^{k-l} \cdot \kappa(X)\} \;&<\; \log 1 = 0 \quad \Rightarrow \\
\Rightarrow \quad \log\{\left(\frac{DC}{dR}\right)^l \cdot \kappa(X)\} + \log\{\left(\frac{R}{C}\right)^k\} \;&<\; 0 \quad \Rightarrow \\
\Rightarrow \quad l \cdot \log\{\frac{DC}{dR} \cdot (\kappa(X))^{\frac{1}{l}}\} \;&<\; \log\{\left(\frac{C}{R}\right)^k\} \quad \Rightarrow \\
\Rightarrow \quad k \;&>\; l \cdot \frac{\log\left(\frac{D \cdot C}{d \cdot R} \cdot \kappa(X)^{\frac{1}{l}}\right)}{\log\left(\frac{C}{R}\right)},
\end{aligned}$$

neboť $C > R$ a tudíž je logaritmus $\log(\frac{C}{R}) > 0$. Tento výsledek dává odhad pro k, i když někdy může být nerozumný (číslo $\kappa(X)$ může být hodně velké). □

Kapitola 4

Numerické výsledky

V této kapitole vyzkoušíme výše uvedené metody na konkrétním příkladu. Vezmeme si tři různě velké matice, na kterých budeme metody testovat. Z dosažených výsledků uděláme závěr.

4.1 Úvod

Uvažujme parciální diferenciální rovnici

$$-(Au_x)_x - (Bu_y)_y + Cu_x + Du_y + (Eu)_x + (Fu)_y + Gu = H \quad \text{na} \quad \Omega \tag{4.1}$$

s Dirichletovou okrajovou podmínkou

$$u/_{\partial\Omega} = 0, \tag{4.2}$$

kde Ω je jednotkový čtverec. Za koeficienty A až G zvolíme konstanty a předpokládáme, že $A > 0,\ B > 0,\ G > 0$, tedy, že výše uvedená rovnice je eliptického typu. Funkci H zvolíme nulovou. Provedeme standardní pětibodovou konečnou diferenční aproximaci rovnice (4.1) a obdržíme obecně pětidiagonální nesymetrickou matici. Při volbě homogenních okrajových podmínek (4.2) a nulové pravé straně vyjde, že přesné řešení je nulové. Jako počáteční aproximaci jsme volili vektor $x_0 = (1, ..., 1)^T$ a proces jsme ukončili, jakmile norma residua nabyla hodnoty $\| r_i \| < 10^{-6}$. Postupně jsme na různě hustých sítích vygenerovali tři různě velké matice o dimenzích 900, 2500 a 4900 a pozorovali jsme chování různých metod s i bez předpodmínění (a to jak zprava, tak zleva). Pod pojmem menší soustavy budeme v této kapitole rozumnět soustavy dimenze 900, středně velká soustava bude mít dimenzi 2500 a větší soustava dimenzi 4900.

Algoritmy uvedené v kapitolách 1 až 3 lze programátorsky různě modifikovat. Uvažujme například metodu sdružených residuí. K výpočtu koeficientů $\beta_j^{(i)}$ potřebujeme vektory $\{Q_1^{-1}Ap_j\}_{j=0}^i$, takže máme nyní dvě možnosti:

- Buď uchovávat v paměti nejen vektory $\{p_j\}_{j=0}^i$, které potřebujeme k výpočtu nového směru p_{i+1}, ale také vektory $\{Q_1^{-1}Ap_j\}_{j=0}^i$, které potřebujeme k výpočtu koeficientů $\beta_j^{(i)}$. Tím sice ušetříme čas potřebný k výpočtu těchto vektorů v každé smyčce, ale potřebujeme o to více paměti pro uložení většího množství vektorů, konkrétně dvě dvourozměrná pole, která se s rostoucím i neustále zvětšují;

- Nebo uchovávat pouze vektory $\{p_j\}_{j=0}^{i}$, a vektory $\{Q_1^{-1}Ap_j\}_{j=0}^{i}$ počítat v každém kroku. V tomto případě bude výpočet pomalejší, ale zato ušetříme hodně paměti, neboť nám bude stačit jen jedno dvourozměrné pole.

Dále můžeme volit při výpočtu aproximace x_i:

- Buď počítat aproximace v každém kroku, to je vhodné třeba k získávání její normy, pokud se chceme podívat na křivku velikosti chyby a známe přesné řešení;

- Nebo spočítat přibližné řešení x_i až v momentě, jakmile bude norma residua menší než námi zadaná tolerance. V tomto případě však musíme uchovávat všechny koeficienty $\alpha_j^{(i)}$ a konečná aproximace se spočte jako lineární kombinace všech vektorů p_j plus počáteční přiblížení x_0. V tomto případě však nebudeme mít průběh grafu normy chyby, ale jen velikost chyby poslední aproximace.

Čas a paměť stojí proti sobě. Abychom ušetřili paměť a výpočty měli obecně pokud možno co nejrychlejší, tak se provádí usekávání algoritmu, čímž získáme tzv. algoritmus Orthomin(k) nebo restartování, tj. algoritmus GCR(k).

Při testování jsme dále sledovali rychlost výpočtu jednotlivých metod a nakonec jsme pozorovali, jak moc se liší spočtené řešení od přesného řešení, které je nula, tj. zajímala nás $\| x_i \|$. Průběh křivek těchto norem je podobný jako průběh křivek norem residuí, proto znázorníme pouze normy aproximací, které se u dané metody počítali jako poslední, tj. norma residua této poslední aproximace je již menší než zadané epsilon. Je to i z toho důvodu, že např. při použití metody Orthores se aproximace x_i nepočítají v každém kroku, ale pouze na závěr (viz Algoritmus 1.18) a naopak použijeme-li např. metodu Orthomin(k), pak aproximace x_i můžeme, ale i nemusíme počítat v každém kroku. Na závěr si ale na jednom grafu ukážeme, jak vypadá celý průběh normy chyby pro největší uvažovanou matici pro různé metody.

Nyní již přistoupíme k testování jednotlivých metod.

4.2 Testování metod

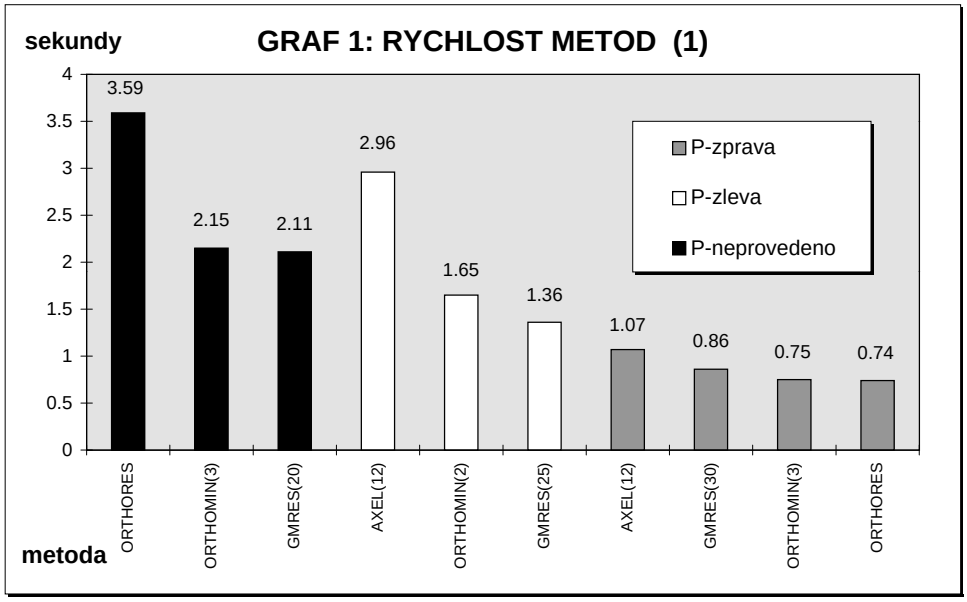
Testy jsme provedli na počítači s procesorem Pentium 100 MHz a operační pamětí 16 Mb. Ke zkompilování programu jsme použili Microsoft Fortran PowerStation 4.0 pod Windows 95.

Jak jsme již uvedli, budeme zkoumat tři různě velké matice, které jsou pětidiagonální a vzniknou diskretizací výše uvedené diferenciální rovnice.

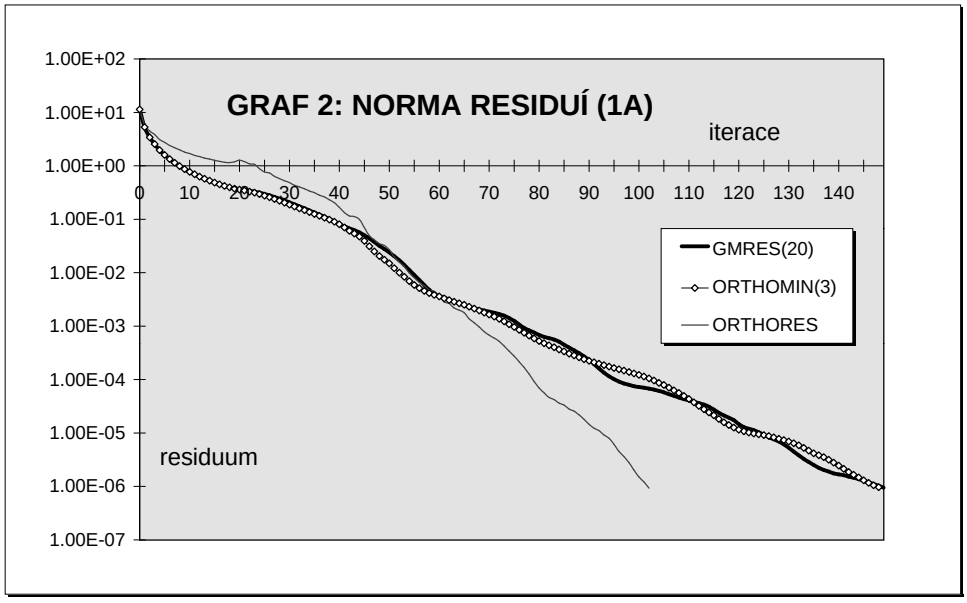
Nejprve se podíváme na první matici velikosti **900 krát 900** s koeficienty

$$A = 1.1, \quad B = 0.9, \quad C = D = 2, \quad E = F = G = 1.$$

Začneme rychlostí výpočtu.

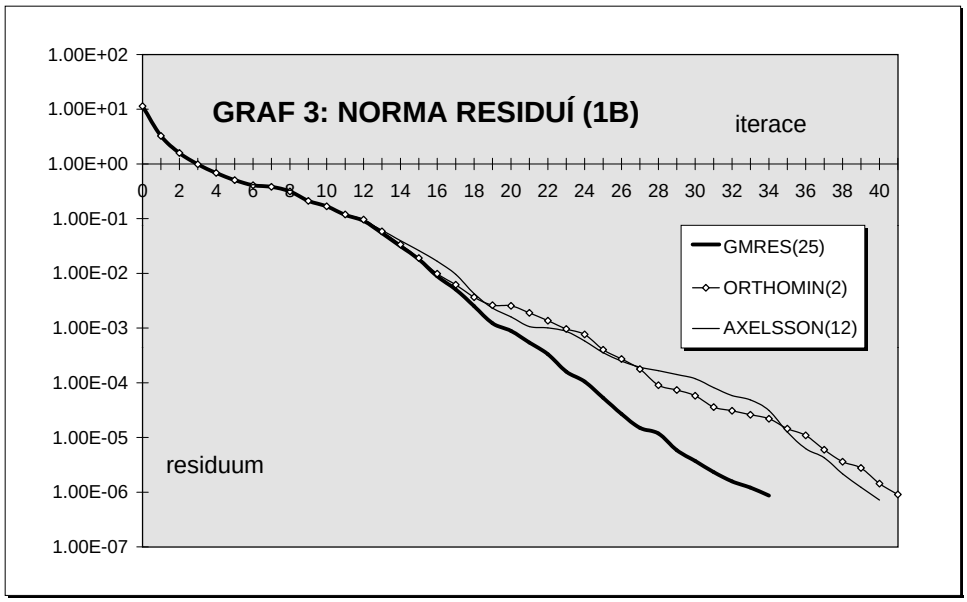


Vidíme, že různé metody počítají celkem rychle a to jak v případě nepředpodmíněného systému, tak i pro systémy předpodmíněné zleva a zprava. Vybrali jsme jen ty metody, které byly u této matice nejlepší. To znamená, že např. metody GCR(k) nebo MR počítali o trochu pomaleji, než metody uvedené na grafu 1. Podívejme se na residua nejprve v případě nepředpodmíněné soustavy.

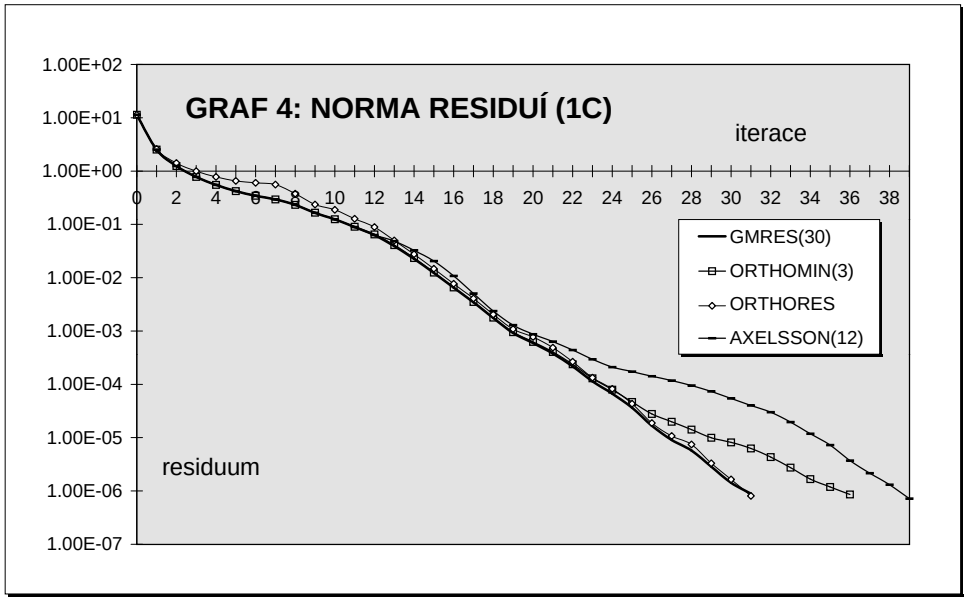


Na grafu 2 vidíme, že nejméně iterací k získání dostatečně malého residua potřebuje

metoda Orthores, zatímco metody GMRES(20) a Orthomin(3) dávají až do 40. iterace stejná residua. Podotýkáme, že u GMRES(k) jsme vybírali restart tak, aby výsledná metoda byla ze všech nejrychlejší. To nastalo pro $k = 20$. Obdobně useknutí u metody Orthomin(k) bylo nejlepší pro $k = 3$. Stejným způsobem jsme hledali hodnoty k i u dalších pokusů. Zkusme levé předpodmínění.

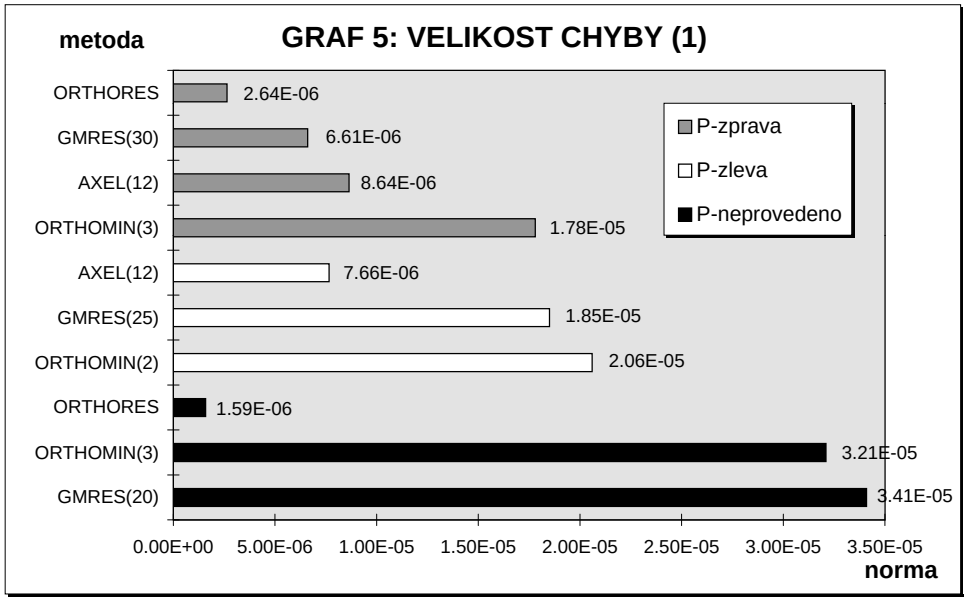


V tomto případě je metoda Orthores pomalejší, proto jsme ji nezařadili. U metody Orthomin(k) lze nyní volit $k = 2$, stačí tedy o jeden vektor méně. Naopak rychlejší než GMRES(20) je GMRES(25). Dále v grafu 3 vidíme, že k získání přesného řešení stačí daleko méně iterací než v prvním případě, ovšem čas již o tolik rychlejší není. Nakonec se podívejme na předpodmínění zprava.



Zajímavé je, že metody GMRES(30), Orthomin(3) a Axelsson(12) dávají až do 12. iterace stejnou normu residuí, zatímco metoda Orthores se od nich liší. Z časového srovnání je patrné, že pouze metoda Axelsson(12) je trochu pomalejší. Podotýkáme, že jakmile půjde řeč o metodě Axelsson(k) v této kapitole, bude to restartovaná verze a je to jen jiné označení pro restartovanou metodu GCG-LS.

Dá se říci, že pro menší matice jsou všechny metody spolehlivé, ne příliš pomalé a předpodmínění se ještě moc neprojevuje. Jedinou nespolehlivou metodou se ukázala být metoda Orthodir(k) pro jakékoli hodnoty k, která se vždy zhroutila. Když jsme však počáteční přiblížení x_0 zmenšili, tj. zvolili jsme x_0 náhodně, jehož norma byla řádově 10^{-3}, pak i tato metoda byla rychlá a spolehlivá. Ovšem pro ještě menší matice dimenze 400 se tato metoda ukázala být dobrou a našla řešení za krátkou dobu pro jakékoli x_0. Nyní si znázorníme normu chyby.

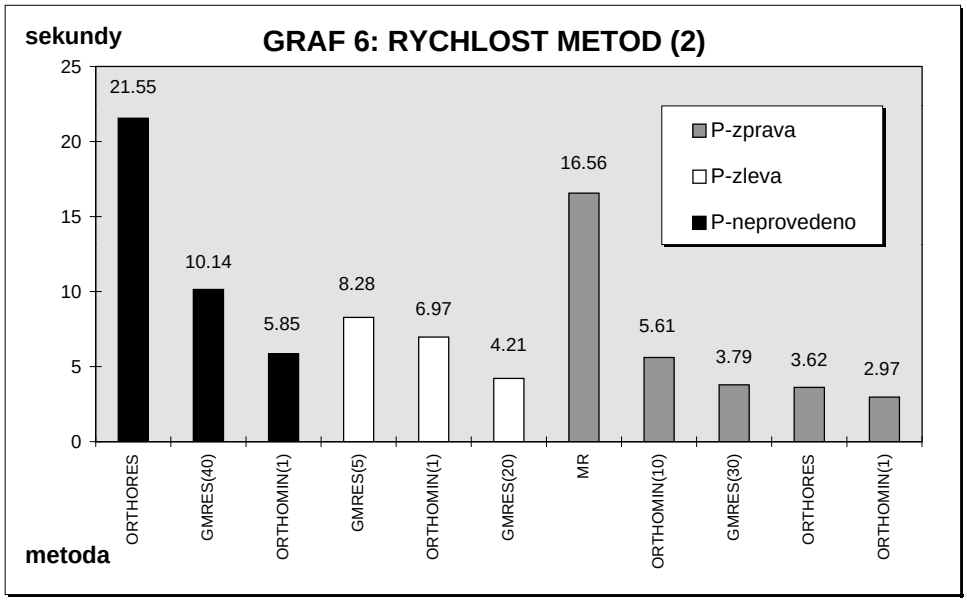


Na tomto grafu je vidět, že všechny uvedené metody dávají dobré řešení, jehož norma je malá a tudíž vektor přibližného řešení je blízko nulovému vektoru, který představuje přesné řešení. Připomínáme, že tyto normy jsou normy aproximací poslední iterace.

Nyní přejdeme k větší matici velikosti **2500 krát 2500** s koeficienty

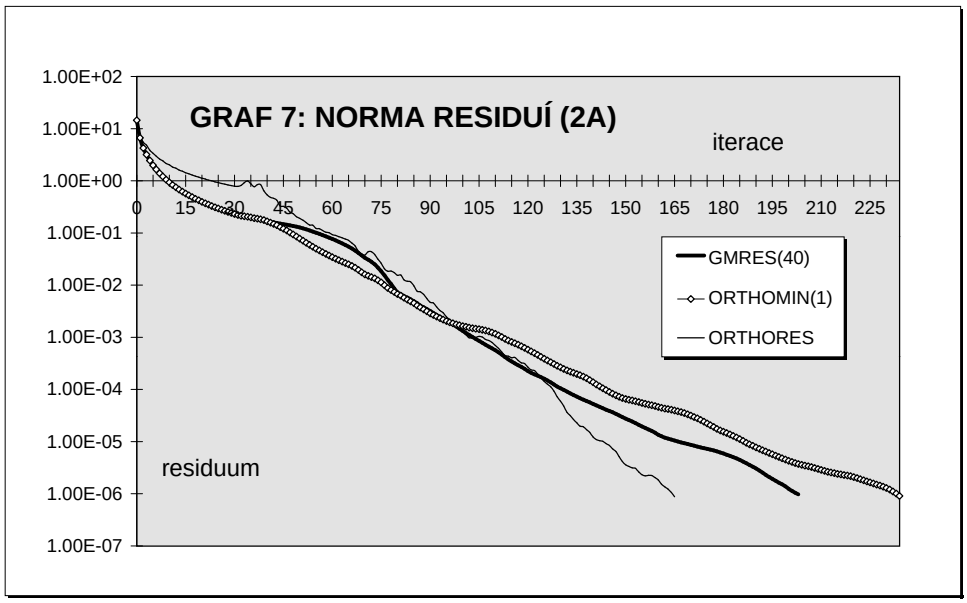
$$A = 1.1, \quad B = 0.9, \quad C = D = 1, \quad E = F = 0, \quad G = 1$$

a podíváme se, jak jsou vybrané metody rychlé.

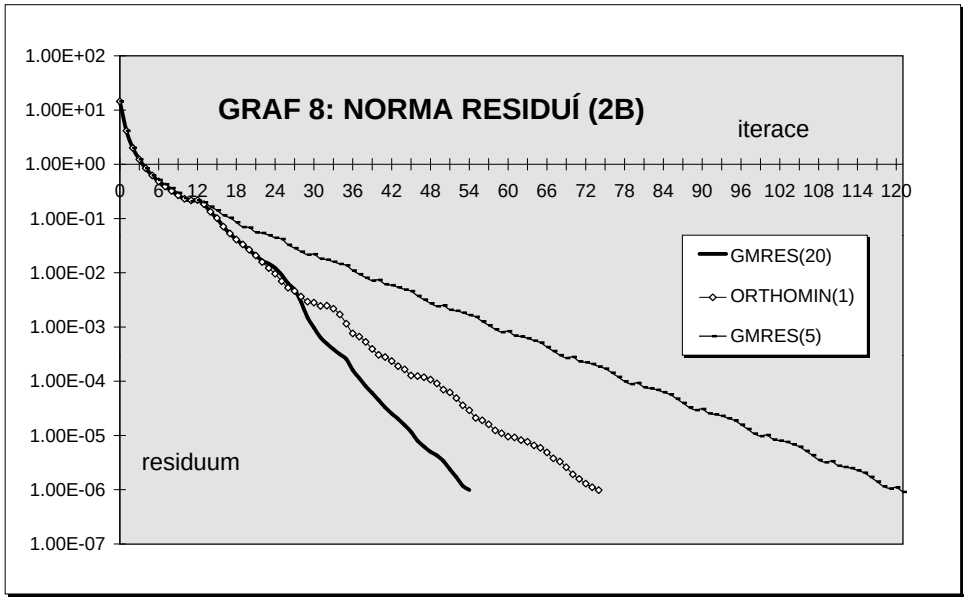


Zde již vidíme větší rozdíl než u první matice. Zvlášť předpodmínění zprava se ukazuje být velice rychlé. U předpodmínění zleva zpomaluje výpočet velký počet násobení vektorů maticí Q_1^{-1}. Těchto operací je daleko více, než v případě násobení maticí Q_2^{-1} u předpodmínění zprava. Proto je levé předpodmínění pomalejší než pravé.

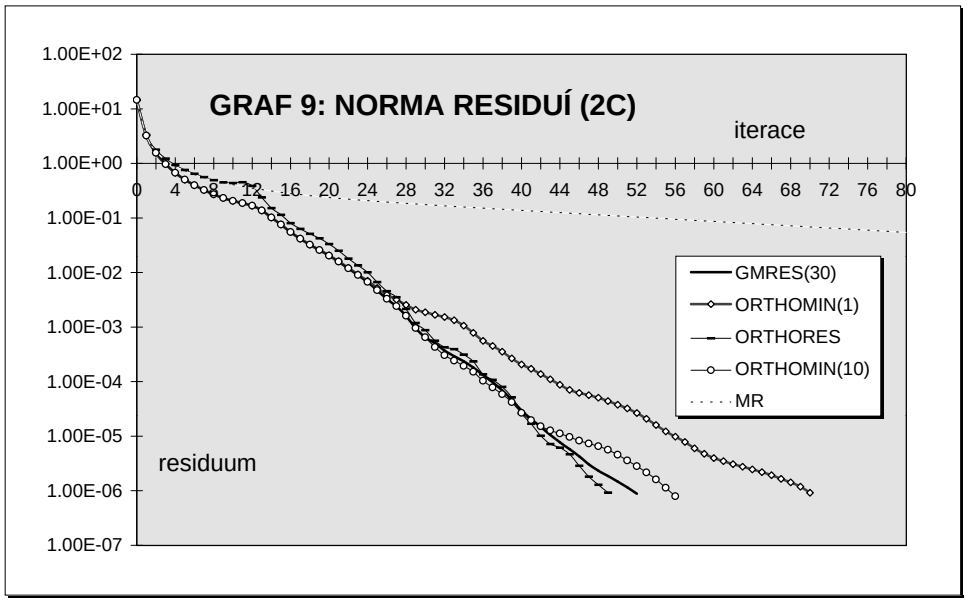
Všimněme si hlavně u levého předpodmínění na metodě GMRES(k), jak rychlost výpočtu závisí na restartu. Restartujeme-li příliš brzy, po pěti krocích, dostaneme přesné řešení za dvojnásobnou dobu, než restartujeme-li po více (v tomto případě po dvaceti) krocích. Stejný problém, ale přesně naopak vidíme u metody Orthomin(k) při použití předpodmínění zprava. Použijeme-li k výpočtu směru p_{i+1} pouze jeden předchozí vektor p_i, dostaneme dvakrát rychlejší proces oproti případu použití deseti vektorů $\{p_j\}_{j=i-9}^{i}$. Podívejme se na residua a zkusme nejprve systém bez předpodmínění.



Nejrychlejší metoda Orthomin(1) dá přesné řešení po nejvíce iteracích, zatímco u metody Orthores je tomu naopak. Ostatní metody jako např. Axelssonova metoda nebo restartovaná GCR jsou pomalejší a již se nehodí použít je na soustavu bez předpodmínění. Nyní zkusíme levé předpodmínění.



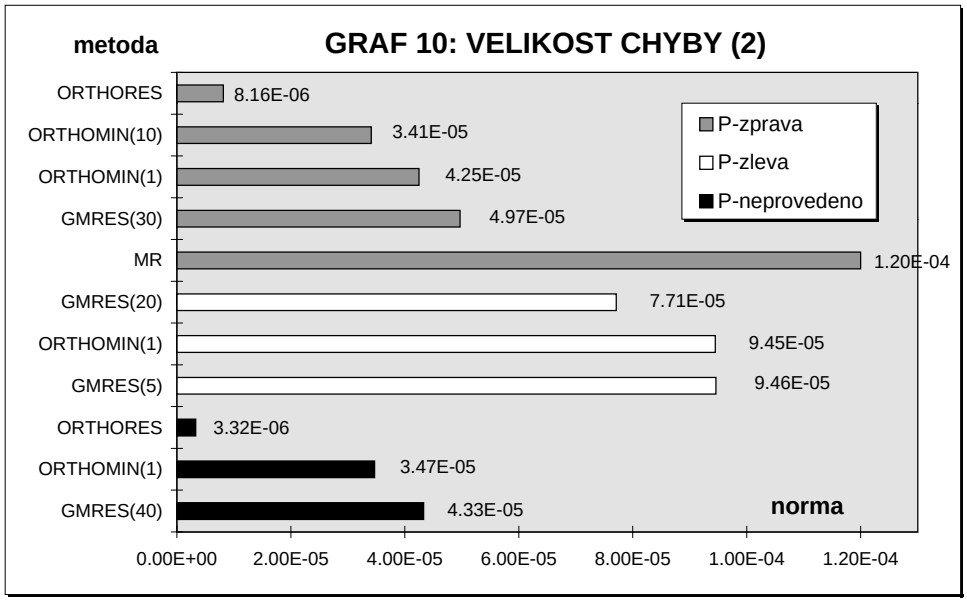
Zde rovněž vidíme závislost na volbě restartu u metody GMRES. Restartujeme-li příliš brzy, např. po pěti krocích, nejenže bude výpočet pomalejší, ale bude potřeba i více iterací. Zkusme předpodmínění zprava.



Metoda Orthomin(10) dává řešení po menším počtu iterací, než Orthomin(1), ale je pomalejší. Důvodem je to, že použijeme-li více vektorů p_j, než pouze jeden, dostaneme lepší směr p_{i+1} a tím aproximace x_{i+1} bude blíže přesnému řešení. Ovšem v každé smyčce se

pracuje s deseti vektory a tím je výpočet pomalejší z časového hlediska. Celkově můžeme říci, že křivky norem residuí jsou přibližně stejné.

Všimněme si jenom nejjednodušší metody, kterou jsme v této práci představili, a tou je metoda MR, Algoritmus 1.6, u které jsme uvažovali předpodmínění soustavy zprava. Tato metoda je velice pomalá, jak vidíme na grafu 9, je dokonce pomalejší než vybrané metody bez předpodmínění a potřebuje 553 iterací k tomu, abychom získali potřebně malé residuum. Tuto metodu jsme uvedli jen pro představu, jak taková nejjednodušší metoda vypadá v praxi. Metoda MR je vlastě metoda Orthomin(0) nebo GCR(0).
Podívejme se na velikost chyby.

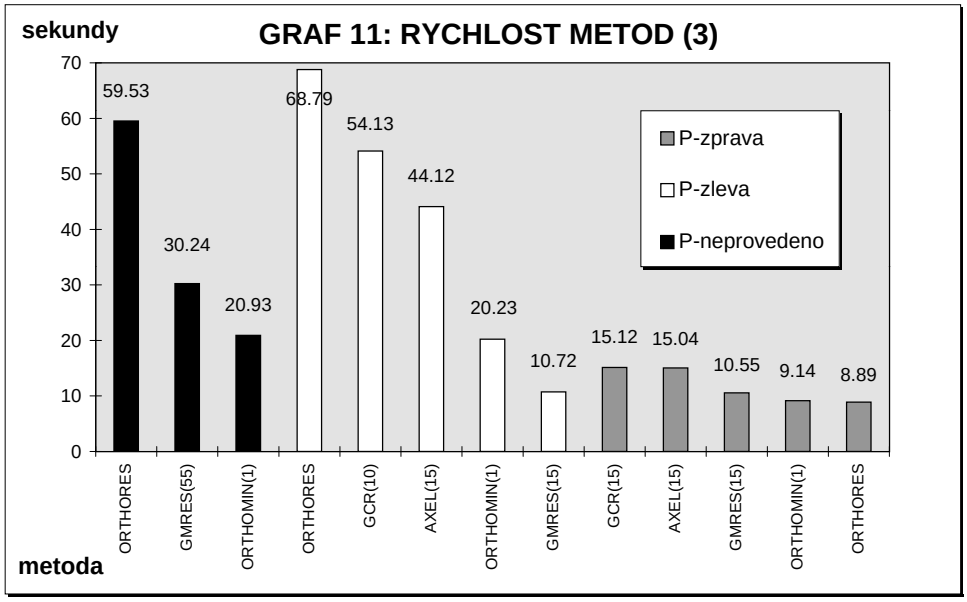


Zde již vidíme větší rozdíly oproti menší soustavě. Velice malou chybu si udržela pouze metoda Orthores, norma chyby je i zde řádově 10^{-6}, ale u ostatních metod se norma chyby pod tuto hranici nedostala. V případě nejjednodušší metody MR je tato norma dokonce špatná. Na tomto grafu tedy vidíme, že nejrychlejší metody Orthomin(1) a GMRES(k) pro $k = 40$, resp. $k = 20$, resp. $k = 30$ nemusí dávat nejpřesnější řešení. Naopak metoda Orthores je sice trochu pomalejší, ale dává docela přesnou aproximaci.

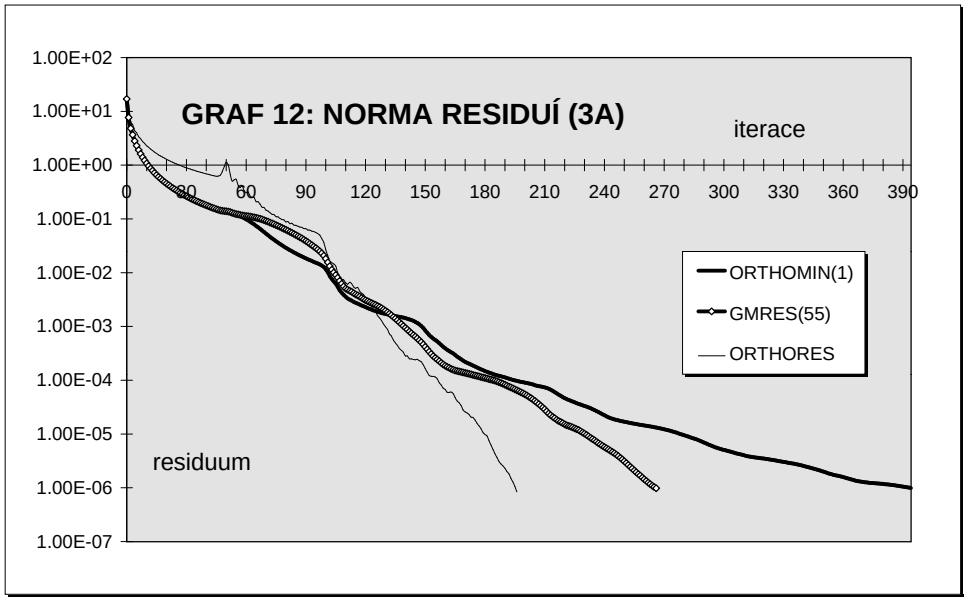
Nyní přejdeme k největší matici velikosti **4900 krát 4900** s koeficienty

$$A = B = C = D = 1, \quad E = F = G = 0.$$

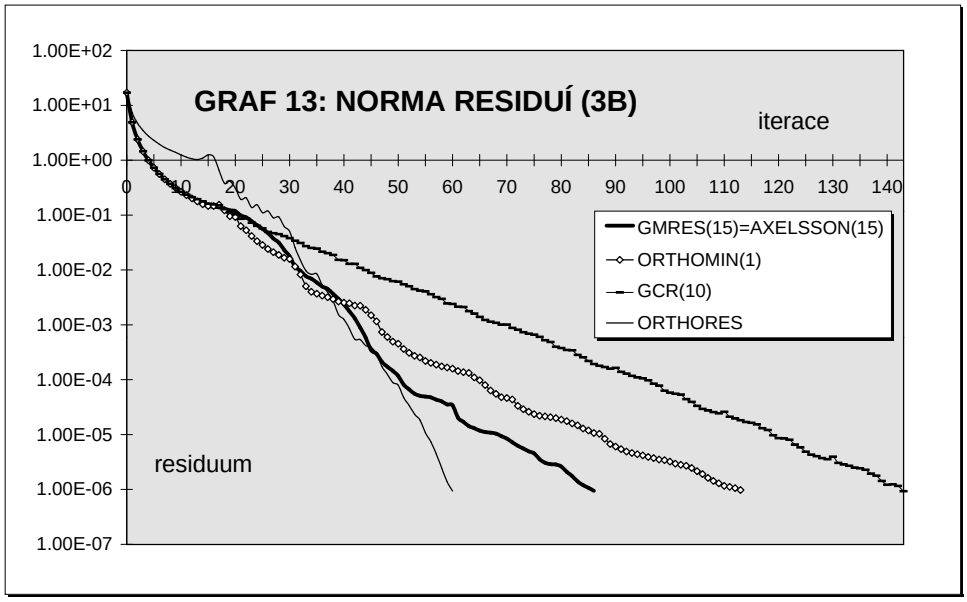
Zde jsou již větší rozdíly.



Bez předpodmínění jsou výpočty velmi pomalé. U ostatních metod je to ještě horší. V případě levého předpodmínění je rozdíl patrný u metody GMRES, u ostatních metod převažuje počet násobení vektorů s maticí Q_1^{-1} a proto jsou časy přibližně stejné ve srovnání se systémem bez předpodmínění. Nejlépe obstálo předpodmínění zprava, kde bylo u většiny metod dosaženo dobrých výsledků. Podívejme se tedy na křivky normy residuí. Nejprve pro nepředpodmíněnou soustavu.

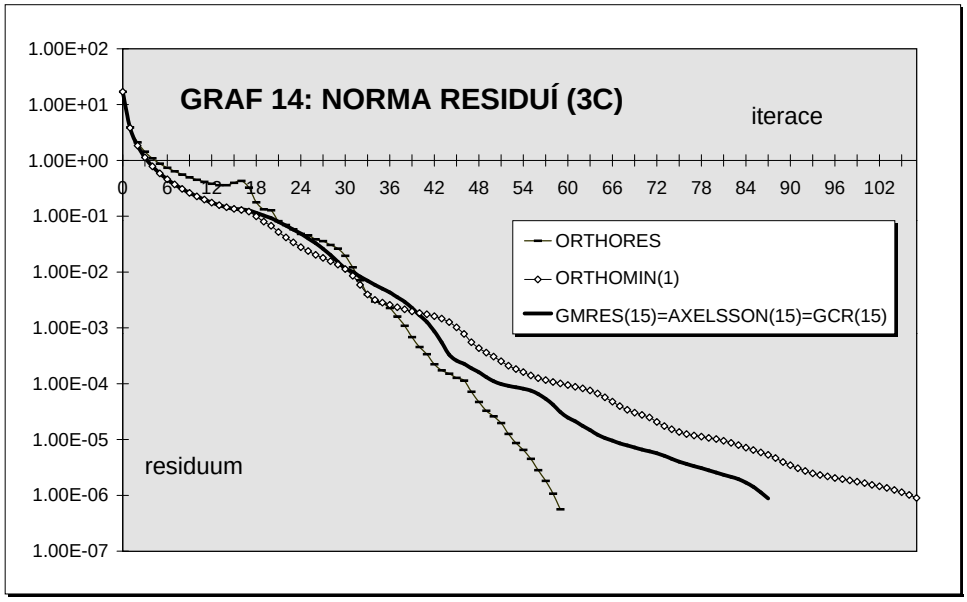


Stejně jako výše, i tady potřebovala nejméně iterací metoda Orthores. Všimněme si zde, i na ostatních grafech, že průběh křivky Orthores je všude podobný, a to i vzhledem k ostatním metodám. Nyní předpodmíníme soustavu zleva.

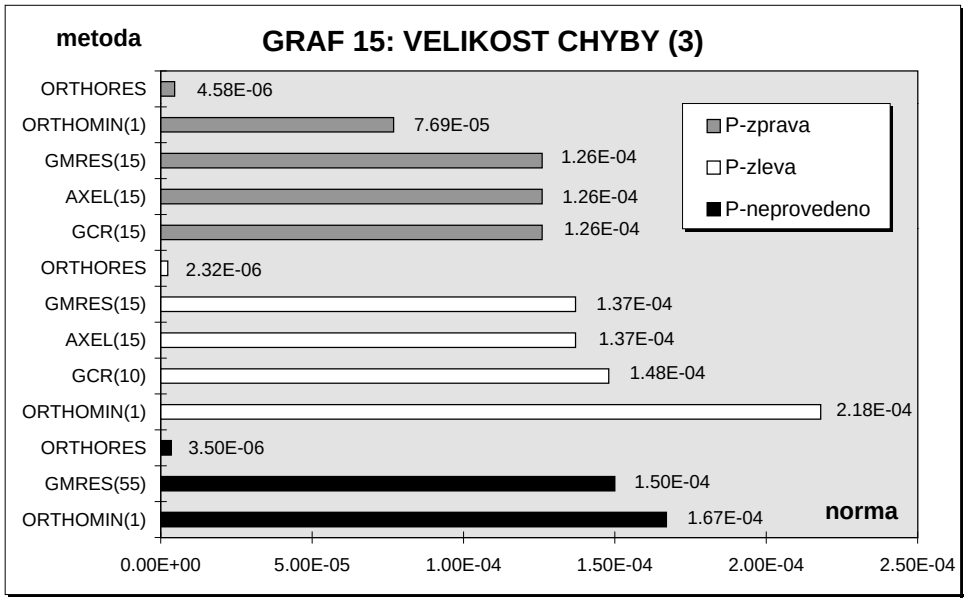


Metody GMRES(15) a Axelsson(15) dávají stejnou posloupnost residuí, avšak časově se hodně liší. Připojili jsme rovněž restartovanou metodu GCR(10), která však není příliš rychlá na rozdíl od useknuté verze GCR, zvané Orthomin(k), jež se ukázala být jednou z nejrychlejších metod. Nejlepší hodnotou pro metodu Orthomin(k) se opět ukázalo $k = 1$. Pro větší k již program pracuje s více vektory p_j, a výpočet je tudíž pomalejší.

Všechny čtyři metody dávají stejnou posloupnost residuí během prvních zhruba dvaceti iterací. Liší se pouze opět metoda Orthores, která je ovšem pomalejší než v případě nepředpodmíněné soustavy. Můžeme říci, že předpodmínění zleva nedalo nejuspokojivější výsledky z časového hlediska, ale co se týče počtu iterací, tak jasně předčilo systém bez předpodmínění. Podívejme se, jak to vypadá, když předpodmíníme soustavu zprava.

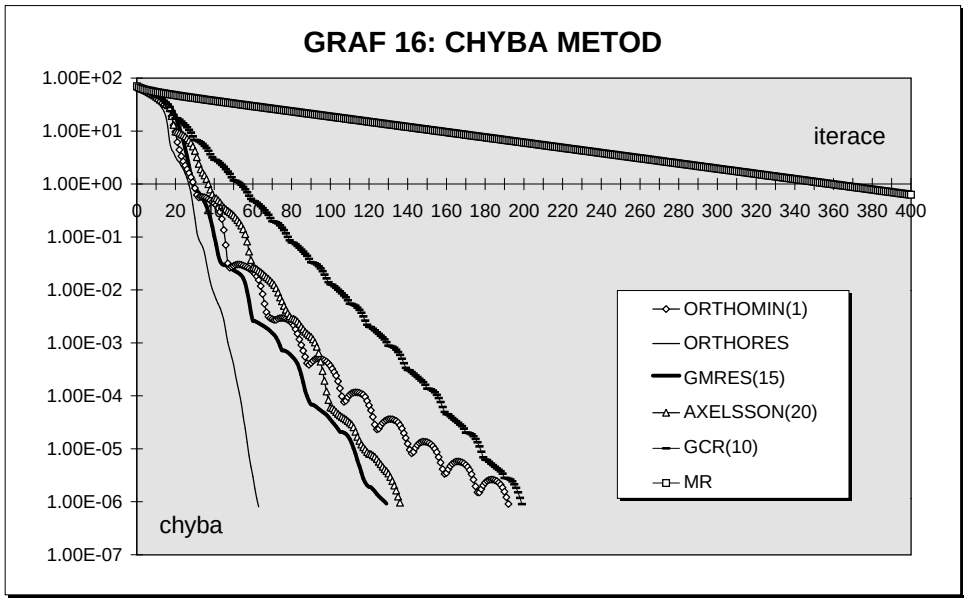


Z grafu 14 je vidět, že tento způsob předpodmínění se jeví jako nejrychlejší. Metody GMRES(k), Axelsson(k) a GCR(k) jsou nejrychlejší shodně pro $k = 15$ a dávají tudíž stejnou posloupnost residuí. Srovnáme-li však časy mezi těmito třemi metodami dohromady, zjistíme, že nejlépe na tom je metoda GMRES(15). Další metoda Orthomin(1) potřebovala stejně jako dříve nejvíce iterací k získání potřebně malého residua. Totéž se dá říci i o metodě Orthores, která opět potřebovala iterací nejméně.
Nejzajímavější jsou výsledky srovnání normy chyby, jak uvidíme na následujícím grafu.



Metody GMRES(15), Axelsson(15) a (u předpodmínění zprava) GCR(15) dávali stejnou posloupnost residuí, proto je stejná i chyba. Všimněme si hlavně rozdílu mezi normou chyby metody Orthores a zbylými metodami, a to ve všech případech předpodmínění systému (tj. 0, L, P). Kromě metody Orthores a eventuelně Orthomin(1) u předpodmínění zprava nedává žádná metoda uspokojivě přesné řešení. Pokud bychom chtěli docílit menší chyby a tedy přesnějšího řešení, řekněme na úroveň metody Orthores, museli bychom provést více iterací. Tím se ovšem výpočet ještě zpomalí a zvětší se rozdíl rychlosti výpočtu a počtu iterací mezi těmito metodami a metodou Orthores. Mimo to, metoda Orthores potřebuje nejméně iterací ve všech případech předpodmínění. Po této úvaze bychom mohli dát metodě Orthores primát nejlepší metody, protože se pro soustavu předpodmíněnou zprava ukázala být nejrychlejší, nejpřesnější a spotřebovala nejméně iterací oproti ostatním metodám a navíc u této metody nemusíme volit hodnotu k.

Nakonec se pro představu podívejme, jak v tomto případě vypadají křivky velikosti chyby. Zde uvidíme největší rozdíly oproti ostatním případům. U každé metody se provedlo tolik iterací, aby pro normu chyby platilo $\| x_i \| < 10^{-6}$. U metod Axelsson(k) a GCR(k) jsme vzali jiné hodnoty k než 15, konkrétně 20, resp. 10, abychom viděli, jak moc se odlišují.



Protože počáteční přiblížení je vektor $x_0 = (1, ..., 1)^T$, je jeho Eukleidovská norma rovna 70. Na grafu 16 vidíme, že skutečně velikost chyby u metody Orthores klesá velmi rychle, zatímco u dalších metod je klesání pomalejší. Nejjednodušší metoda MR si i zde zachovává lineární klesání stejně jako v případě normy residua na grafu 9. Všimněme si, že velikost chyby se dostala pod nulu až u 400. iterace. Norma residua je v tuto chvíli řádově 10^{-2}. Odtud je jasné, že tato metoda potřebuje ještě hodně moc iterací ke zkonvergování.

Dále si všimněme, že metoda Orhomin(1) začíná po padesáti iteracích dávat vždy konstantní oblouk, který se zhušťuje a metoda GCR(10) klesá téměř lineárně.

Poznamenáváme, že obdobný průběh grafu dostáváme i pro ostatní případy uvedené

výše, tj. pro matice dimenze 900 i 2500 a to jak bez předpodmínění, tak i s oběma typy předpodmínění. Proto na grafech 5, 10 a 15 znázorňujeme pouze normy iterací, které se počítají jako poslední. Na grafu 16 pro největší matici můžeme nejlépe vidět jednotlivé rozdíly v průběhu křivek.

4.3 Závěr

Podobným způsobem jako výše bychom mohli testovat metody na stále větších a větších soustavách. Z uvedených grafů vidíme, že se metody chovají ve všech případech podobně. Řešíme-li malou soustavu, vystačíme s jakoukoli metodou. Pro středně velké soustavy je počet vhodných metod již omezen a pro větší soustavy lze doporučit jen některé metody. Řešíme-li ještě větší a větší soustavy (dimenze 7225, 10000), průběh grafů je podobný, avšak výpočty se čím dál více zpomalují.

Metoda Orthodir se hodí jen pro malé soustavy, pro střední a větší nastávají problémy, pokud není počáteční x_0 blízko přesnému řešení. Pro metodu Orthomin(k) musíme vždy najít správnou hodnotu k, abychom měli výpočet co nejrychlejší. Pro střední a větší soustavy jsme však zjistili, že stačí volit $k = 1$, pro větší hodnoty k jsou výpočty pomalejší. Totéž platí pro restartovanou metodu GMRES(k), kde ovšem jsou optimální hodnoty k různé, podle toho, jak velkou soustavu řešíme a z které strany ji předpodmíníme. U metody Orthores se žádné k nevolí, ale soustavu musíme předpodmínit zprava (jinak nedostaneme rychlé výsledky). V tomto případě se tato metoda stává jednou z nejrychlejších a nejpřesnějších metod. Optimální hodnota k pro restartovanou metodu Axelsson(k) se zvětšuje se zvětšováním dimenze řešeného systému. Pro menší soustavy je to zhruba $k = 12$, pro větší soustavy $k = 15$ a pro ještě větší soustavy dimenze 7225 je $k = 20$.

Bez předpodmínění se dají řešit jen velmi malé soustavy, použijeme-li předpodmínění, pak jasně nejlepší je soustavu předpodmínit zprava, abychom dostali výsledky rychle a spolehlivě. Počet násobení maticí Q_2^{-1} je méně než počet násobení maticí Q_1^{-1}, a proto je předpodmínění zprava rychlejší než zleva.

Na závěr řekneme, které metody vyšly z těchto pokusů nejlépe. Jsou to tyto metody:

- Orthomin(k) ... algoritmus 1.5 , stačí $k = 1$ – rychlá metoda, potřebuje však více iterací ke konvergenci a tento počet se zvyšuje s dimenzí řešeného systému;
- Orthores ... algoritmus 1.18 – pro předpodmínění zprava nejrychlejší metoda, která spotřebuje nejméně iterací, pro zkonvergování nepotřebuje příliš zvětšovat počet iterací s rostoucí dimenzí systémů, zřejmě vůbec nejlepší metoda;
- Axelsson(k), restartovaná verze ... algoritmus 2.2 – pro správné hodnoty k, zhruba mezi 10 a 25 podle velikosti systému, získáváme sice pomalejší, zato spolehlivou metodu obdobně jako Orthomin(k);
- GMRES(k) ... algoritmus 3.4 – pro různé hodnoty k však dostáváme různě rychlé procesy s rozdílnými počty iterací a tyto rozdíly mohou být dost velké, pro optimální k je to jedna z nejlepších metod.

Tím jsme ukončili numerické experimenty a vyslovili zhodnocení.

Literatura

[Axel 1] Owe Axelsson : Conjugate Gradient Type Methods for Unsymm. and Inconsistent Systems of Lin. Eq., Linear Algebra and Its Applications 29:1-16, 1980.

[Axel 2] Owe Axelsson : A Generalized Conjugate Gradient Least Square Method, Department of Mathematics, Catholic University, Toernooiveld, NL-6525 ED Nijmegen, The Netherlands.

[Elman] Howard C. Elman : It. Methods for Large, Sparse, Nonsymm. Systems of Lin. Eq., A Dissertation Presented to the Faculty of the Graduate School of Yale Univ. in Candidacy for the Degree of Doctor of Philosophy, May 1982.

[Gant] F.R.Gantmacher : The Theory of Matrices, vol. I,II. New York: Chelsea 1959.

[Golub] Gene H. Golub, Charles F. van Loan : Matrix Computations 1989, Johns Hopkins University Press, Baltimore, MD

[Saad 1] Youcef Saad : Krylov Subspace Methods for Solving Large Unsymmetric Linear Systems. Mathematics of Computation 37:105-126, 1981.

[Saad 2] Youcef Saad, Martin H. Schultz : A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems, Department of Computer Science, Yale University, New Haven, Connecticut 06520.

[Young] David M. Young, Kang C. Jea : Generalized Conjugate Gradient Acceleration of Nonsymmetrizable Iterative Methods. Linear Algebra and Its Applications 34:159-194, 1980.

[Weiss] R.Weiss : A theoretical overview of Krylov subspace methods, Applied Numerical Mathematics 19 (1995) 207-233.