ADAPTIVNÍ PŘEDPODMÍNĚNÍ METODY GMRES*

Jan Zítko, Jurjen D. Tebbens

Abstract

Jednou z nejčastěji užívaných metod pro řešení velkých řídkých soustav lineárních rovnic s nehermitovskou maticí je restartovaná metoda GMRES(m). Tento postup ovšem někdy stagnuje a celá řada prací se zabývá tím, jak tuto stagnaci odstranit. V tomto článku se věnujeme konstrukci adaptivního předpodmínění, které bylo studováno v pracích [1, 2, 3]. Cílem článku je shrnout přehledně hlavní výsledky publikované v [1, 2, 3] a [12] a objasnit principy konstrukce jednak předpodmínění, které se obecně mění restart od restartu, a jednak polynomiálního filtru studovaného v práci [12]. V článku uvádíme i algoritmus který je zpracovaný s podrobnými poznámkami podle [1]. Numerické výsledky uzavírají celou práci.

1. Úvod

Mnoho problémů ve fyzice, chemii a inženýrské praxi může být popsáno soustavou parciálních diferenciálních rovnic. Numerické řešení takových soustav metodou konečných diferencí, konečných prvků nebo konečných objemů vede na úkol řešit obrovské soustavy lineárních algebraických rovnic obvykle se řídkou maticí. Použití přímých metod se pro takové soustavy ukazuje nevhodné a ve většině případů nepoužitelné protože i současné počítače nemají k dispozici požadovanou paměť. V poslední době se velmi intenzivně studují jednak multigridní metody a jednak Krylovovské metody (Krylov subspace methods). Myšlenka Krylovovských metod je taková, že se snažíme sestrojit prostor malé dimenze $\mathcal{K}$ a vektor (řekněme x_0) tak, abychom ve varietě $x_0+\mathcal{K}$ našli dobrou aproximaci k řešení původní soustavy. Když se nám toto povede, převedli jsme řešení obsáhlého systému na řešení úlohy malé dimenze. Tento postup ovšem může také selhat. Nejčastější těžkost, která se nám může vyskytnout je stagnace celého procesu. Bylo vyvinuto několik postupů, které stagnaci odstraní. A jedním z nich je adaptivní předpodmínění, o kterém pojednává následující článek. Abychom mohli celý postup korektně vyložit, musíme přistoupit k rigorózní matematické formulaci.

Uvažujme soustavu lineárních algebraických rovnic

$$Ax = b, \quad A \in \mathbb{C}^{n\times n}; \quad x, b \in \mathbb{C}^n. \tag{1.1}$$

Předpokládáme, že A je regulární, nehermitovská matice. Předpoklad, že A je nehermitovská matice bychom sice mohli škrtnout, avšak my chceme zdůraznit, že nás v tomto článku budou zajímat postupy pro soustavy s nehermitovskou a v reálném případě nesymetrickou maticí, speciálně metoda GMRES a její modifikace. Pro soustavy se symetrickou a pozitivně definitní maticí bychom uživateli doporučili metodou sdružených gradientů s předpodmíněním, pro menší soustavy Choleského rozklad. Nechť x_0 je počáteční aproximace k řešení (1.1), $r_0 = b - Ax_0 \neq 0$ odpovídající reziduální vektor a

*Tento článek byl napsán díky podpoře grantů GAČR 201/00/0080 a MSM 113200007.

$$\mathcal{K}_k(A, r_0) = \text{ span}\{r_0, Ar_0, \ldots, A^{k-1}r_0\}$$

Krylovův prostor vytvořený maticí A a vektorem r_0. Princip metody GMRES spočívá v tom, že aproximaci x_k sestrojujeme tak aby $x_k - x_0 \in \mathcal{K}_k(A, r_0)$ a aby euklidovská norma rezidua $r_k = b - Ax_k$ byla minimální. Metoda GMRES patří do třídy metod, které se v literatuře uvádějí pod názvem "Conjugate Krylov Subspace Iterative Methods". Tato metoda byla poprve atraktivně publikována v práci [8].

Stručně o metodě GMRES.

U metody GMRES se k-tá approximace k řešení $x^\star$ určuje z podmínky

$$x_k = \arg \min_{x \in x_0 + \mathcal{K}_k(A, r_0)} \|b - Ax\|, \tag{1.2}$$

nebo jinak zapsáno

$$x_k = x_0 + \arg \min_{u \in \mathcal{K}_k(A, r_0)} \|r_0 - Au\|. \tag{1.3}$$

Protože $Au \in A\mathcal{K}_k(A, r_0)$, můžeme psát $Au = S_k z$, kde $S_k = (Ar_0, A^2 r_0, \ldots, A^k r_0)$, a $z \in \mathbb{C}^k$. Pak

$$r_k = b - Ax_k = r_0 - S_k z_k, \quad \text{kde} \quad z_k = \arg \min_{z \in \mathbb{C}^k} \|r_0 - S_k z\|. \tag{1.4}$$

Vypočítáme z_k ze druhého vztahu (1.4), dosadíme do první rovnosti pro r_k a dostaneme

$$r_k = r_0 - \underbrace{S_k (S_k^H S_k)^{-1} S_k^H}_{=P_k} r_0. \quad \text{Je vidět, že } P_k = P_k^2 \text{ a } P_k = P_k^H.$$

V praxi se minimum normy $\|r_k\|$ počítá tak, že se sestrojí orthonormální báze $v_1, v_2, \ldots, v_k$ v prostoru $\mathcal{K}_k(A, r_0)$ na příklad modifikovaným Arnoldiho procesem, takže $\mathcal{K}_k(A, r_0) = \text{span}\{v_1, \ldots, v_k\}$. Označme $V_k = (v_1, v_2, \ldots, v_k)$, $V_{k+1} = (V_k, v_{k+1})$. Arnoldiho procesem zároveň sestrojíme horní Hessenbergovu matici $\tilde{H}_k \in \mathbb{C}^{(k+1)\times k}$ tak, že $AV_k = V_{k+1}\tilde{H}_k$.

Praktická implementace GMRES.

Vektor $u \in \mathcal{K}_k(A, r_0)$, můžeme po konstrukci ortonormální báze napsat ve tvaru $u = V_k z$. Spočítáme-li podle toho co bylo výše řečeno

$$z_k = \arg \min_{z \in \mathbb{C}^k} \|r_0 - AV_k z\|, \tag{1.5}$$

pak $x_k = x_0 + V_k z_k$. Pro zkrácení označme $\beta = \|r_0\|$. Pak

$$\|r_0 - AV_k z\| = \|\beta V_{k+1} e_1^{(k+1)} - V_{k+1}\tilde{H}_k z\| = \|\beta e_1^{(k+1)} - \tilde{H}_k z\|. \tag{1.6}$$

Provedeme QR rozklad matice $\tilde{H}_k$, t.j., $\tilde{H}_k = Q_k R_k$, kde $Q_k \in \mathbb{C}^{(k+1)\times(k+1)}$ je unitární a $R_k \in \mathbb{C}^{(k+1)\times k}$ horní trojúhelníková s nulovým posledním řádkem. Pokračujeme-li v úpravách, obdržíme

$$\|\beta e_1^{(k+1)} - \tilde{H}_k z\|^2 = \|\beta e_1^{(k+1)} - Q_k R_k z\|^2 = \|\beta Q_k^H e_1^{(k+1)} - R_k z\|^2. \tag{1.7}$$

Položme $g_k = \beta Q_k^T e_1^{k+1}$. Je-li $\hat{g}_k \in \mathbb{C}^k$ vector sestávající se z prvních k složek vektoru g_k a je-li $\hat{R}_k \in \mathbb{C}^{k\times k}$ čtvercová matice, kterou obdržíme škrtnutím posledního nulového řádku v matici R_k, pak ze vztahu (1.7) je vidět, že z_k je jediné řešení rovnice $\hat{R}_k z = \hat{g}_k$ a poslední složka vektoru g_k je normou rezidua r_k.

Aplikujeme-li uvedený postup na našich počítačích, zjistíme, že u velkých soustav selhává z důvodů nedostačující kapacity počítače. Jestliže totiž k roste, pak růst počtu operací je vyjádřen symbolem $O(k^2 n)$ a nároky na paměť symbolem $O(kn)$.

Jak se těmto těžkostem vyhnout?

- useknout ortogonalizační proces,
- restartovat GMRES,
- restartovat useknutou verzi.

Nejvíce se v literatuře studuje restartovaná verze GMRES(m). Avšak restart zpomaluje konvergenci a často dochází ke stagnaci procesu. Nejčastěji si pomáháme předpodmíněním. Hodně se používá neúplný LU-rozklad. V této práci pojednáme o adaptivním předpodmínění. Během jednoho restartu metody GMRES(m) získáme řadu informací (Hessenbergovu matici, ortonormální bázi Krylovova podprostoru), které se při dalších restartech ztratí, aniž by se využily. Musíme si nejprve říci, čeho konstrukcí adaptivního předpodmínění chceme dosáhnout. Je známo, že přítomnost malých vlastních čísel ve spektru matice soustavy může způsobit zpomalení konvergence eventuelně stagnaci. Dejme si tedy za cíl sestrojit po každém restartu matici M tak, aby matice $M^{-1}A$ neměla ve spektru malá vlastní čísla a to právě na základě znalostí Hessenbergovy matice a ortonormální báze Krylovova podprostoru. Toto předpodmínění, které se mění restart od restartu nevylučuje navíc použití LU-rozkladu. Celá myšlenka vypadá velmi rozumně. Tato práce si dala za cíl zpracovat dostupnou literaturu v pojednání, ve kterém bude vysvětleno, jak se adaptivní předpodmínění konstruuje a hlavně proč se to tak dělá. Závěr práce je doplněn numerickými experimenty. V celé práci používáme euklidovskou normu vektoru a matice.

2. Adaptivně předpodmíněná metoda GMRES(m)

Nejprve popíšeme Arnoldiho faktorizaci a QR-rozklad s použitím posunutí. Podrobnosti je možné nalézt v práci ([12]). Mějme dán Arnoldiho rozklad, t.j., matici V s ortonormálními sloupci, Hessenbergovu matici H a vektor f tak, že platí

$$AV = VH + fe_*^T, \quad V^H f = 0,$$

nebo jinak zapsáno

$$AV = (V, v)\begin{pmatrix} H \\ \gamma e_*^T \end{pmatrix},$$

kde $\gamma = \|f\|$, $v = \frac{1}{\gamma} f$ a za $*$ se dosadí dimenze Krylovova prostoru.
Další krok: matice a vektory sestrojené v dalším kroku Arnoldiho procesu označíme křížkem v indexu. Následující vzorce popisují tento krok a jsou seřazeny tak, že je možné si v jedné až třech dimenzích celý proces geometricky představit.

$$V_+ = (V, v), \quad w = Av, \quad \begin{pmatrix} h \\ \alpha \end{pmatrix} = V_+^H w,$$

$$
\begin{aligned}
H_+ &= \begin{pmatrix} H & h \\ \gamma e_*^T & \alpha \end{pmatrix} \\
f_+ = w - V_+ \begin{pmatrix} h \\ \alpha \end{pmatrix} &= (I - V_+ V_+^H) w.
\end{aligned}
$$

Odtud je již vidět, že

$$AV_+ = V_+ H_+ + f_+ e_{*+1}^T.$$

Jak v dalším uvidíme, půjde nám o to najít počáteční vektor v Arnoldiho rozkladu tak aby V byl invariantní vlastní podprostor matice A odpovídající nejmenším vlastním číslům. Pro tento účel provedeme následující úvahu.

Nechť k, p jsou přirozená čísla a nechť z Arnoldiho rozkladu po $k+p$ krocích obdržíme rovnost

$$
\begin{aligned}
AV_{k+p} &= V_{k+p} H_{k+p} + f_{k+p} e_{k+p}^T \qquad (2.1) \\
&= (V_{k+p}, v_{k+p+1}) \begin{pmatrix} H_{k+p} \\ \gamma_{k+p} e_{k+p}^T \end{pmatrix}.
\end{aligned}
$$

Nechť μ_* je posunutí, $(H_{k+p} - \mu_* I) = Q_* R_{k+p}$, kde Q_* je unitární a R_{k+p} je horní trojúhelníková. Poznamenejme, že Q_* je horní Hesenbergova matice. Z rovnosti (2.1) plynou snadno následující vztahy:

$$
\begin{aligned}
(A - \mu_* I) V_{k+p} - V_{k+p}(H_{k+p} - \mu_* I) &= f_{k+p} e_{k+p}^T, \qquad (2.2) \\
(A - \mu_* I) V_{k+p} - V_{k+p} Q_* R_{k+p} &= f_{k+p} e_{k+p}^T, \\
(A - \mu_* I)(V_{k+p} Q_*) - (V_{k+p} Q_*)(R_{k+p} Q_*) &= f_{k+p} e_{k+p}^T Q_*, \\
A \underbrace{(V_{k+p} Q_*)}_{V_{k+p}^+} - \underbrace{(V_{k+p} Q_*)}_{V_{k+p}^+} \underbrace{(R_{k+p} Q_* + \mu_* I)}_{H_{k+p}^+} &= f_{k+p} e_{k+p}^T Q_*. \qquad (2.3)
\end{aligned}
$$

A nyní si představme, že v rovnosti (2.3) škrtneme + a provedeme celý postup od (2.2) k (2.3) s jiným posunutím. Za $*$ dosadíme postupně indexy 1,2,…,p. Je jasné, že po provedení p posunutí dostaneme

$$AV_{k+p}^+ = (\underbrace{V_{k+p}^+}_{V_{k+p}\hat{Q}}, v_{k+p+1}) \begin{pmatrix} H_{k+p}^+ \\ \gamma_{k+p} e_{k+p}^T \hat{Q} \end{pmatrix}, \quad \text{kde} \quad H_{k+p}^+ = \hat{Q}^H H_{k+p} \hat{Q},$$

$\hat{Q} = Q_1 Q_2 \dots Q_p$, Q_j je unitární a V_{k+p} a H_{k+p} jsou matice definované vztahem (2.1). Provedeme rozdělení na bloky

$$V_{k+p}^+ = (V_k^+, \hat{V}_p), \quad H_{k+p}^+ = \begin{pmatrix} H_k^+ & M \\ \hat{\gamma}_k e_1 e_k^T & \hat{H}_p \end{pmatrix},$$

kde označíme $\gamma_{k+p} e_{k+p}^T \hat{Q} = (\underbrace{0, 0, \dots, \hat{\gamma}_{k+p}}_{k}, \underbrace{c^H}_{p})$. Porovnáním dostaneme

$$AV_k^+ = V_k^+ H_k^+ + f_k^+ e_k^T, \qquad (2.4)$$

kde $f_k^+ = \hat{V}_p e_1 \hat{\gamma}_k + v_{k+p+1}\hat{\gamma}_{k+p}$. Je vidět, že $(V_k^+)^H f_k^+ = 0$, takže (2.4) je Arnoldiho rozklad. Označíme-li $\gamma_k^+ = \|f_k^+\|$, $v_{k+1}^+ = f_k^+/\gamma_k^+$, pak

$$AV_k^+ = (V_k^+, v_{k+1}^+) \begin{pmatrix} H_k^+ \\ \gamma_k^+ e_k^T \end{pmatrix}.$$

Jak lze těchto posunutí (shiftů) použít ke konstrukci invariantního podprostoru uvidíme v dalším. Nejprve si dokažeme následující dvě věty. (Viz. [12].)

Věta 2.1. *Nechť po k krocích Arnoldiho faktorizace obdržíme vztah $AV_k - V_k H_k = f_k e_k^T$ přičemž $f_j \neq 0$ pro $j \in \{1, 2, \ldots, k-1\}$. Pak $f_k = 0$ právě když $v_1 = Ty$, kde $T \in \mathbb{C}^{n\times k}$, $AT = TJ$, $\operatorname{rank}(T) = k$ a $J \in \mathbb{C}^{k\times k}$ je Jordanova matice.*

Důkaz. Je-li $f_k = 0$ a $H_k\hat{T} = \hat{T}J$, kde J je Jordanův kanonický tvar matice H_k, položme $T = V_k\hat{T}$. Pak $AT = TJ$, $\operatorname{rank}(T) = k$ a

$$v_1 = V_k e_1 = V_k \hat{T}\hat{T}^{-1} e_1 = Ty, \quad \text{kde} \quad y = \hat{T}^{-1} e_1.$$

Obráceně: Nechť $AT = TJ$, $\operatorname{rank}(T) = k$ a $v_1 = Ty$. Pak $A^m T = TJ^m$ pro libovolné nezáporné celé m, $A^m v_1 = A^m Ty = TJ^m y \in \mathcal{R}(T)$ $\forall m$ a tedy $\dim(\mathcal{K}_{k+1}(A, v_1)) \leq k = \operatorname{rank}(T)$; $\dim(\mathcal{K}_j(A, v_1)) = j$ pro $j \in \{1, 2, \ldots k\}$ a tedy $f_k = 0$. Poznamenejme, že $\mathcal{R}(T)$ značí obor hodnot lineárního operátoru (matice) T. $\square$

A nyní přistoupíme k důkazu věty (viz. [12]) která nám napoví, jak by mohl vypadat iterační proces na sestrojení invariantního podprostoru.

Věta 2.2. *Nechť spektrum matice H_{k+p} se sestává z následujících dvou množin:*

$$\lambda(H_{k+p}) = \{\theta_1, \theta_2, \ldots, \theta_k\} \cup \{\mu_1, \mu_2, \ldots, \mu_p\}.$$

Nechť Q_j je definováno posunutím μ_j, $Q = Q_1 Q_2 \ldots, Q_p$ a $H_{k+p}^+ = Q^H H_{k+p} Q$. Je-li $\gamma_j \neq 0$ pro $j = 1, 2, \ldots, k-1$, pak $\gamma_k = 0$ a

$$H_{k+p}^+ = \begin{pmatrix} H_k^+ & M^+ \\ 0 & R_p \end{pmatrix},$$

kde $\lambda(H_k^+) = \{\theta_1, \theta_2, \ldots, \theta_k\}$, $\lambda(R_p) = \{\mu_1, \mu_2, \ldots, \mu_p\}$ a R_p je horní trojúhelníková matice. Dále platí

$$v_1^+ = V_{k+p} Q e_1 = \sum_{j=1}^{k} \zeta_j x_j$$

kde $x_j = V_{k+p}\, y_j$ a $H_{k+p}\, y_j = \theta_j y_j$ pro $j \in \{1, 2, \ldots, k\}$.

Důkaz. Po p posunutích dostaneme

$$\psi(A) \underbrace{v_1}_{V_{k+p}e_1} = V_{k+p} Q e_1, \quad \text{kde} \quad \psi(\lambda) = \frac{1}{\varrho} \Pi_{j=1}^{p} (\lambda - \mu_j), \tag{2.5}$$

kde ϱ je normující parametr. Z první rovnosti (2.5) plyne

$$Qe_1 = V_{k+p}^T \psi(A) V_{k+p} e_1 = \psi(H_{k+p}) e_1 \overset{\text{def}}{=} q_1$$

a tudíž ve spektrálním rozkladu vektoru q_1 chybí vlastní vektory odpovídající vlastním číslům μ_j, t.j. $q_1 = \sum_{j=1}^{k} \zeta_j y_j$ pro vhodná komplexní čísla ζ_j. Označme matici $T = (y_1, y_2, \ldots, y_k)$, takže $H_{k+p}T = T\mathrm{diag}(\theta_1, \theta_2, \ldots, \theta_k)$. Z konstrukce posunutí plyne, že

$$H_{k+p}Q = QH_{k+p}^{+}. \tag{2.6}$$

Uvažujeme-li nyní jenom prvních k resp. $k+1$ sloupců matice Q na levé resp. pravé straně rovnosti (2.6), obdržíme Arnoldiho rozklad matice H_{k+p} se startujícím vektorem q_1. Avšak $q_1 = Qe_1$ je lineární kombinací vlastních vektorů $y_1, y_2, \ldots, y_k$ matice H_{k+p} příslušných vlastním číslům $\theta_1, \theta_2, \ldots, \theta_k$. Dosadíme-li si ve větě 2.1 za matici A matici H_{k+p}, za vektor v_1 vektor q_1, dále J =diag$(\theta_1, \theta_2, \ldots, \theta_k)$ a za V_k matici sestavenou z prvních k sloupců matice Q, vidíme podle věty 2.1 že $\gamma_k = 0$. Navíc je $v_1^{+} = VQ_1e_1 = Vq_1 = \sum_{j=1}^{k} \zeta_j V y_j = \sum_{j=1}^{k} \zeta_j x_j$, kde $x_j = Vy_j$ jsou Ritzovy vektory. Tvar matice H_{k+p}^{+} a množin $\lambda(H_k^{+})$ a $\lambda(R_p)$ obdržíme vhodnou organizací QR-rozkladů uvědomíme-li si, že matice $H - \mu_j I$ jsou singulární. $\square$

Řekněme si nyní, jaká je interpretace poslední věty. Pro zjednodušení předpokládáme, že Hessenbergova matice H_{k+p} je diagonalizovatelná a rozdělíme její spektrum na dvě skupiny. První skupina vlastních čísel, ve větě 2.2 označená $\theta_1, \theta_2, \ldots, \theta_k$, je v literatuře pojmenována jako "hledaná nebo žádoucí" vlastní čísla a druhá skupina, ve větě 2.2 označená $\mu_1, \mu_2, \ldots, \mu_p$, je nazývána "nežádoucí" vlastní čísla. Jestliže provedeme posunutí s nežádoucími vlastními čísly, obdržíme pro Arnoldiho faktorizaci nový startující vektor $v_1 = \sum_{j=1}^{k} \zeta_j x_j$ který je lineární kombinací Ritzových vektorů svázaných s hledanými vlastními čísly. Vztah (2.4) představuje nyní Arnoldiho rozklad, kde span$\{V_k^{+}\} = \mathcal{K}_k(A, v_1)$ a $v_1 = \sum_{j=1}^{k} \zeta_j x_j$. Na základě těchto úvah si sestavme následující iterační proces. Provedeme $k+p$ kroků Arnoldiho rozkladu. Obdržíme matici H_{k+p} a spočteme její vlastní čísla, která seřadíme podle velikosti. Provedeme s p v absolutní hodnotě největšími vlastními čísly posunutí podle formulí (2.2)-(2.3) a obdržíme matici H_{k+p}^{+} a vektor v_1^{+}, který ovšem nemusíme počítat. Podle (2.4) stačí vyjít od H_k^{+} a V_k^{+} a dopočítat zbývajících p vektorů. Dostaneme opět novou matici H_{k+p}^{+}, spočítáme její vlastní čísla, srovnáme podle velikosti v absolutní hodnotě a provedeme posunutí. A tento proces stále opakujeme. Dostaneme tímto iterační proces, od kterého očekáváme, že f_k^{+} bude konvergovat k nule. Podrobněji viz práci [12]. Pokud je f_k^{+} ve (2.4) rovno nule a jsou-li (θ_j, y_j) pro $j \in \{1, 2, \ldots, k\}$ vlastní dvojice matice H_k^{+}, pak $(\theta_j, V_k y_j)$ jsou vlastní dvojice matice A. Tímto postupem můžeme tudíž získat invariantní podprostor matice A příslušný malým vlastním číslům matice A.

V předcházejících odstavcích jsme popsali, jak sestrojit startovací vektor v Arnoldiho procesu tak, aby dobře aproximoval lineární kombinací vlastních vektorů, které odpovídají malým vlastním číslům matice A. Uvedeme si v dalším algoritmické zpracování této myšlenky. K sestavení algoritmu jsme použili hlavně [1].

V dalších úvahách budeme předpokládat,že matice A je diagonalizovatelná, t.j. nechť

$$A = Z\Lambda Z^{-1}, \qquad \text{kde} \qquad \Lambda = \mathsf{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n).$$

Předpokládejme, že

$$0 < |\lambda_1| \le |\lambda_2| \le \ldots \le |\lambda_n| = 1.$$

Uvažujme jeden restartovaný běh GMRES(m); ve srovnání s předchozím označením je $m = k + p$ a tedy $p = m - k$. Napišme si Arnoldiho rozklad:

$$\begin{aligned} AV_m &= V_m H_m + f_m e_m^T, \\ V_m &\in \mathbb{C}^{n\times m}, \quad f_m \in \mathbb{C}^n, \quad \gamma_m = \|f_m\|, \\ V_m^H V_m &= I_m, \quad V_m^H f_m = 0, \quad H_m \in \mathbb{C}^{m\times m}. \end{aligned}$$

Nechť matice $V_k \in \mathbb{C}^{n\times k}$ se sestává z prvních k sloupců matice V_m a nechť sloupce matice W_{n-k} představují ortogonální doplněk k prostoru span$\{V_k\}$, takže $W_{n-k}^H W_{n-k} = I_{n-k}$ a $\mathbb{C}^n =$ span$\{V_k\}\oplus$ span$\{W_{n-k}\}$; následující dvě věty nám ukazují, jak konstruovat předpodmínění, které eliminuje vliv malých vlastních čísel. Položme $M = V_k H_k V_k^H + W_{n-k} W_{n-k}^H$.

Věta 2.3. *Nechť $Q \in \mathbb{C}^{n\times n}$ je unitární matice $Q = (V\ W)$, kde $V \in \mathbb{C}^{n\times k}$. Předpokládejme, že matice $H = V^H AV$ je regulární. Pak matice*

$$M = VHV^H + WW^H \tag{2.7}$$

je regulární a její inverze je dána formulí

$$M^{-1} = VH^{-1}V^H + WW^H. \tag{2.8}$$

Důkaz. Správnost tvrzení je ihned vidět z následujících dvou rovností:

$$M = (V\ W)\begin{pmatrix} H & 0 \\ 0 & I_{n-k}\end{pmatrix}\begin{pmatrix} V^H \\ W^H\end{pmatrix}, \quad M^{-1} = (V\ W)\begin{pmatrix} H^{-1} & 0 \\ 0 & I_{n-k}\end{pmatrix}\begin{pmatrix} V^H \\ W^H\end{pmatrix}. \quad \square$$

Věta 2.4. *Nechť matice V, W, H mají stejný význam jako ve větě 2.3. Předpokládejme navíc, že* span$\{V\}$*je invariantní podprostor matice A odpovídající vlastním číslům $\lambda_1, \lambda_2, \ldots, \lambda_k$. Pak*

$$\lambda(M^{-1}A) = \{\lambda_{k+1}, \lambda_{k+2}, \ldots, \lambda_n, \underbrace{1, 1, \ldots, 1}_{k}\} \tag{2.9}$$

přičemž vlastní číslo 1 má násobnost alespoň k.

Důkaz. Položíme-li

$$\tilde{A} = \begin{pmatrix} V^H \\ W^H\end{pmatrix} A(V\ W) = \begin{pmatrix} H & \tilde{A}_{12} \\ 0 & \tilde{A}_{22}\end{pmatrix},$$

pak $\lambda(\tilde{A}_{22}) = \{\lambda_{k+1}, \lambda_{k+2}, \ldots, \lambda_n\}$ *a*

$$M^{-1}A = (V\ W)\begin{pmatrix} I_k & H^{-1}\tilde{A}_{12} \\ 0 & \tilde{A}_{22}\end{pmatrix}\begin{pmatrix} V^H \\ W^H\end{pmatrix}.$$

Z poslední rovnosti je již vidět tvrzení věty. $\square$

Z uvedených vět je vidět, že konstrukce matice předpodmínění je založena na konstrukci invariantního podprostoru odpovídajícímu malým vlastním číslům a k jeho konstrukci použijeme polynomiálního filtru, který byl vyložen na začátku tohoto paragrafu. Arnoldiho rozklad budeme potřebovat tak, jak je formulován v následujícím algoritmu.

ALGORITMUS 1. - ARNOLDIHO PROCES.

1. ***Input:*** k, $m > k$, $H_k \in \mathbb{C}^{k\times k}$, $V_k \in \mathbb{C}^{n\times k}$, $f_k \in \mathbb{C}^n \setminus \{0\}$, *such that* $V_k^H V_k = I_k$, $V_k^H f_k = 0$, $AV_k = V_k H_k + f_k e_k^T$;

2. ***Output:*** $H_m \in \mathbb{C}^{m\times m}$, $V_m \in \mathbb{C}^{n\times m}$, $f_m \in \mathbb{C}^n \setminus \{0\}$, *such that* $V_m^T V_m = I_m$, $V_m^H f_m = 0$, $AV_m = V_m H_m + f_m e_m^T$;

3. $\gamma_k := \|f_k\|$;

4. **if** $k > 0$ **then** $h_{k+1,k} := \gamma_k$; **endif**
   for $j = k+1, k+2, \ldots, m$ *do*
   $\quad v_j := f_{j-1}/\gamma_{j-1}$; $f_j = Av_j$;
   $\quad$**for** $l = 1, 2, \ldots, j$ **do**
   $\quad\quad h_{l,j} := v_l^T f_j$; $f_j := f_j - h_{l,j} v_l$;
   $\quad$**endfor** l;
   $\quad \gamma_j := \|f_j\|$; $h_{j+1,j} := \gamma_j$;
   endfor j;

Protože chceme uvažovat iterační proces pro konstrukci invariantního podprostoru, budeme předpokládat, že všechna f_j jsou nenulová. Označíme-li ještě $V_{m+1} = [V_m \;\; f_m/\gamma_m]$, je $\widehat{H}_m = V_{m+1}^H A V_m$. Napišme-li si ještě jednou formule (2.2)-(2.3) pro posunutí μ_1 a restart m, dostaneme

$$\begin{aligned}
(A-\mu_1 I)V_m - V_m(H-\mu_1 I) &= f_m e_m^T, && (2.10)\\
(A-\mu_1 I)V_m - V_m Q^{(1)} R^{(1)} &= f_m e_m^T, &&\\
(A-\mu_1 I)(V_m Q^{(1)}) - (V_m Q^{(1)}) R^{(1)} Q^{(1)} &= f_m e_m^T Q^{(1)}, &&\\
A\underbrace{(V_m Q^{(1)})}_{V_m^{(1)}} - (V_m Q^{(1)}) \underbrace{(R^{(1)} Q^{(1)} + \mu_1 I)}_{H^{(1)}} &= f_m e_m^T Q^{(1)}. && (2.11)
\end{aligned}$$

$$(2.12)$$

Aplikujeme-li $m-k$ posunutí $\mu_1, \mu_2, \ldots, \mu_{m-k}$, dostaneme rozklad

$$AV_m^{(m-k)} = V_m^{(m-k)} H_m^{(m-k)} + f_m e_m^T Q, \tag{2.13}$$

kde

$$Q = Q^{(1)} Q^{(2)} \ldots Q^{(m-k)},$$
$$V_m^{(m-k)} = (v_1^{(m-k)}, v_2^{(m-k)}, \ldots, v_m^{(m-k)}) = V_m Q, \quad H_m^{(m-k)} = Q^H H_m Q.$$

Rozdělme matici $H_m^{(m-k)}$ na bloky,

$$H_m^{(m-k)} = \begin{pmatrix} H_k^{(m-k)} & G \\ \tilde{\gamma}_k e_1 e_k^T & \widehat{H}_{m-k} \end{pmatrix},$$

dosaďme do (2.13) a srovnejme prvních k sloupců. Označme-li $V_m^{(m-k)} = (V_k^{(m-k)}, \tilde{V}_{m-k}^{(m-k)})$, dostaneme rovnici

$$\begin{aligned}
AV_k^{(m-k)} &= V_k^{(m-k)} H_k^{(m-k)}\\
&+ \underbrace{\tilde{V}_{m-k}^{(m-k)} e_1 \tilde{\gamma}_k e_k^T + f_m e_m^T Q e_k e_k^T}_{(*)}
\end{aligned}$$

Ve $(*)$ vytkneme e_k^T, to co zůstane označíme $f_k^{(m-k)}$ a dostaneme

$$AV_k^{(m-k)} = V_k^{(m-k)} H_k^{(m-k)} + f_k^{(m-k)} e_k^T, \tag{2.14}$$

což je Arnoldiho rozklad matice A se startujícím vektorem $v_1^{(m-k)} = \delta p(A) v_1$, kde $p(x) = (x-\mu_1)(x-\mu_2)\dots(x-\mu_{m-k})$, $v_1 = V_m e_1$ a δ je normující parametr.
Nechť $\{\theta_j^m\}_{j=1}^m \dots$jsou vlastní čísla matice H_m,

$$|\theta_1^m| \leq |\theta_2^m| \leq \dots \leq |\theta_m^m|. \tag{2.15}$$

Položme $\mu_j = \theta_{m+1-j}^{(m)}$ pro $j \in \{1, 2, \dots, m-k\}$. Nechť $\{\theta_j^{(k)}, y_j^{(k)}\}_{j=1}^k$, jsou vlastní dvojice $H_k^{(m-k)}$ a definujme

$$x_j = V_k^{(m-k)} y_j^k, \quad j = 1, 2, \dots k.$$

Pak pro $j \in \{1, 2, \dots, k\}$ je podle (2.13)

$$\|Ax_j - x_j\theta_j^{(k)}\| = \|(AV_k^{(m-k)} - V_k^{(m-k)} H_k^{(m-k)}) y_j^{(k)}\| = \|f_k^{(m-k)}\| \, |e_k^T y_j^{(k)}|. \tag{2.16}$$

Řekneme, že span$\{V_k^{(m-k)}\}$ je dobrá aproximace k invariantnímu podprostoru matice A je-li

$$\|f_k^{(m-k)}\| \, |e_k^T y_j^{(k)}| \leq \|H_k^{(m-k)}\| \, \epsilon \tag{2.17}$$

pro všechna $j \in \{1, 2, \dots, k\}$, kde ϵ je v datech zadaná tolerance. Pokud zadané přesnosti nedosáhneme, spočteme vlastní čísla aktuální Hessenbergovy matice $H_m^{(m-k)}$, provedeme posunutí a tento proces můžeme několikrát opakovat. Podrobně je vše patrné z algoritmu, který následuje. Označme výsledné matice $V_k^{(m-k)}$ a $H_k^{(m-k)}$ jako V_k a H_k. První matice pro předpodmínění je tedy $M_1^{-1} = V_k H_k^{-1} V_k^H + I_n - V_k V_k^H$. Celý proces nyní aplikujeme na systém $M_1^{-1} Ax = M_1^{-1} b$ a sestrojíme matici M_2 a systém $M_2^{-1} M_1^{-1} Ax = M_2^{-1} M_1^{-1} b$. V algoritmu se tato konstrukce opakuje α_0-krát. Výsledná matice předpodmínění je $M^{-1} = M_{\alpha_0}^{-1} M_{\alpha_0 - 1}^{-1} \dots M_1^{-1}$. Skálování systému (1.1) které se provádí kvůli tomu, aby největší v absolutní hodnotě vlastní číslo matice A se rovnalo 1 se provádí faktorem $1/|\theta_m^{(m)}|$, který určíme z Hessenbergovy matice, kterou vypočteme napoprvé. A nyní budou následovat dva algoritmy a sice nejprve na pár řádcích připomeneme restartovanou metodu GMRES(m) a poté bude následovat metoda GMRES(m) s adaptivním předpodmíněním, které eliminuje k vlastních čísel, kterou budeme značit GMRES(m, k). Připomeňme, že adaptivní se tomuto předpodmínění říká proto, že se sestrojuje z informací získaných během jednoho restartu a v každém restartu se mění.

ALGORITMUS 2. - GMRES(m)

1. ***Input:*** *m, x_0, ε;*
2. ***Output:*** *přibližné řešení x_m a odpovídající r_m;*
3. *$r_m := r_0 := b - Ax_0$; $x_m := x_0$;*
4. **while** $\|r_m\|/\|r_0\| > \varepsilon$ **do**
   Proveď ALGORITMUS 1 *pro $k = 0$ a $f_0 = r_m$; tím získáme $\tilde{H}_m$ a V_{m+1};*
   Vypočti $y_m \in \mathbb{R}^m$ řešením úlohy $\min_{y \in \mathbb{R}^m} \| \, \|r_m\| e_1 - \tilde{H}_m y \|$;
   Vypočti $x_m := x_m + V_m y_m$; $r_m := r_m - V_{m+1} \tilde{H}_m y_m$;
   endwhile;

ALGORITMUS 3. - ADAPTIVNĚ PŘEDPODMÍNĚNÁ METODA GMRES(m)

1. ***Input:*** x_0,– *počáteční aproximace;*
   ε– *tolerance pro ukončení výpočtu;*
   ϵ– *tolerance pro výpočet "chtěných" vlastních dvojic;*
   m– *restart;*
   k – *dimenze počítaného invariantního podprostoru;*
   α_0 – *maximální počet počítaných matic předpodmínění* M_j^{-1}*;*
   β_0 – *v jednom restartu je* β_0 *maximální počet Arnoldiho rozkladů do řádu* m *ve kterých se používá posunutí.*
2. ***Output:*** *přibližné řešení* x_j, *odpovídající* r_j, *a výsledné předpodmínění* $M^{-1} = M_\alpha^{-1}M_{\alpha-1}^{-1}\dots M_1^{-1}$ *pro nějaké* $\alpha \le \alpha_0$.
3. ***Položíme*** $x_0 := 0; r_0 := b;\ j = 0;\ M^{-1} = I$.
4. ***Cyklus***, *který registruje kolikrát se adaptuje předpodmínění.*

   for $\alpha = 1, 2, \dots, \alpha_0$ **do**
   Provedeme Arnoldiho rozklad s počátečním vektorem $v_1 = M^{-1}r_j/\|M^{-1}r_j\|$,
   jehož výsledkem je rovnost $M^{-1}AV_m = V_mH_m + f_me_m^T$*;*
   spočteme $V_{m+1}, \tilde{H}_m$;
   určíme $y_m \in \mathbb{C}^m$ *jakožto řešení úlohy* $\min_{y\in\mathbb{C}^m} \| \|M^{-1}r_j\|e_1 - \tilde{H}_my\|$;
   spočteme
   $x_{j+m} := x_j + V_my_m$;
   $M^{-1}r_{j+m} := M^{-1}r_j - V_{m+1}\tilde{H}_my_m;\quad r_{j+m} := b - Ax_{j+m};\ j := j+m.$

   Cyklus *který testuje, kolik rozkladů s posunutím se maximálně provede v jednom restartu.*

   for $\beta = 1, 2, \dots, \beta_0$ **do**
   Vypočteme $\{\theta_j^{(m)}\}$ *a uspořádáme podle (2.15) a pro* $j = m$
   soustavu vydělíme faktorem $|\theta_m^{(m)}|$.

   Cyklus *který registruje provedení rozkladů (2.10)-(2.11) pro všechna "nechtěná" vlastní čísla*

   for $l = 1, 2, \dots, m-k$ **do**
   S posunutím θ_{m+1-l}^m *provedeme (2.10)-(2.11); tím dostaneme*
   $M^{-1}AV_m^{(l)} = V_m^{(l)}H_m^{(l)} + f_m^{(l)}e_m^T$;
   endfor *l;*
   spočteme matice $H_k^{(m-k)}$ *a* $V_k^{(m-k)}$ *z matic* $H_m^{(m-k)}$ *a* $V_m^{(m-k)}$*;*
   spočteme vlastní dvojice $\{\theta_j^{(k)}, y_j^{(k)}\}_{j=1}^k$ *a přesvědčíme se, zdali je splněno (2.17);*

   if *platí (2.17)* **or** $\beta = \beta_0$ **then goto** *5;*
   jinak užitím Arnoldiho rozkladu
   $M^{-1}AV_k^{(m-k)} = V_k^{(m-k)}H_k^{(m-k)} + f_k^{(m-k)}e_k^T$
   dopočítáme podle algoritmu 1 rozklad až do m *a označme jej*
   $M^{-1}AV_m = V_mH_m + f_me_m^T$;
   provedeme GMRES(m) takto: dopočítáme V_{m+1} *a* $\tilde{H}_m$;

vypočteme $y_m \in \mathbb{C}^m$ *řešením úlohy* $\min_{y\in\mathbb{C}^m} \| V_{m+1}^H M^{-1} r_j - \tilde{H}_m y\|$
[viz. poznámku za algoritmem];
spočteme
$x_{j+m} := x_j + V_m y_m$;
$M^{-1} r_{j+m} := M^{-1} r_j - V_{m+1} \tilde{H}_m y_m$; $r_{j+m} := b - Ax_{j+m}$; $j := j+m$;
endfor β.

5. ***Provedeme*** *GMRES(k) a upravíme matici předpodmínění: máme k dispozici Arnoldiho rozklad*
$M^{-1}AV_k = V_k H_k + f_k e_k^T$ *a matice* V_{k+1} *a* $\tilde{H}_{k+1}$;
vypočteme řešení $\min_{y\in\mathbb{C}^k} \|V_{k+1}^H M^{-1} r_j - \tilde{H}_k y\|$;
spočteme
$x_{j+k} := x_j + V_k y_k$; $M^{-1} r_{j+k} := M^{-1} r_j - V_{k+1} \tilde{H}_k y_k$.

6. ***Adaptace*** *matice předpodmínění*
$M_\alpha^{-1} := V_k (H_k)^{-1} V_k^H + I - V_k V_k^T$;
$M^{-1} := M_\alpha^{-1} M^{-1}$.

7. $r_{j+k} := b - Ax_{j+k}$; $\quad j := j+k$;
if $\|r_j\|/\|r_0\| \le \varepsilon$ **then** *konec výpočtu;*
endfor α.

Pokud jsme s adaptací matice předpodmínění skončili a nedosáhli požadovanou přesnost, aplikujeme až do konvergence na předpodmíněnou soustavu restartovanou metodu GMRES(m).

while $\|r_j\|/\|r_0\| \ge \varepsilon$ **do**
Provedeme Arnoldiho rozklad $M^{-1}AV_m = V_m H_m + f_m e_m^T$ *s počátečním vektorem* $v_1 = M^{-1} r_j / \|M^{-1} r_j\|$; *spočteme* $V_{m+1}, \tilde{H}_m$;
určíme $y_m \in \mathbb{C}^m$ *jakožto řešení úlohy* $\min_{y\in\mathbb{C}^m} \|\|M^{-1} r_j\| e_1 - \tilde{H}_m y\|$;
spočteme
$x_{j+m} := x_j + V_m y_m$;
$M^{-1} r_{j+m} := M^{-1} r_j - V_{m+1} \tilde{H}_m y_m$;
$r_{j+m} := b - Ax_{j+m}$; $j := j+m$;
endwhile

Konec algoritmu 3.

Poznámka 1 (k algoritmu 3). Přesně podle algoritmu GMRESu(m) bychom měli počítat minimum $\|M^{-1} r_j - M^{-1} A V_m y_m\|$. Avšak tato norma je rovna $\|M^{-1} r_j - V_{m+1} \tilde{H}_m y_m\|$. Abychom mohli použít postupu, který je běžný u metody GMRES, potřebujeme eliminovat V_{m+1} ve druhém členu posledního výrazu v normě. Řešme tedy úlohu nalezení vektoru $z \in \mathbb{C}^{m+1}$ tak, aby $\|V_{m+1} z - M^{-1} r_j\|$ byla minimální. Řešením normální rovnice dostaneme $z = (V_{m+1}^H V_{m+1})^{-1} V_{m+1}^H M^{-1} r_j$ a tato úvaha nás vedla k tomu, minimalizovat normu

$$\|V_{m+1} V_{m+1}^H M^{-1} r_j - V_{m+1} \tilde{H}_m y_m\| = \|V_{m+1}^H M^{-1} r_j - \tilde{H}_m y_m\|$$

obvyklým způsobem.

Poznámka 2. Rezidua jsou v algoritmu indexována písmenem j. Vzhledem k tomu, že v cyklu pro β opakujeme GMRES, proces restartujeme a měníme α, zachycuje index j počet iterací (zhruba řečeno počet násobení matice soustavy a vektoru) a r_j počáteční reziduum pro každý aktuálně prováděný proces GMRES(m) resp. GMRES(k). V literatuře se někdy místo označení r_j užívá označení r_0. Vzhledem k tomu, že r_0 značí reziduum úplně na začátku výpočtu a v programu používáme i podíl $\|r_j\|/\|r_0\|$, domníváme se, že je rozumnější naše značení.

3. Numerický experiment

Uvažujeme bidiagonální matici s vlastními čísly na diagonále 0.06, 0.08, 0.1, 4, 5,. . . ,100 a s prvky nad diagonálou 0.1. Pravá strana je rovna vektoru $e = (1, 1, \ldots, 1)^T$ a za x_0 vezmeme nulový vektor. V následující tabulce srovnáme GMRES (full GMRES), GMRES(20), GMRES(20,3) kde za β_0 vezmeme 6 a za α_0 číslo 1. Jak by to vypadalo pro jiná β_0 a α_0 popíšeme za chvíli. V tabulce zaokrouhlujeme na dvě desetinná místa. Zvolili jsme $\varepsilon = 10^{-16}$ a ve druhém a třetím sloupci tabulky jsme zapsali poslední dekadický logaritmus podílu $\|r_j\|/\|r_0\|$ pro který byl tento podíl větší než 10^{-16}. Vzhledem k restartu počítáme každou dvacátou iteraci. GMRES(20) konvergoval pomaleji a tabulku jsme o výpočet pod 10^{-16} neprodlužovali. V následující tabulce jsou pro výše citované tři metody uvedeny hodnoty $\log_{10} \|r_{20}\|/\|r_0\|$, kde $\|r_{20}\|$ je norma vždy po dvacátém restartu a $\|r_0\|$ je norma rezidua na začátku výpočtu.

	$\log_{10} \|r_{20}\|/\|r_0\|$		
Iterace	GMRES(20)	GMRES(20,3)	full GMRES
20	-1.36	-1.36	-1.36
40	-1.52	-1.36	-1.52
60	-1.56	-1.47	-5.44
80	-1.60	-1.81	-13.20
100	-1.63	-1.86(*)	-13.56
120	-1.71	-5.77	
140	-2.05	-9.37	
160	-3.12	-12.81	
200	-3.83		
300	-7.34		
400	-10.74		
440	-11.08		

Hvězdička ve sté iteraci ve sloupci GMRES(20,3) znamená, že v této iteraci byla splněna podmínka (2.17) ke změně β, to znamená, že s přesností ϵ jsme obdrželi invariantní podprostor příslušný třem nejmenším vlastním číslům. Poznamenejme, že v prostředním sloupci počínaje iterací 100 koresponduje odpovídající logaritmus iteraci o $k = 3$ větší, t.j., v tabulce by měly být pro prostřední sloupec ještě řádky 103, 123, ... Z technických důvodů (lámání stránky) jsme toto nedělali.

Historie výpočtu. Podle algoritmu 3 nejprve vypočteme aproximaci k největšímu vlastnímu číslu matice A a vydělíme touto aproximací matici A, pravou stranu a Hessenbergovu matici. Vlastní čísla takto vydělené matice se jen nepatrně lišila od diagonály matice A vydělené číslem 100. S touto maticí dále počítáme, t.j. $A := A/100$, $b := b/100$. Při aplikaci metody GMRES(20,3) jsme zvolili $\epsilon = 10^{-4}$. Pro $\beta = 5$ jsme obdrželi přibližný invariantní vlastní podprostor příslušný třem nejmenším vlastním číslům. Znamená to tedy, že byla splněna podmínka (2.17). Spočítali jsme předpodmínění M_1 a vlastní čísla matice $M_1^{-1}A$. Tato čísla vyšla přibližně 0.04, 0.05, 0.06,..., $\underbrace{1, 1, 1, 1}_{4\times}$. To znamená, že při aplikaci předpodmínění M_1 došlo skoro úplně přesně k eliminaci prvních tří nejmenších vlastních čísel. Zkusili jsme zvětšit α_0 o jedničku, t.j. $\alpha_0 = 2$ a podle algoritmu 3 jsme počítali M_2 a vlastní čísla matice $M_2^{-1}M_1^{-1}A$. K našemu překvapení vyšla přibližně 0.07, 0.08, 0.09, ..., $\underbrace{1, 1, 1, 1, 1, 1, 1}_{7\times}$. Opět tedy došlo k eliminaci dalších tří vlastních čísel, dohromady šesti. Pro $\alpha_0 = 3$ došlo již dosti nepřesně k eliminaci dalších dvou vlastních čísel. Pravděpodobně proto, že jsme pro $\alpha_0 = 2$ dostali reziduum o velikosti 10^{-16} a na dalším výpočtu se projevily silně zaokrouhlovací chyby. V praxi by se totiž při dosažení takové anebo menší přesnosti výpočet ukončil a na výpočet M_3 by již vůbec nedošlo. Pro informaci uveďme ještě tabulku hodnot $\|f_k^{(m-k)}\|\,|e_k^T y_j^{(k)}|$ a $\|H_k^{(m-k)}\|\,\epsilon$ (viz. (2.17). První tabulka je pro $m = 20$, $k = 3$ a $\alpha_0 = 1$.

Iterace	j	$\|f_k^{(m-k)}\|\,|e_k^T y_j^{(k)}|$	$\|H_k^{(m-k)}\|\,\epsilon$
20	1	1.81×10^{-1}	
	2	2.60×10^{-1}	
	3	1.70×10^{-1}	8.7×10^{-4}
40	1	5.71×10^{-2}	
	2	5.51×10^{-2}	
	3	8.12×10^{-4}	5.41×10^{-5}
60	1	2.85×10^{-2}	
	2	7.06×10^{-3}	
	3	2.88×10^{-5}	3.97×10^{-6}
80	1	1.33×10^{-5}	
	2	7.85×10^{-6}	
	3	2.47×10^{-6}	1.64×10^{-6}
100	1	3.61×10^{-8}	
	2	2.35×10^{-8}	
	3	1.04×10^{-8}	1.64×10^{-6}

Vidíme, že po pátém opakování posunutí je skutečně splněna podmínka (2.17).

Nyní ještě uvedeme tabulku pro $m = 20$, $k = 3$ a $\alpha_0 = 2$.

Iterace	j	$\|f_k^{(m-k)}\| \, \|e_k^T y_j^{(k)}\|$	$\|H_k^{(m-k)}\| \, \epsilon$
123	1	$1.11 \; 10^{-2}$	
	2	$1.76 \; 10^{-2}$	
	3	$4.40 \; 10^{-3}$	$9.04 \; 10^{-5}$
143	1	$2.93 \; 10^{-3}$	
	2	$8.00 \; 10^{-3}$	
	3	$1.41 \; 10^{-3}$	$6.01 \; 10^{-5}$
163	1	$9.87 \; 10^{-5}$	
	2	$9.88 \; 10^{-5}$	
	3	$1.96 \; 10^{-6}$	$6.00 \; 10^{-5}$
183	1	$1.21 \; 10^{-6}$	
	2	$4.79 \; 10^{-7}$	
	3	$1.13 \; 10^{-7}$	$6.00 \; 10^{-5}$

Vidíme, že tady je podmínka (2.17) splněna pro $\beta = 4$.

Zbývá udělat ještě jednu velmi důležitou poznámku. V tomto prvním případě byla matice soustavy reálná a všechna vlastní čísla byla reálná. Pokud pracujeme s reálnými soustavami, pak by bylo velmi nešikovné přecházet do komplexní aritmetiky v případě, že Hessenbergova matice má komplexně sdružená vlastní čísla. V tomto případě zůstaneme dále v reálné aritmetice a využijeme toho, že vlastní čísla jsou komplexně sdružená. Podrobněji viz. [5] (second edition pp.376-další). Zde jenom uvedeme bez důkazu, jak se to dělá. Nechť μ_1 a μ_2 jsou dvě komplexně sdružená vlastní čísla, se kterými provádíme posunutí. Napišme si oba rozklady:

$$
\begin{aligned}
H - \mu_1 I = Q_1 R_1, \quad H_1 = R_1 Q_1 + \mu_1 I, \\
H_1 - \mu_2 I = Q_2 R_2, \quad H_+ = R_2 Q_2 + \mu_2 I,
\end{aligned}
$$

Matice $G = (H - \mu_1 I)(H - \mu_2 I) = H^2 - (\mu 1 + \mu_2) H + \mu 1 \mu_2 I$ je reálná a platí pro ni rozklad

$$
G = (Q_1 Q_2)(R_1 R_2),
$$

což jest QR-rozklad reálné matice G. Podle teorie uvedené ve výše citované knize mohou být matice Q_1 a Q_2 vybrány tak, aby matice $S = Q_1 Q_2$ byla reálná ortogonální matice. Pak matice

$$
H^+ = Q_2^H H_1 Q_2 = Q_2^H (Q_1^H H Q_1) Q_2 = (Q_1 Q_2)^H H (Q_1 Q_2) = S^T H S
$$

je opět reálná. Matici H^+ je tudíž reálná a algoritmus na výpočet reálné matice H^+ po dvou komplexně sdružených posunutích je podle této úvahy velmi jednoduchý. Vypočteme reálný QR-rozklad $G = SR$ matice G (toto se provádí užitím Householderových transformací) a položíme $H^+ = S^T H S$. Technické provedení je popsáno v [5].

Pro lepší představu uvažujme ještě jeden praktický numerický příklad.

2) Uvažujme konenčně diferenční aproximaci rovnice

$$-e^{-xy}\triangle u + (10 + ye^{-xy})u_x + (10 + xe^{-xy})u_y - 60u = 1,$$

na jednotkovém čtverci s Dirichletovou okrajovou podmínkou $u = 0$ na hranici. Konečně diferenční aproximací na síti 20×20 resp. 30×30 obdržíme soustavu o 400 resp. 900 neznámých. Pravá strana a x_0 je jako v předchozím případě. Zvolíme $\alpha_0 = \beta_0 = 1$, $\epsilon = 10^{-4}$ a $\varepsilon = 10^{-8}$ Následující tabulka srovnává full GMRES, GMRES(30) a GMRES(30,25) pro matici 400×400 a opět jako v předchozím případě srovnáváme $\log_{10} \|r_{20}\|/\|r_0\|$.

Iterace	GMRES	GMRES(30)	GMRES(30,25)
30	-0.19	-0.18	-0.19
60	-1.64	-0.20	-0.19
90	-3.79	-0.20	-1.21
120	-5.84	-0.20	-2.57
150	konvergence	-0.20	-3.65
180		-0.20	-4.61
210		-0.20	-5.78
240		-0.20	-6.59
270		-0.20	-7.16
300		stagnace	konvergence

Následuje tabulka ve které pro matici 900×900 srovnáme full GMRES, GMRES(40), GMRES(40,35) a GMRES(40,39).

Iterace	GMRES	GMRES(40)	GMRES(40,35)	GMRES(40,39)
40	-0.14	-0.14	-0.14	-0.14
120	-2.91	-0.15	-0.76	-0.99
200	-5.89	-0.15	-2.91	-3.12
240	konvergence	-0.15	-3.69	-3.85
280		-0.15	-4.36	-4.58
320		-0.15	-4.87	-5.17
360		-0.15	-5.52	-5.76
400		-0.15	-5.91	-6.23
440		-0.15	-6.43	-6.79
480		-0.15	-6.93	-7.26
520		stagnace	-7.71	konvergence

4. Poznámka k adaptivnímu předpodmínění zprava

Arnoldiho proces se v tomto případě aplikuje na matice AM^{-1}, vypočítá se V_m. Po m iteracích je $x_m := x_0 + M^{-1}V_m y_m$, kde $y_m = \arg\ \min_{y\in\mathbb{C}^m} \|\beta e_1 - \tilde{H}_m y\|$. Podrobně je vše vyloženo v práci [2]. Pro důkaz se mlčky předpokládá, že matice A je diagonalizovatelná s vlastními čísly

$$|\lambda_1| \le |\lambda_2| \le \dots |\lambda_n|.$$

Předpokládejme, že P je invariantní podprostor dimenze r odpovídající r nejmenším vlastním číslům a U je ortonormální báze v P.

Věta 4.1. *Nechť $T = U^H AU$ a $M = I_n + U(1/|\lambda_n|T - I_r)U^H$. Pak matice M je regulární, $M^{-1} = I_n + U(|\lambda_n|T^{-1} - I_r)U^H$ a vlastní čísla matice AM^{-1} jsou*

$$\lambda_{r+1}, \lambda_{r+2}, \dots, \lambda_n, |\lambda_n|, \quad kde$$

$|\lambda_n|$ má násobnost alespoň r.

Myšlenka důkazu. Nechť $Z = (U, W)$ je ortonormální báze v $\mathbb{C}^n$;

$$A = (U\ W) \underbrace{\begin{pmatrix} T & \tilde{A}_{12} \\ 0 & \tilde{A}_{22} \end{pmatrix}}_{\tilde{A}} \begin{pmatrix} U^H \\ W^H \end{pmatrix}.$$

Dále je

$$Z^H M Z = \begin{pmatrix} T/|\lambda_n| & 0 \\ 0 & I_{n-r} \end{pmatrix} \stackrel{\text{def}}{=} \tilde{M}$$

a

$$\tilde{M^{-1}} = \begin{pmatrix} |\lambda_n|T^{-1} & 0 \\ 0 & I_{n-r} \end{pmatrix}.$$

Konečně

$$\begin{aligned} M^{-1} &= Z\tilde{M^{-1}}Z^H = I_n + U(|\lambda_n|T^{-1} - I_r)U^H, \\ & \quad Z^H AM^{-1}Z = \tilde{A}\tilde{M}^{-1} \\ \tilde{A}\tilde{M}^{-1} &= \begin{pmatrix} |\lambda_n|I_r & \tilde{A_{12}} \\ 0 & \tilde{A_{22}} \end{pmatrix}. \end{aligned}$$

ALGORITMUS 4. - SESTROJENÍ INVARIANTNÍHO PODPROSTORU

1. ***Input:*** *$\varepsilon \dots$ tolerance pro normu rezidua;*
   $x_0 \dots$ počáteční aproximace;
   $\{r_j\}$ posloupnost přirozených čísel.
2. ***Output:*** *přibližné řešení x_j, odpovídající r_j, a výsledné předpodmínění.*
3. ***Položíme*** *convergence:=* **false;** $M := I_n$; $U := \{\}$; $j := 1$.
4. ***Cyklus****, který registruje adaptaci předpodmínění a ukončení výpočtu.*

5. **until** *convergence* **do**
$r_0 = b - Ax_0$; *aplikujeme Arnoldiho proces s maticí* AM^{-1}, *obdržíme* V_m *a řešíme* $\min_{y\in\mathbb{C}^m} \|\beta e_1 - \tilde{H}_m y\|$; *spočteme* $x_m := x_0 + M^{-1}V_m y_m$;

6. **if** $\|b - Ax_r\| < \varepsilon$ *convergence:=true*
else
Spočteme $H_m = V_m^H AM^{-1}V_m$, $r(j)$ *nejmenších vlastních čísel* H_m *a odpovídajících vlastních vektorů* $S_{r(j)}$;

    if *j=1* **then**

    položíme $U = V_m S_{r(j)}$; *určíme aproximaci k* $|\lambda_n|$, *r=r(j); j:= j+1;*

    else

    ortogonalizujeme sloupce matice $V_m S_{r(j)}$ *proti* U

    a zvětšíme U *o* $V_m S_{r(j)}$, $j := j + 1$; $r = r + r(j)$;

    endif

Můžeme tudíž sestrojit $\bar{T}$ *a* $\bar{M}^{-1}$ *a teoreticky i* $\bar{Z}$, $\bar{A}$ *a* $\bar{E} = \bar{A}\bar{M}^{-1}$. *Přitom*

$$\bar{E} = \begin{pmatrix} |\bar{\lambda_n}|I_r & \bar{A_{12}} \\ |\bar{\lambda_n}|\bar{A_{21}}\bar{T}^{-1} & \tilde{A_{22}} \end{pmatrix}.$$

---

7. *Spočteme* $T = U^H AU$. $M^{-1} = I_n + U(|\lambda_n|T^{-1} - I_r)U^H$;
endif
enddo

Tomuto algoritmu se říká DEFLGMRES(m, l). Podrobněji je vše uvedeno v práci [2].

Vidíme, že celý problém je opět postaven na tom, jak přesně dovedeme určit invariantní vlastní podprostor odpovídající několika nejmenším vlastním číslům.

References

[1] Baglama J, Calvetti D, Golub GH, Reichel L. Adaptively Preconditioned GMRES Algorithms. *SIAM J. Sci. Comput.* 1998; 20(1): 243–269.

[2] Burrage K, Erhel J. On the Performance of Various Adaptive Preconditioned GMRES Strategies. *Numer. Linear Algebra Appl.* 1998; 5: 101–121.

[3] Erhel J, Burrage K, Pohl B. Restarted GMRES Preconditioned by Deflation. *Journal of Computational and Applied Mathematics* 1996; 69: 303–318.

[4] Morgan RB. A restarted GMRES method augmented with eigenvectors. *SIAM J. Matrix Anal. Appl.* 1995; 16(4): 1154–1171.

[5] Golub GH, Van Loan CHF. *Matrix Computation.* The John Hopkins University Press, Baltimore, MD, 1984.

[6] Householder AS. *The Theory of Matrices in Numerical Analysis.* Blaisdell: New York, 1964.

[7] Elman HC Iterative methods for large sparse nonsymmetric systems of linear equations. Ph. D. thesis. Computer Science Department, Yale University, New Haven, CT, 1982.

[8] Saad Y, Schultz MH. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing,* 1986; 7: 856–869.

[9] Saad Y. *Iterative Methods for Sparse Linear Systems.* PWS Publishing Company, 1996.

[10] Saad Y. Further Analysis of Minimum Residual Iterations. *Numer. Linear Algebra Appl.* 2000; 7: 67–93.

[11] Zítko J. Generalization of convergence conditions for a restarted GMRES. *Numer. Linear Algebra Appl.* 2000; 7: 117–131.

[12] Sorensen D.C. Implicit application of polynomial filters in k-step Arnoldi method. *SIAM J. Matrix Anal. Appl.*, Vol. 13, No 1, pp. 357-385, 1992.