4. Metode, ki temeljijo na Petrov-Galerkinovem pristopu

Če matrika A ni hermitska, potem pri Arnoldijevem algoritmu dobimo Hessenbergovo matriko. Ker je potrebno vsak nov bazni vektor podprostora Krilova ortogonalizirati na vse prejšnje, ne moremo dobiti algoritmov s kratko rekurzijo.

Do kratkih rekurzij lahko pridemo, če delamo z biortogonalnimi bazami. To pomeni, da zgradimo biortogonalni bazi za podprostora Krilova $\mathcal{K}_k(A, v_1)$ in $\mathcal{K}_k(A^T, w_1)$, tako da:

- stolpci $V_k = [v_1 \ \cdots \ v_k]$ tvorijo bazo za $\mathcal{K}_k(A, v_1)$,
- stolpci $W_k = [w_1 \ \cdots \ w_k]$ tvorijo bazo za $\mathcal{K}_k(A^T, w_1)$,
- velja $v_i^T w_j = 0$ za $i \neq j$ (in $v_i^T w_i = 1$).

4.1 Dvostranski Lanczosev algoritem za biortogonalizacijo

izberi vektorja v_1, w_1, da je $v_1^T w_1 = 1$
$\beta_0 = \delta_0 = 0$, $v_0 = w_0 = 0$
$j = 1, 2, \ldots, k$
$\quad \alpha_j = (Av_j, w_j)$
$\quad \widetilde{v}_{j+1} = Av_j - \alpha_j v_j - \beta_{j-1} v_{j-1}$
$\quad \widetilde{w}_{j+1} = A^T w_j - \alpha_j w_j - \delta_{j-1} w_{j-1}$
$\quad \delta_j = |(\widetilde{v}_{j+1}, \widetilde{w}_{j+1})|^{1/2}$
$\quad$ če je $\delta_j = 0$, prekini računanje
$\quad \beta_j = (\widetilde{v}_{j+1}, \widetilde{w}_{j+1})/\delta_j$
$\quad v_{j+1} = \widetilde{v}_{j+1}/\delta_j$
$\quad w_{j+1} = \widetilde{w}_{j+1}/\beta_j$

Dobimo

$$AV_k = V_k T_k + \delta_k v_{k+1} e_k^T,$$

$$A^T W_k = W_k T_k^T + \beta_k w_{k+1} e_k^T,$$

$$W_k^T A V_k = T_k,$$

kjer je

$$T_k = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \delta_1 & \alpha_2 & \ddots & \\ & \ddots & \ddots & \beta_{k-1} \\ & & \delta_{k-1} & \alpha_k \end{bmatrix}.$$

Skalarja β_j in δ_j nista enolično določena, važno je le, da velja $\beta_j \delta_j = \widetilde{v}_{j+1}^T \widetilde{w}_{j+1}$.

Izrek 1. *Če se algoritem ne zalomi do koraka k, potem velja*

- $v_i^T w_j = 0$ *za* $i \neq j$ *in* $v_i^T w_i = 1$ *za* $i, j = 1, \ldots, k$,
- *stolpci* $V_k = [v_1 \ \cdots \ v_k]$ *tvorijo bazo za* $\mathcal{K}_k(A, v_1)$,
- *stolpci* $W_k = [w_1 \ \cdots \ w_k]$ *tvorijo bazo za* $\mathcal{K}_k(A^T, w_1)$.

Bi-Lanczos

izberi vektorja v_1, w_1, da je $v_1^T w_1 = 1$
$\beta_0 = \delta_0 = 0$, $v_0 = w_0 = 0$
$j = 1, 2, \ldots, k$

$$\alpha_j = (Av_j, w_j)$$
$$\widetilde{v}_{j+1} = Av_j - \alpha_j v_j - \beta_{j-1} v_{j-1}$$
$$\widetilde{w}_{j+1} = A^T w_j - \alpha_j w_j - \delta_{j-1} w_{j-1}$$
$$\delta_j = |(\widetilde{v}_{j+1}, \widetilde{w}_{j+1})|^{1/2}$$

če je $\delta_j = 0$, prekini računanje

$$\beta_j = (\widetilde{v}_{j+1}, \widetilde{w}_{j+1})/\delta_j$$
$$v_{j+1} = \widetilde{v}_{j+1}/\delta_j$$
$$w_{j+1} = \widetilde{w}_{j+1}/\beta_j$$

Metoda se zalomi, če je $\delta_j = 0$. Do tega lahko pride zaradi:

- $\widetilde{v}_{j+1} = 0$: Če rešujemo sistem $Ax = b$ in je $v_1 = r_0 = b - Ax_0$ je to srečni zalom.
- $\widetilde{w}_{j+1} = 0$: Če rešujemo sistem z matriko A^T je to srečni zalom, sicer pa ne pomaga.
- $\widetilde{v}_{j+1} \neq 0$, $\widetilde{w}_{j+1} \neq 0$, toda $\widetilde{v}_{j+1}^T \widetilde{w}_{j+1} = 0$: To je resni zalom.

Kaj narediti, če se Bi-Lanczos zalomi

Če se Bi-Lanczos zalomi, imamo več možnosti:

- Restart: Takoj ko se zalomi (oziroma ko je numerično δ_j blizu 0), naredimo ponovni zagon. Na ta način sicer nadaljujemo, ampak izgubimo podprostor Krilova in možnost superlinearne konvergence.
- Look-Ahead: Pokvarimo tridiagonalno strukturo, tako da uporabimo več vektorjev in pridemo do bločno biortogonalne baze in bločne tridiagonalne matrike T_k. To pomeni, da lahko npr. najdemo par (v_{j+2}, w_{j+2}), čeprav par (v_{j+1}, w_{j+1}) ne obstaja.

Reševanje sistema preko Bi-Lanczosa [Lanczos (1952)]

Rešujemo linearni sistem $Ax = b$. Začetni približek je x_0.

Vzamemo $v_1 = r_0/\|r_0\|$, izberemo w_1 in zgradimo biortogonalni bazi V_k in W_k za podprostora Krilova $\mathcal{K}_k(A, v_1)$ in $\mathcal{K}_k(A^T, w_1)$.

Sedaj iščemo $x_k = x_0 + V_k y_k$, da je $W_k^T(b - Ax_k) = 0$, torej

$$W_k^T(b - Ax_0 - AV_k y_k) = 0,$$

od koder sledi $T_k y_k = \|r_0\| e_1$.

$r_0 = b - Ax_0$

$v_1 = r_0/\|r_0\|$, izberi w_1, da je $v_1^T w_1 = 1$

Bi-Lanczos$(A, v_1, w_1, k) \Longrightarrow$
$$\begin{aligned} AV_k &= V_k T_k + \delta_k v_{k+1} e_k^T \\ A^T W_k &= W_k T_k^T + \beta_k w_{k+1} e_k^T \end{aligned}$$

$y_k = T_k^{-1}\|r_0\| e_1$

$x_k = x_0 + V_k y_k$

Za ostanek velja $\|r_k\| = \delta_k |e_k^T y_k| \cdot \|v_{k+1}\|$.

4.2 Bi-CG [Fletcher (1974)]

Na podoben način, kot smo pri s.p.d. matriki iz D-Lanczosa dobili metodo CG, sedaj iz Bi-Lanczosa dobimo metodo bikonjugiranih gradientov (Bi-CG).

Z implicitnim računanjem in uporabo LU razcepa matrike T_k se izognemo temu, da je potrebno shraniti vse vektorje matrik V_k in W_k in pridemo do kratke rekurzije.

$r_0 = b - Ax_0, \quad p_1 = r_0$

izberi r_0^*, da je $r_0^T r_0^* \neq 0, \quad p_1^* = r_0^*$

$j = 1, 2, \ldots, k$

$$
\begin{aligned}
\alpha_j &= \frac{(r_{j-1}, r_{j-1}^*)}{(Ap_j, p_j^*)}\\
x_j &= x_{j-1} + \alpha_j p_j\\
r_j &= r_{j-1} - \alpha_j A p_j\\
r_j^* &= r_{j-1}^* - \alpha_j A^T p_j^*\\
\beta_j &= \frac{(r_j, r_j^*)}{(r_{j-1}, r_{j-1}^*)}\\
p_{j+1} &= r_j + \beta_j p_j\\
p_{j+1}^* &= r_j^* + \beta_j p_j^*
\end{aligned}
$$

Standarni CG za s.p.d. sistem

$r_0 = b - Ax_0, \quad p_1 = r_0$

$j = 1, 2, \ldots, k$

$$
\begin{aligned}
\alpha_j &= \frac{(r_{j-1}, r_{j-1})}{(Ap_j, p_j)}\\
x_j &= x_{j-1} + \alpha_j p_j\\
r_j &= r_{j-1} - \alpha_j A p_j\\
\beta_j &= \frac{(r_j, r_j)}{(r_{j-1}, r_{j-1})}\\
p_{j+1} &= r_j + \beta_j p_j
\end{aligned}
$$

Lastnosti Bi-CG

$$
\begin{aligned}
&r_0 = b - Ax_0, \quad p_1 = r_0\\
&\text{izberi } r_0^*\text{, da je } r_0^T r_0^* \neq 0, \quad p_1^* = r_0^*\\
&j = 1, 2, \ldots, k\\
&\qquad \alpha_j = \frac{(r_{j-1}, r_{j-1}^*)}{(Ap_j, p_j^*)}\\
&\qquad x_j = x_{j-1} + \alpha_j p_j\\
&\qquad r_j = r_{j-1} - \alpha_j A p_j\\
&\qquad r_j^* = r_{j-1}^* - \alpha_j A^T p_j^*\\
&\qquad \beta_j = \frac{(r_j, r_j^*)}{(r_{j-1}, r_{j-1}^*)}\\
&\qquad p_{j+1} = r_j + \beta_j p_j\\
&\qquad p_{j+1}^* = r_j^* + \beta_j p_j^*
\end{aligned}
$$

Metoda Bi-CG se za razliko od CG lahko zalomi. Poleg težav, ki se pojavijo pri Bi-Lanczosu, se lahko zgodi še *sekundarni zalom*, kadar ne obstaja LU razcep matrike T_k.

Izrek 2. *Če se metoda Bi-CG ne zalomi, potem vektorji zadoščajo*

$$(r_j, r_i^*) = 0, \qquad (Ap_j, p_i^*) = 0 \qquad \text{za } i \neq j.$$

4.3 QMR [Freund, Nachtigal (1991)]

Podobno, kot je GMRES povezana s FOM, je QMR povezana z BiCG.

Bi-Lanczos vrne $AV_k = V_kT_k + \delta_k v_{k+1} e_k^T = V_{k+1}T_{k+1,k}$. Za razliko od simetrične matrike sedaj stolpci matrike V_{k+1} niso ortonormirani.

Iščemo vektor oblike $x_k = x_0 + V_k y_k$, za katerega bi bila norma ostanka

$$\|r_k\| = \|b - Ax_k\| = \|r_0 - V_k y_k\| = \|V_{k+1}(\|r_0\|e_1 - T_{k+1,k}y_k)\|$$

minimalna. Kljub temu, da stolpci V_{k+1} niso ortonormirani, kot približek vzamemo y_k, ki minimizira *kvazi-ostanek*

$$\|\|r_0\|e_1 - T_{k+1,k}y_k\|.$$

Tako dobimo metodo *minimalnega kvazi-ostanka* (QMR).

Če označimo $\|r_k^Q\| = \min(\|\|r_0\|e_1 - T_{k+1,k}y_k\|)$, potem za ostanek $r_k^{QMR} = b - Ax_k$ velja

$$\|r_k^{QMR}\| \le \|V_{k+1}\|_F \|r_k^Q\|.$$

Metodo QMR izvedemo tako, da preko Givensovih rotacij delamo QR razcep matrike T_k. Tako spet lahko delamo kratke rekurzije in posodabljamo x_k.

Primerjava QMR z drugimi metodami

Med QMR in BiCG veljajo podobne zveze kot med GMRES in FOM. Označimo

$$\rho_k^Q = \|r_k^Q\|, \quad \rho_k^{QMR} = \|r_k^{QMR}\|, \quad \rho_k^{BiCG} = \|r_k^{BiCG}\| \quad \text{in} \quad \rho_k^{GMRES} = \|r_k^{GMRES}\|.$$

Za kvazi-ostanke po metodi QMR velja $\rho_k^Q = |s_k|\rho_{k-1}^Q$.

Velja $\rho_k^Q = |c_k|\rho_k^{BiCG}$, kjer je c_k koeficient iz zadnje Givensove rotacije.

Dobimo še $\rho_k^{BiCG} = \dfrac{\rho_k^Q}{\sqrt{1 - \left(\frac{\rho_k^Q}{\rho_{k-1}^Q}\right)^2}}$, od tod pa

$$\frac{1}{(\rho_k^{BiCG})^2} + \frac{1}{(\rho_{k-1}^Q)^2} = \frac{1}{(\rho_k^Q)^2}.$$

Med QMR in GMRES velja ocena

$$\rho_k^Q \leq \kappa_2(V_{k+1})\rho_k^{GMRES}.$$

Vemo še $\rho_k^{QMR} \leq \|V_{k+1}\|_F \rho_k^Q$.

4.4 Metode brez A^T

Obstajajo številni primeri, ko imamo opravka z nesimetričnimi matrikami in množenje z matriko A^T sploh ni na voljo ali pa ni tako ekonomično kot množenje z matriko A.

Zgled 1. *Denimo, da rešujemo nelinearni sistem $F(x) = 0$ z Newtonovo metodo.*

V vsakem koraku moramo rešiti sistem z Jacobijevo matriko $JF(x)$. Če to rešujemo preko metod podprostorov Krilova, lahko za poljuben vektor v produkt $JF(x)v$ dobro aproksimiramo z

$$JF(x)v = \frac{F(x + \epsilon v) - F(x)}{\epsilon}$$

za primerno majhen ϵ.

Podobna ekonomična formula za izračun produkta $JF(x)^T w$ ne obstaja.

CGS [Sonneveld (1984)]

Gre za izpeljanko BiCG, kjer ne uporabljamo A^T.

V BiCG za ostanek r_k velja $r_k = \phi_k(A)r_0$, kjer je ϕ_k polinom stopnje k, ki zadošča $\phi_k(0) = 1$. Podobno velja $p_k = \pi_{k-1}(A)r_0$, kjer je π_{k-1} polinom stopnje $k-1$.

Oba polinoma nastopata tudi pri r_k^* in p_k^*, saj je $r_k^* = \phi_k(A^T)r_0^*$ in $p_k^* = \pi_{k-1}(A^T)r_0^*$.

Pri BiCG računamo $r_j = r_{j-1} - \alpha_j A p_j$ in $p_{j+1} = r_j + \beta_j p_j$, kjer za skalarja velja

$$\alpha_j = \frac{(r_{j-1}, r_{j-1}^*)}{(Ap_j, p_j^*)} \qquad \text{in} \qquad \beta_j = \frac{(r_j, r_j^*)}{(r_{j-1}, r_{j-1}^*)}.$$

Dobimo

$$\alpha_j = \frac{(r_{j-1}, r_{j-1}^*)}{(Ap_j, p_j^*)} = \frac{(\phi_j(A)r_0, \phi_j(A^T)r_0^*)}{(A\pi_{j-1}(A)r_0, \pi_{j-1}(A^T)r_0^*)} = \frac{(\phi_j^2(A)r_0, r_0^*)}{(A\pi_{j-1}^2(A)r_0, r_0^*)}.$$

$$\beta_j = \frac{(r_j, r_j^*)}{(r_{j-1}, r_{j-1}^*)} = \frac{(\phi_j(A)r_0, \phi_j(A^T)r_0^*)}{(\phi_{j-1}(A)r_0, \phi_{j-1}(A^T)r_0^*)} = \frac{(\phi_j^2(A)r_0, r_0^*)}{(\phi_{j-1}^2(A)r_0, r_0^*)}.$$

Če izpeljemo rekurzivne zveze za vektorje $\phi_j^2(A)r_0$ in $\pi_{j-1}^2(A)r_0$, potem matrike A^T ne potrebujemo.

CGS

Če definiramo $r_j = \phi_j^2(A)r_0$, $p_j = \pi_{j-1}^2(A)r_0$, $q_j = \phi_j(A)\pi_{j-1}(A)r_0$ in pomožni vektor $u_j = r_{j-1} + \beta_{j-1}q_{j-1}$, lahko izpeljemo rekurzivne zveze

$$\begin{aligned} r_j &= r_{j-1} - \alpha_j A(u_j + q_j) \\ p_j &= u_j + \beta_{j-1}(q_{j-1} + \beta_{j-1}p_{j-1}) \\ q_j &= u_j - \alpha_j A p_j. \end{aligned}$$

Tako pridemo do metode *kvadriranih konjugiranih gradientov* (CGS)

$r_0 = b - Ax_0, \quad u_1 = p_1 = r_0$

izberi r_0^*, da je $r_0^T r_0^* \neq 0$

$j = 1, 2, \ldots, k$

$$\alpha_j = \frac{(r_{j-1}, r_0^*)}{(Ap_j, r_0^*)}$$

$$q_j = u_j - \alpha_j A p_j$$

$$x_j = x_{j-1} + \alpha_j(u_j + q_j)$$

$$r_j = r_{j-1} - \alpha_j A(u_j + q_j)$$

$$\beta_j = \frac{(r_j, r_0^*)}{(r_{j-1}, r_0^*)}$$

$$u_{j+1} = r_j + \beta_j q_j$$

$$p_{j+1} = u_{j+1} + \beta_j(q_j + \beta_j p_j)$$