

Algoritmi za konstrukcijo baze za $\mathcal{K}_k(A, b)$

Lanczos(1950), matrika A je simetrična

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

stolpci V_k so ortonormirani, T_k je tridiagonalna

Arnoldi(1951)

$$AV_k = V_k H_k + h_{k+1,k} v_{k+1} e_k^T$$

stolpci V_k so ortonormirani, H_k je zgornja Hessenbergova
varianata RGS:

Lanczos(1952)

$$\begin{aligned} AV_k &= V_k T_k + \delta_k v_{k+1} e_k^T \\ A^T W_k &= W_k T_k^T + \beta_k w_{k+1} e_k^T \end{aligned}$$

stolpci V_k in W_k so biortogonalni, T_k je tridiagonalna.

Iščemo rešitev sistema $Ax = b$.

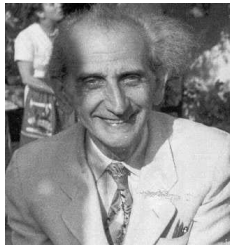
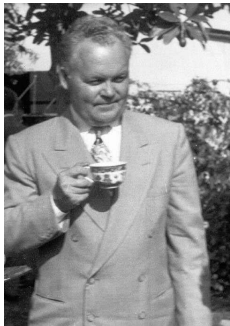
Splošni nastavek je, da iščemo $x_k \in x_0 + \mathcal{K}_k(A, r_0)$, da je $r_k \perp \mathcal{L}$. Možnosti:

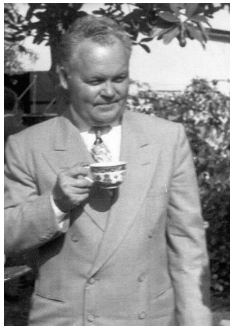
- 1 $\mathcal{L} = \mathcal{K}_k(A, r_0)$: to vodi do metod tipa FOM in CG
- 2 $\mathcal{L} = A\mathcal{K}_k(A, r_0)$: to vodi do metod tipa GMRES
- 3 $\mathcal{L} = \mathcal{K}_k(A^T, \tilde{r}_0)$: to vodi do metod tipa BiCG

Lema

Rešitev, ki ustreza pogoju $r_k \perp A\mathcal{K}_k(A, r_0)$, minimizira ostanek $\|b - Ax_k\|_2$.

Konjugirani gradienti

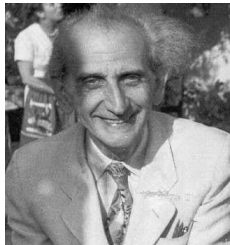




Magnus Hestenes

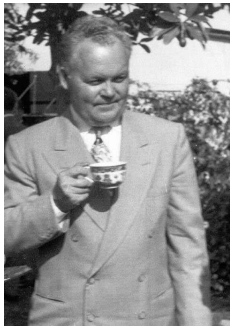


Eduard Stiefel



Cornelius Lanczos

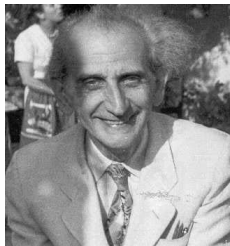
Konjugirani gradienti



Magnus Hestenes



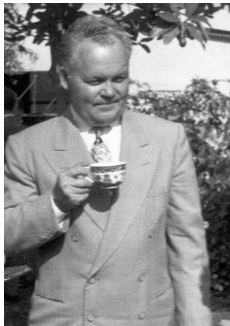
Eduard Stiefel



Cornelius Lanczos

Hestenes in Stiefel sta leta 1951 na konferenci ugotovila, da sta neodvisno okrila novo metodo - konjugirane gradiente. V kasnejšem skupnem članku (1952) sta navedla, da je metoda sorodna metodi, ki jo je objavil Lanczos leta 1952.

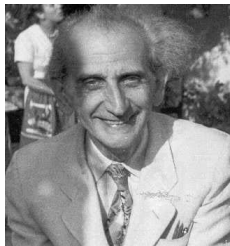
Konjugirani gradienti



Magnus Hestenes



Eduard Stiefel

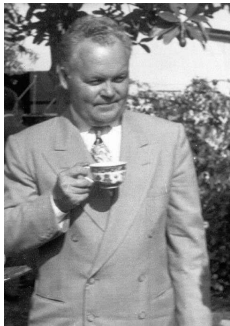


Cornelius Lanczos

Hestenes in Stiefel sta leta 1951 na konferenci ugotovila, da sta neodvisno okrila novo metodo - konjugirane gradiente. V kasnejšem skupnem članku (1952) sta navedla, da je metoda sorodna metodi, ki jo je objavil Lanczos leta 1952.

Metoda je bila od začetka mišljena kot direktna metoda za reševanje linearnih sistemov, ki pa ima lepo lastnost, da včasih že z manj dela vrne dober rezultat.

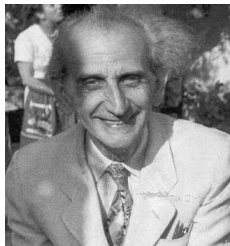
Konjugirani gradienti



Magnus Hestenes



Eduard Stiefel



Cornelius Lanczos

Hestenes in Stiefel sta leta 1951 na konferenci ugotovila, da sta neodvisno okrila novo metodo - konjugirane gradiente. V kasnejšem skupnem članku (1952) sta navedla, da je metoda sorodna metodi, ki jo je objavil Lanczos leta 1952.

Metoda je bila od začetka mišljena kot direktna metoda za reševanje linearnih sistemov, ki pa ima lepo lastnost, da včasih že z manj dela vrne dober rezultat.

Hestenes se je s konjugiranimi bazami ukvarjal že leta 1936, pa je profesor na Harvardu menil, da je to preveč očitno in nepriljubeno za publikacijo...

Situacija 1952

$$\mathcal{L} = \mathcal{K}_k(A, r_o)$$

Lanczos

CG

$$\mathcal{L} = A\mathcal{K}_k(A, r_o)$$

$$\mathcal{L} = \mathcal{K}_k(A^T, \tilde{r}_o)$$

BiLanczos

Naslednjih 20 let metode podprostorov Krilova niso prišle do veljave:

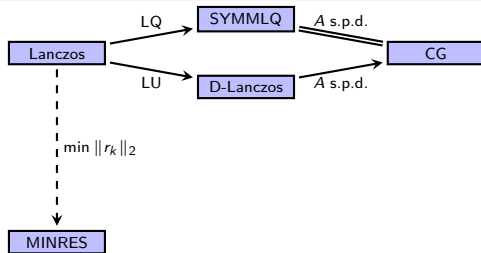
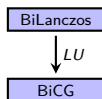
- Arnoldijev algoritem je bil najprej zamišljen za pretvorbo matrike v zgornjo Hessenbergovo matriko in računanje lastnih vrednosti. Kmalu zatem so pojavile Givensove rotacije in Householderjeva zrcaljenja (1958), s katerimi to lahko naredimo učinkoviteje.
- Leta 1971 je John Reid ugotovil, da so konjugirani gradienti zelo primerna metoda za reševanje velikih razpršenih matrik in razvoj se je spet začel.

Situacija 1975 - Fletcher (1974), Paige, Saunders (1975)

$$\mathcal{L} = \mathcal{K}_k(A, r_o)$$

$$\mathcal{L} = A\mathcal{K}_k(A, r_o)$$

$$\mathcal{L} = \mathcal{K}_k(A^T, \tilde{r}_o)$$

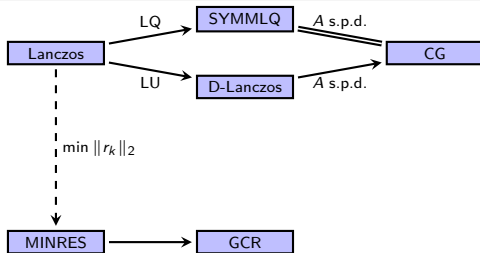


Situacija 1984 - Jea, Young (1980), Sonneveld (1984)

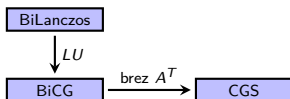
$$\mathcal{L} = \mathcal{K}_k(A, r_o)$$

$$\mathcal{L} = A\mathcal{K}_k(A, r_o)$$

$$\mathcal{L} = \mathcal{K}_k(A^T, \tilde{r}_o)$$



ORTHODIR

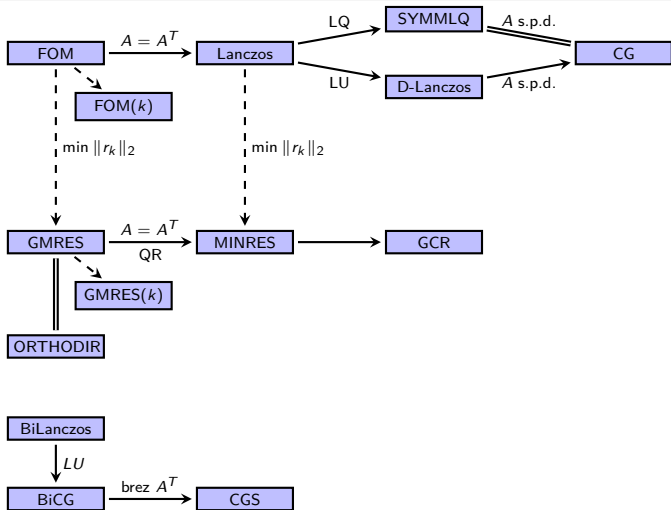


Situacija 1986: Saad, Schultz (1986)

$$\mathcal{L} = \mathcal{K}_k(A, r_o)$$

$$\mathcal{L} = A\mathcal{K}_k(A, r_o)$$

$$\mathcal{L} = \mathcal{K}_k(A^T, \tilde{r}_o)$$

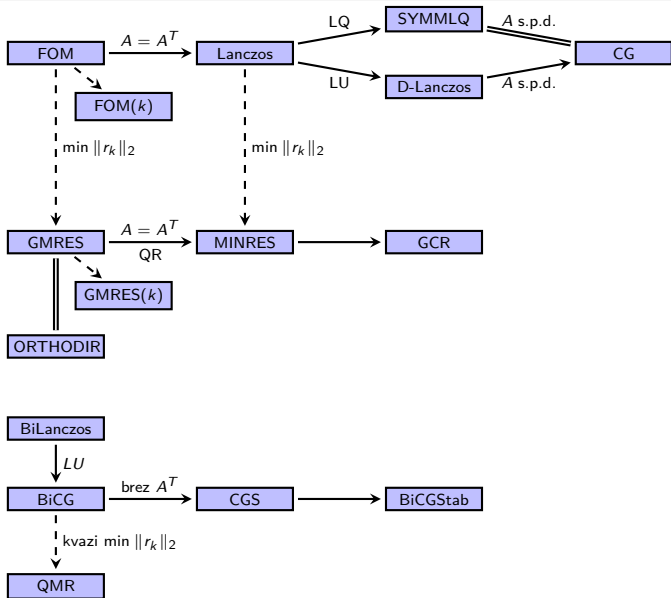


Situacija 1992: Freund, Nachtigal (1991), van der Vorst (1992)

$$\mathcal{L} = \mathcal{K}_k(A, r_o)$$

$$\mathcal{L} = \mathcal{AK}_k(A, r_o)$$

$$\mathcal{L} = \mathcal{K}_k(A^T, \tilde{r}_o)$$

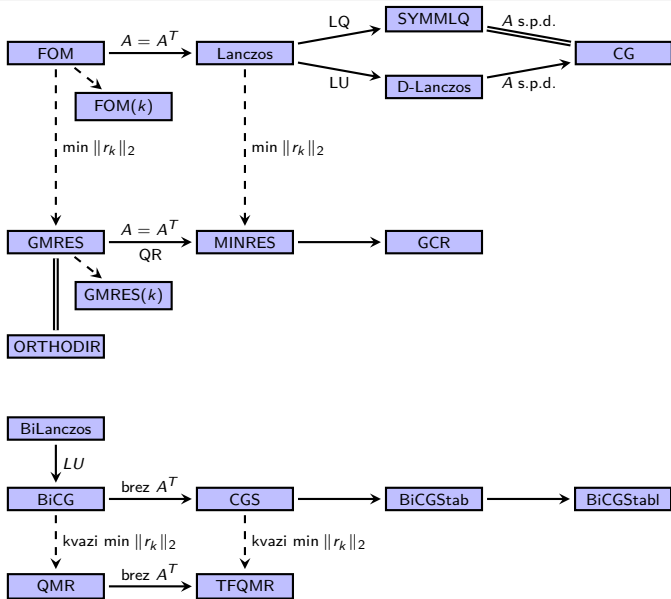


Situacija 1993: Freund (1993), Sleijpen, Fokkema (1993)

$$\mathcal{L} = \mathcal{K}_k(A, r_o)$$

$$\mathcal{L} = A\mathcal{K}_k(A, r_o)$$

$$\mathcal{L} = \mathcal{K}_k(A^T, \tilde{r}_o)$$



CGLS

Če rešujemo $Ax = b$ in je A nesimetrična, lahko uporabimo CG na normalnem sistemu

$$A^T Ax = A^T b.$$

Prednost: izognemo se dolgim rekurzijam, ki jih potrebujeta FOM in GMRES.

Slabost: občutljivost matrike $A^T A$ je kvadrat občutljivosti matrike A .

To deluje v redu, če so singularne vrednosti matrike A lepo razporejene po gručah oziroma kadar je matrika A blizu ortogonalne matrike.

Ustrezna metoda brez eksplicitnega računanja $A^T A$ je CGLS (Paige in Saunders 1982).

Izrek 1. Metoda CGLS vrne x_j , ki minimizira $\|b - Ax_j\|_2$ po afinem podprostoru $x_0 + \mathcal{K}_j(A^T A, A^T r_0)$.

LSQR

Normalni sistem lahko uporabimo tudi za reševanje predoločenega sistema $Ax = b$ po m.n.k.. Še boljšo metodo pa dobimo, če upoštevamo ekvivalenten *razširjeni sistem*

$$\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}.$$

Če začnemo s približkom $\begin{bmatrix} 0 \\ 0 \end{bmatrix}$, je prvi vektor v ON bazi za podp. Krilova $w_1 = \frac{1}{\|b\|} \begin{bmatrix} b \\ 0 \end{bmatrix}$.

Naslednji vektor je $w_2 = \frac{1}{\|A^T b\|} \begin{bmatrix} 0 \\ A^T b \end{bmatrix}$.

Po vrsti se izmenjujejo vektorji oblike $\begin{bmatrix} u \\ 0 \end{bmatrix}$ in $\begin{bmatrix} 0 \\ v \end{bmatrix}$.

Če sestavimo matriki U_k in V_k iz prvih k stolpcev u in v , velja

$$\beta_1 u_1 = b$$

$$\alpha_1 v_1 = A^T u_1$$

$$i = 1, 2, \dots$$

$$\beta_{i+1} u_{i+1} = Av_i - \alpha_i u_i$$

$$\alpha_{i+1} v_{i+1} = A^T u_{i+1} - \beta_{i+1} v_i$$

Skalarji α_i, β_i so izbrani tako, da so vektorji u_i in v_i normirani.

Delna bidiagonalizacija

$$\beta_1 u_1 = b, \quad \alpha_1 v_1 = A^T u_1$$

$$i = 1, 2, \dots$$

$$\beta_{i+1} u_{i+1} = Av_i - \alpha_i u_i, \quad \alpha_{i+1} v_{i+1} = A^T u_{i+1} - \beta_{i+1} v_i$$

Če sestavimo matriki U_k in V_k iz prvih k stolpcev u in v , velja

$$\beta_1 U_{k+1} e_1 = b$$

$$AV_k = U_{k+1} B_k$$

$$A^T U_{k+1} = V_k B_k^T + \alpha_{k+1} v_{k+1} e_{k+1}^T$$

za

$$B_k = \begin{bmatrix} \alpha_1 & & & & \\ \beta_2 & \alpha_2 & & & \\ & \beta_3 & \ddots & & \\ & & \ddots & \alpha_k & \\ & & & \beta_{k+1} & \end{bmatrix}.$$

Povezava z Lanczosevo metodo za $A^T Ax = A^T b$:

Izrek 2. Velja $T_k = B_k^T B_k$, kjer je T_k tridiagonalna matrika iz Lanczoseve metode za matriko $A^T A$ in začetni vektor $A^T b$. Velja še

$$A^T AV_k = V_k T_k + \alpha_{k+1} \beta_{k+1} v_{k+1} e_{k+1}^T.$$

Rešitev LSQR

Iščemo $\begin{bmatrix} r_k \\ x_k \end{bmatrix}$, da bo ostanek

$$\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} r_k \\ x_k \end{bmatrix} - \begin{bmatrix} b \\ 0 \end{bmatrix}$$

pravokoten na podprostor Krilova.

Iz $A^T A V_k = V_k T_k + \alpha_{k+1} \beta_{k+1} v_{k+1} e_{k+1}^T$ sledi, da iščemo približek oblike $x_k = V_k y_k$.

Ostanek r_k je potem oblike $r_k = U_{k+1} t_{k+1}$.

Vzeti moramo podprostor Krilova, ki ga razpenjajo stolpci matrike $\begin{bmatrix} U_{k+1} & 0 \\ 0 & V_k \end{bmatrix}$.

Iz $\begin{bmatrix} U_{k+1}^T & 0 \\ 0 & V_k^T \end{bmatrix} \left(\begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} U_{k+1} t_{k+1} \\ V_k y_k \end{bmatrix} - \begin{bmatrix} b \\ 0 \end{bmatrix} \right) = 0$ dobimo

$$\begin{bmatrix} I & B_k \\ B_k^T & 0 \end{bmatrix} \begin{bmatrix} t_{k+1} \\ y_k \end{bmatrix} = \begin{bmatrix} \beta_1 e_1 \\ 0 \end{bmatrix},$$

kar je ravno predoločen sistem $\min \|B_k y_k - \beta_1 e_1\|$.

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

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

Dobimo

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

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

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

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 $r_0^T r_0^* \neq 0$

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

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

$$q_j = u_j - \alpha_j Ap_j$$

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

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

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

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

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

Bi-CGSTAB [van der Vorst (1992)]

Konvergenca pri metodi CGS je lahko dokaj nepravilna. Zaradi tega lahko pride do velikih zaokrožitvenih napak, še posebno zaradi samega vpliva kvadriranja na ostanek.

Za razliko od CGS pri metodi *stabiliziranih bikonjugiranih gradientov* metodo BiCG posplošimo tako, da bo konvergenca bolj gladka, še vedno pa računanje z matriko A^T ne bo potrebno.

Pri metodi CGS se računanju z matriko A^T izognemo tako, da vpeljemo vektorje

$$r'_j = \phi_j^2(A)r_0.$$

Pri metodi Bi-CGSTAB pa dobimo ostanke oblike

$$r'_j = \psi_j(A)\phi(A)r_0,$$

kjer je polinom $\phi_j(A)$ določen z metodo BiCG, polinom $\psi_j(A)$ pa je polinom, ki poskrbi za gladko konvergenco oz. za stabilizacijo konvergence. Definiramo ga rekurzivno kot

$$\psi_j(t) = (1 - \omega_j t)\psi_{j-1}(t),$$

kjer v vsakem koraku določimo konstanto ω_j .

Bi-CGSTAB kot kombinirana metoda

Metoda Bi-CGSTAB spada med kombinirane metode.

Na začetku predavanj smo spoznali t.i. posodobitev Richardsonove iteracije, kjer vpeljemo skalarje ω_i in iteriramo

$$x_k = x_{k-1} + \omega_{k-1}r_{k-1}.$$

Tako dobimo

$$r_k = (I - \omega_{k-1}A)r_{k-1}$$

oziroma $r_k = \psi_k(A)r_0$.

Pri Bi-CGSTAB kombiniramo zgornjo iteracijo z metodo BiCG.

Ozadje Bi-CGSTAB

Pri metodi BiCG mora veljati

$$\langle \phi_i(A)r_0, \phi_j(A^T)r_0^* \rangle = 0$$

za $j < i$, saj to pomeni, da je ostanek $r_i = \phi_i(A)r_0$ pravokoten na $\mathcal{K}_i(A^T, r_0^*)$.

Ekvivalentno bi bilo zahtevati da je r_i pravokoten na vektorje $\psi_j(A^T)r_0^*$ za take polinome, da je ψ_j stopnje j za $j = 0, \dots, i - 1$.

To pa pomeni

$$\langle \phi_i(A)r_0, \psi_j(A^T)r_0^* \rangle = \langle \psi_j(A)\phi_i(A)r_0, r_0^* \rangle = 0.$$

Pri metodi CGS vzamemo kar $\psi_j = \phi_j$, pri Bi-CGSTAB pa vzamemo polinome oblike

$$\psi_j(t) = (1 - \omega_1 t) \cdots (1 - \omega_j t).$$

V vsakem koraku imamo na voljo še prosti parameter ω_j . Izberemo ga tako, da minimiziramo $\|r_j\|$.

Algoritem Bi-CGSTAB

Če definiramo $r_j = \psi_j(A)\phi_j(A)r_0$, $p_j = \psi_{j-1}(A)\pi_{j-1}(A)r_0$, $q_j = \phi_j(A)\pi_{j-1}(A)r_0$, lahko izpeljemo rekurzivne zveze

$$\begin{aligned}r_j &= (I - \omega_j A)(r_{j-1} - \alpha_j A p_j) \\ p_j &= r_{j-1} + \beta_{j-1}(I - \omega_{j-1} A)p_{j-1}.\end{aligned}$$

Ko ustrezno preračunamo še koeficiente α_j , β_j in ω_j , pridemo do metode Bi-CGSTAB.

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

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

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

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

$$s_j = r_{j-1} - \alpha_j A p_j$$

$$\omega_j = \frac{(A s_j, s_j)}{(A s_j, A s_j)}$$

$$x_j = x_{j-1} + \alpha_j p_j + \omega_j s_j$$

$$r_j = s_j - \omega_j A s_j$$

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

$$p_{j+1} = r_j + \beta_j(p_j - \omega_j A p_j)$$

Kaj še obstaja

Obstaja še mnogo algoritmov, npr:

- *TFQMR* Transpose-free QMR. Ta metoda je posplošitev QMR z enakim pristopom, kot ga uporabimo pri CGS. Tako se znebimo računanja z A^T , obenem pa v vsakem koraku minimiziramo kvazi-residual.
- *Bi-CGSTAB(l)*. Mi smo v bistvu spoznali Bi-CGSTAB(1), za katerega se da pokazati, da je kombinacija GMRES(1) in BiCG. Problem z GMRES(1) je, da ima polinom ψ_j po konstrukciji za realno matriko same realne ničle, tak polinom pa ne more dobro uničiti lastnih vrednosti matrike A , ki so lahko tudi kompleksne.
Pri Bi-CGSTAB(l) polinom $\psi_j(A)$ zgeneriramo na drugačen način in dobimo kombinacijo GMRES(l) in BiCG.

Kratka ponovitev lastnosti algoritmov

Metoda	MV	skal. prod.	SAXPY	Pomnilnik
FOM	1	$k + 1$	$k + 1$	$(k + 4)n$
D-Lanczos	1	2	4	$5n$
CG	1	2	3	$4n$
GMRES	1	$k + 1$	$k + 1$	$(k + 5)n$
BiCG	1/1	2	5	$6n$
QMR	1/1	2	8	$10n$
CGS	2	2	6	$9n$
Bi-CGSTAB	2	4	6	$8n$

FOM in GMRES:

- Za nesimetrične matrike.
- GMRES vrne rešitev z najmanjšo normo ostanka, FOM pa rešitev s pravokotnim ostankom.
- Če GMRES naredi v enem koraku bistveno izboljšavo, potem FOM vrne približno enako dober rezultat. Sicer pa FOM lahko nekaj korakov stagnira, GMRES pa vedno pridobiva.
- Če tudi FOM rešujemo preko QR, se metodi ne moreta zalomiti.
- Prostorska in časovna zahtevnost koraka linearno narašča, zato uporabljamo ponovni zagon.

Kratka ponovitev lastnosti algoritmov

Metoda	MV	skal. prod.	SAXPY	Pomnilnik
FOM	1	$k + 1$	$k + 1$	$(k + 4)n$
D-Lanczos	1	2	4	$5n$
CG	1	2	3	$4n$
GMRES	1	$k + 1$	$k + 1$	$(k + 5)n$
BiCG	1/1	2	5	$6n$
QMR	1/1	2	8	$10n$
CGS	2	2	6	$9n$
Bi-CGSTAB	2	4	6	$8n$

D-Lanczos:

- Za simetrične matrike.
- Teoretično vrne isti rezultat kot FOM.
- Lahko se zalomi, če LU razcep matrike T_k brez pivotiranja ne obstaja.

CG:

- Za simetrične pozitivno definitne matrike.
- Teoretično vrne isti rezultat kot FOM.
- Superlinearna konvergenca, če so ekstremne lastne vrednosti lepo separirane.

Kratka ponovitev lastnosti algoritmov

Metoda	MV	skal. prod.	SAXPY	Pomnilnik
FOM	1	$k + 1$	$k + 1$	$(k + 4)n$
D-Lanczos	1	2	4	$5n$
CG	1	2	3	$4n$
GMRES	1	$k + 1$	$k + 1$	$(k + 5)n$
BiCG	1/1	2	5	$6n$
QMR	1/1	2	8	$10n$
CGS	2	2	6	$9n$
Bi-CGSTAB	2	4	6	$8n$

BiCG:

- Za nesimetrične matrice, potrebuje množenje z A in A^T .
- Zalomi se, če se zalomi Lanc. biortogonalizacija ali ne obstaja LU razcep matrice T_k .
- Če je A simetrična, se ujema z D-Lanczosem.

QMR:

- Za nesimetrične matrice, potrebuje množenje z A in A^T , minimizira kvazi-residual.
- Bolj gladka konvergenca kot BiCG.
- Če je A s.p.d., se ujema s CG.
- Zalomi se, če se zalomi Lanczoseva biortogonalizacija.

Kratka ponovitev lastnosti algoritmov

Metoda	MV	skal. prod.	SAXPY	Pomnilnik
FOM	1	$k + 1$	$k + 1$	$(k + 4)n$
D-Lanczos	1	2	4	$5n$
CG	1	2	3	$4n$
GMRES	1	$k + 1$	$k + 1$	$(k + 5)n$
BiCG	1/1	2	5	$6n$
QMR	1/1	2	8	$10n$
CGS	2	2	6	$9n$
Bi-CGSTAB	2	4	6	$8n$

CCS:

- Za nesimetrične matrice, potrebuje samo množenje z A .
- Običajno zelo nepravilna konvergenca.
- Zaradi kvadratov so lahko prisotne velike zaokrožitvene napake.

Bi-CGSTAB:

- Za nesimetrične matrice, potrebuje samo množenje z A .
- Običajno bolj gladka konvergenca kot CGS in manjše napake.

Katero metodo izbrati?

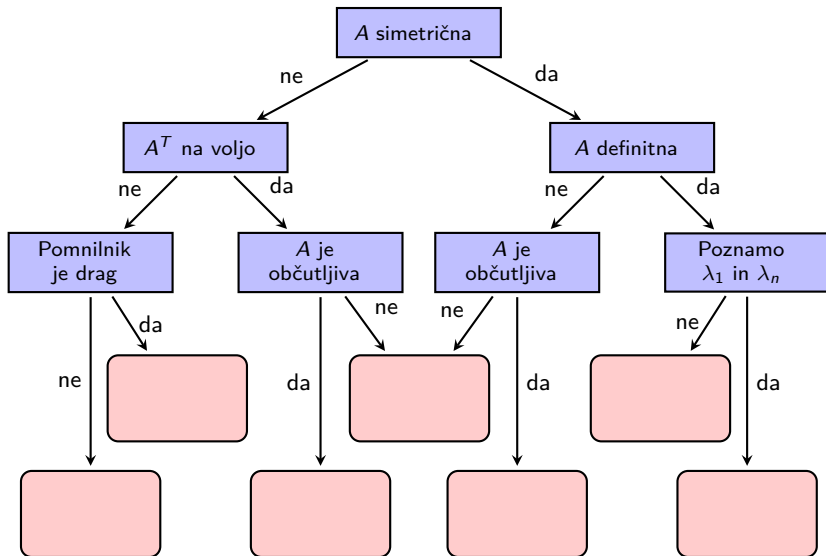
Točnega pravila ni. Za razliko od direktnih metod se pri iterativnih metodah ne da dobiti splošne robustne metode, ki bo dobra za vse primere.

Ustrezna metoda je odvisna od

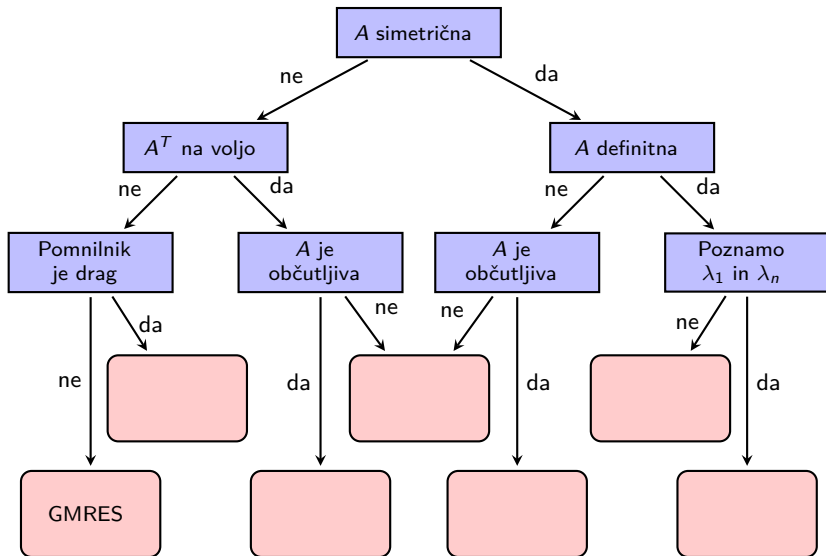
- lastnosti matrike A ,
- zahtevnosti množenja z matriko A (in A^T , če je na voljo),
- pomnilnika, ki ga imamo na voljo,
- poznavanja spektra matrike A ,
- *možnosti dobrega predpogojevanja*.

Nachtigal, Reddy in Trefethen so leta 1992 pokazali, da za vsako izmed metod tipa CG, BiCG, GMRES, CGS, CGLS obstaja razred linearnih sistemov, za katero je izbrana metoda najboljša. To pomeni, da najboljša metoda ne obstaja.

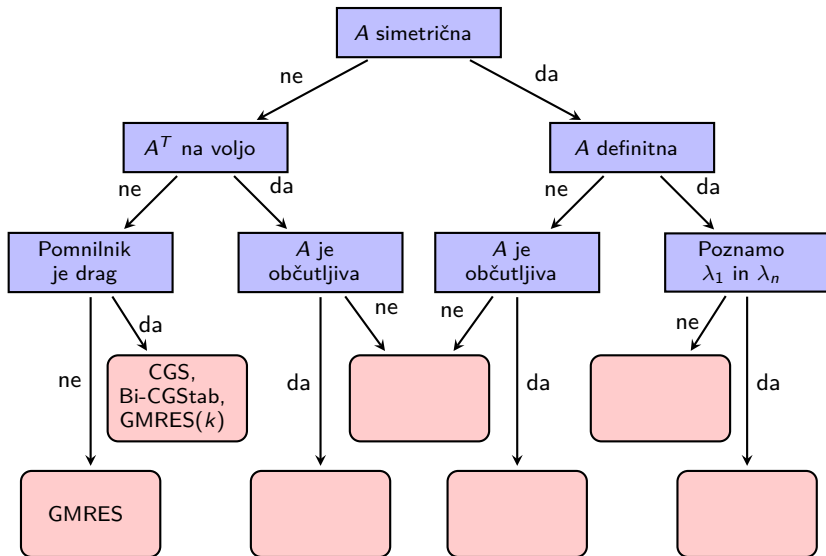
Odločitveno drevo (Demmel)



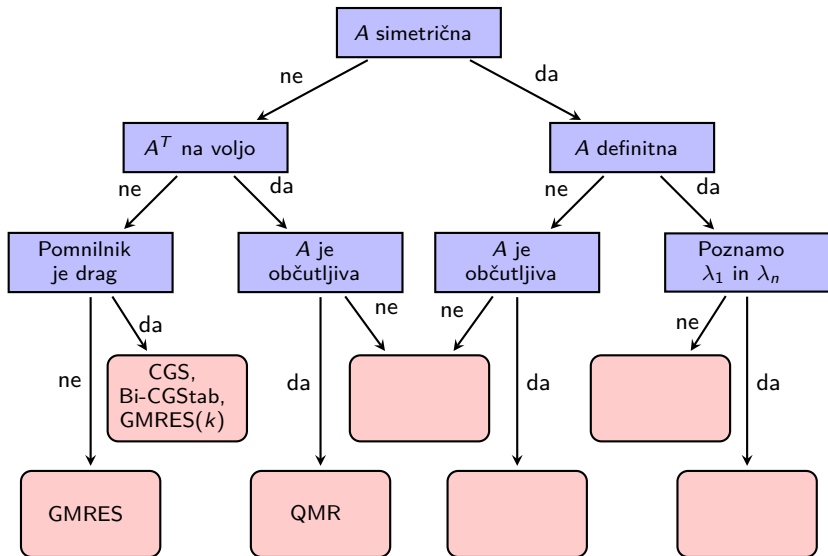
Odločitveno drevo (Demmel)



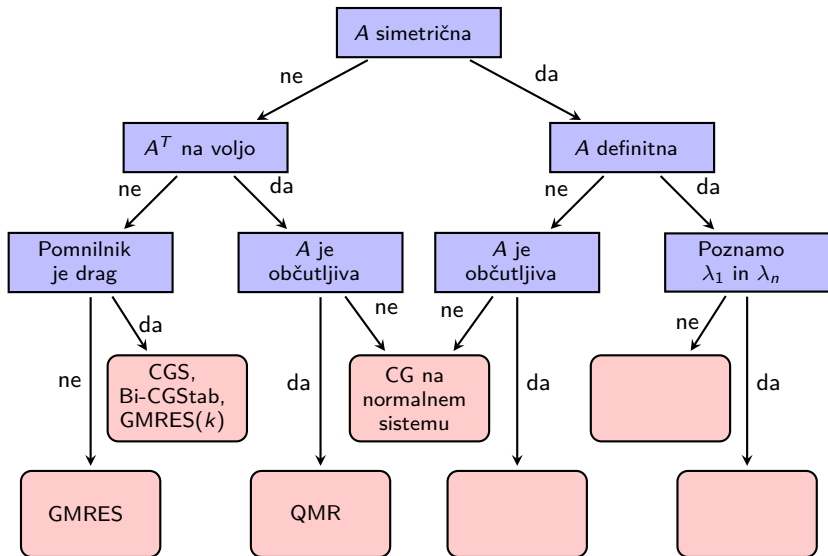
Odločitveno drevo (Demmel)



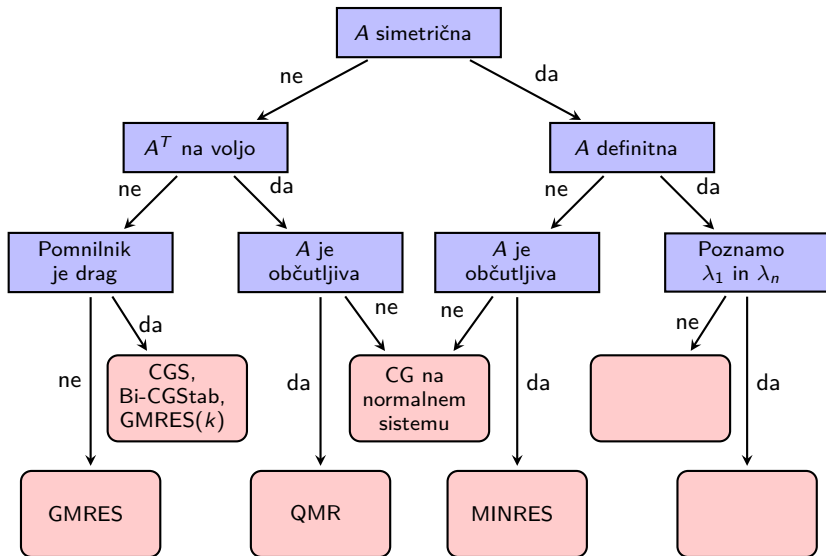
Odločitveno drevo (Demmel)



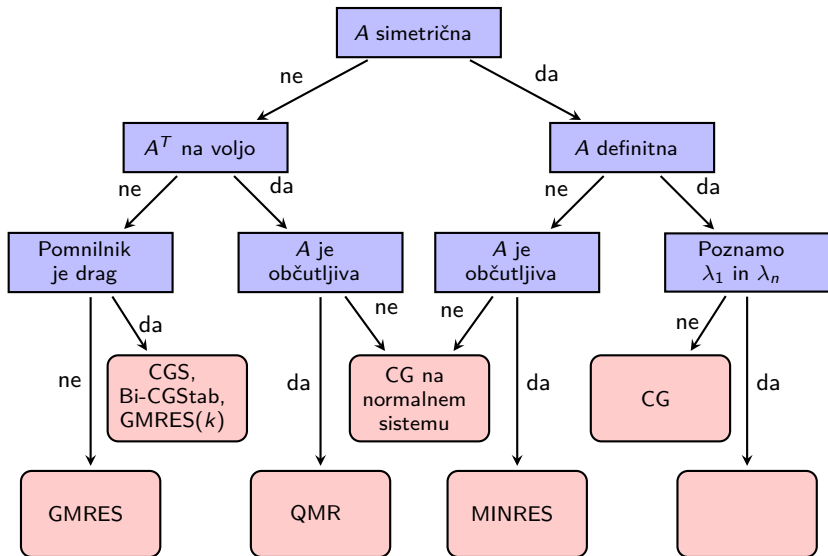
Odločitveno drevo (Demmel)



Odločitveno drevo (Demmel)



Odločitveno drevo (Demmel)



Odločitveno drevo (Demmel)

