Deadbeat Control Solution of Large Linear Systems

Fabio Marcuzzi [1] Caterina Calgaro [2]

Giornate di Algebra Lineare Numerica
Padova, 26 Febbraio 2007

[1] Dipartimento di Matematica Pura ed Applicata, Università di Padova
[2] Laboratoire Paul Painlevé, Université des Sciences et Technologies de Lille

Sia dato il seguente sistema lineare a matrice sparsa:

$$K_h \bar{x}_h = f_h \tag{1}$$

dove la matrice $K_h \in \Re^{n \times n}$ ed il vettore $f_h \in \Re^n$ sono ottenuti dalla discretizzazione di un BVP utilizzando ad es. il Metodo degli Elementi Finiti su una griglia di passo h; $\bar{x}_h \in \Re^n$ è la soluzione incognita.

Iterazione di Richardson

- $$x_h^{(i+1)} = x_h^{(i)} + \alpha_i P_h^{-1} r_h^{(i)} \quad , \quad r_h^{(i)} = K_h x_h^{(i)} - f_h \quad , \qquad i = 0, 1, \ldots \tag{2}$$

Iterazione di Richardson

- $$x_h^{(i+1)} = x_h^{(i)} + \alpha_i P_h^{-1} r_h^{(i)} \quad , \quad r_h^{(i)} = K_h x_h^{(i)} - f_h \quad , \qquad i = 0, 1, \ldots \tag{2}$$

- $$\begin{aligned} e_h^{(i+1)} &= e_h^{(i)} - P_h^{-1} r_h^{(i)} \\ r_h^{(i)} &= K_h e_h^{(i)} \end{aligned} \tag{3}$$

Iterazione di Richardson

- $$x_h^{(i+1)} = x_h^{(i)} + \alpha_i P_h^{-1} r_h^{(i)} \quad , \quad r_h^{(i)} = K_h x_h^{(i)} - f_h \quad , \qquad i = 0, 1, \ldots \tag{2}$$

- $$\begin{aligned} e_h^{(i+1)} &= e_h^{(i)} - P_h^{-1} r_h^{(i)} \\ r_h^{(i)} &= K_h e_h^{(i)} \end{aligned} \tag{3}$$

- $$\begin{aligned} e_h^{(i+1)} &= A e_h^{(i)} - B G r_h^{(i)} \\ r_h^{(i)} &= C e_h^{(i)} \end{aligned} \tag{4}$$

$A = I_h \in \Re^{n \times n}$, $B = I_h$, $G = P_h^{-1}$ and $C = K_h$

Controllo a retroazione e Precondizionamento

- Con il *controllo a retroazione dall'uscita* $-BGr_h^{(i)}$, si ha:

$$e_h^{(i+1)} = \Gamma e_h^{(i)} \quad , \quad \Gamma = A - BGC \tag{5}$$

dove Γ è detta *closed-loop matrix* ed i suoi autovalori determinano la dinamica con cui evolve il vettore di stato.

Controllo a retroazione e Precondizionamento

- Con il *controllo a retroazione dall'uscita* $-BGr_h^{(i)}$, si ha:

$$e_h^{(i+1)} = \Gamma e_h^{(i)} \quad , \quad \Gamma = A - BGC \tag{5}$$

  dove Γ è detta *closed-loop matrix* ed i suoi autovalori determinano la dinamica con cui evolve il vettore di stato.
- Come scegliere G in modo che $\rho(\Gamma)$ sia minimo ?

Controllo a retroazione e Precondizionamento

- Con il *controllo a retroazione dall'uscita* $-BGr_h^{(i)}$, si ha:

$$e_h^{(i+1)} = \Gamma e_h^{(i)} \quad , \quad \Gamma = A - BGC \tag{5}$$

  dove Γ è detta *closed-loop matrix* ed i suoi autovalori determinano la dinamica con cui evolve il vettore di stato.
- Come scegliere G in modo che $\rho(\Gamma)$ sia minimo ?
- Un approccio ben noto nella teoria dei sistemi è il cosiddetto *Pole Placement* : scelte le posizioni desiderate per i poli nel piano complesso, si determina la matrice guadagno G tale per cui gli autovalori di Γ coincidano con tali posizioni.

Controllo a retroazione e Precondizionamento

- Con il *controllo a retroazione dall'uscita* $-BGr_h^{(i)}$, si ha:

$$e_h^{(i+1)} = \Gamma e_h^{(i)} \quad , \quad \Gamma = A - BGC \tag{5}$$

  dove Γ è detta *closed-loop matrix* ed i suoi autovalori determinano la dinamica con cui evolve il vettore di stato.
- Come scegliere G in modo che $\rho(\Gamma)$ sia minimo ?
- Un approccio ben noto nella teoria dei sistemi è il cosiddetto *Pole Placement* : scelte le posizioni desiderate per i poli nel piano complesso, si determina la matrice guadagno G tale per cui gli autovalori di Γ coincidano con tali posizioni.
- Notare il confronto tra precondizionamento, che conduce ad una miglior distribuzione degli autovalori della matrice, e pole-placement, che alloca gli autovalori in posizioni prestabilite. Il secondo è un concetto più forte e costoso.

Controllo deadbeat

- Si parla di *controllo deadbeat* quando si vuole che Γ abbia tutti gli autovalori nulli. In tal caso, l'evoluzione libera dello stato del sistema retroazionato va a zero in al più $r < n$ passi, con r pari all'indice di raggiungibilità, cioè al più piccolo intero r per cui

$$rank([B \quad AB \quad A^2B \quad \dots \quad A^{r-1}B]) = n \qquad (6)$$

Controllo deadbeat

- Si parla di *controllo deadbeat* quando si vuole che Γ abbia tutti gli autovalori nulli. In tal caso, l'evoluzione libera dello stato del sistema retroazionato va a zero in al più $r < n$ passi, con r pari all'indice di raggiungibilità, cioè al più piccolo intero r per cui

$$rank([B \quad AB \quad A^2B \quad \dots \quad A^{r-1}B]) = n \tag{6}$$

- Nel nostro caso, con $m = n$ ingressi indipendenti, si ha $r = 1$ e l'errore si annulla in un solo passo, come farebbe l'eliminazione di Gauss. Infatti, con $G = K_h^{-1}$ l'errore si annulla in un passo.

Controllo deadbeat

- Si parla di *controllo deadbeat* quando si vuole che Γ abbia tutti gli autovalori nulli. In tal caso, l'evoluzione libera dello stato del sistema retroazionato va a zero in al più $r < n$ passi, con r pari all'indice di raggiungibilità, cioè al più piccolo intero r per cui

$$rank([B \quad AB \quad A^2B \quad \dots \quad A^{r-1}B]) = n \tag{6}$$

- Nel nostro caso, con $m = n$ ingressi indipendenti, si ha $r = 1$ e l'errore si annulla in un solo passo, come farebbe l'eliminazione di Gauss. Infatti, con $G = K_h^{-1}$ l'errore si annulla in un passo.
- Per calcolare G occorre risolvere un problema inverso agli autovalori. Per n grande l'algoritmo disponibile [VanDooren 1984] ha un costo eccessivo: $14/3n^3 + 15/2mn^2 + 3m^2n$.

Proiezione su uno spazio A-invariante sinistro

- Seguendo il lavoro di Saad (1988) per ottenere un pole-placement parziale, scelto $k > 0$ calcoliamo una base ortonormale Q_k dello spazio invariante sinistro $\mathcal{Q}_k$ corrispondente ai primi k autovalori di A, con la fattorizzazione parziale di Schur di A^T:

$$A^T Q_k = Q_k R_k \tag{7}$$

Proiezione su uno spazio A-invariante sinistro

- Seguendo il lavoro di Saad (1988) per ottenere un pole-placement parziale, scelto $k > 0$ calcoliamo una base ortonormale Q_k dello spazio invariante sinistro $\mathcal{Q}_k$ corrispondente ai primi k autovalori di A, con la fattorizzazione parziale di Schur di A^T:

$$A^T Q_k = Q_k R_k \tag{7}$$

- Ora:

$$\begin{aligned} \Gamma^T Q_k &= \left[A^T - C^T G^T\right] Q_k \\ &= Q_k R_k - C^T G^T Q_k \\ &\neq Q_k M \qquad , \qquad M \in \Re^{k \times k} \end{aligned} \tag{8}$$

$\Rightarrow$ la dimensione non si riduce, causa la presenza di C.

Proiezione su uno spazio A-invariante

- Allora, prendendo un controllore di tipo $G = Q_k \tilde{G} Q_k^T$:

$$\begin{aligned} Q_k^T \Gamma^T Q_k &= Q_k^T \left[A^T - C^T G^T \right] Q_k \\ &= R_k - Q_k^T C^T Q_k \tilde{G}^T Q_k^T Q_k \\ &= \left[R_k - S \tilde{G}^T \right] \end{aligned} \tag{9}$$

Proiezione su uno spazio A-invariante

- Allora, prendendo un controllore di tipo $G = Q_k \tilde{G} Q_k^T$:

$$\begin{aligned} Q_k^T \Gamma^T Q_k &= Q_k^T \left[A^T - C^T G^T \right] Q_k \\ &= R_k - Q_k^T C^T Q_k \tilde{G}^T Q_k^T Q_k \\ &= \left[R_k - S \tilde{G}^T \right] \end{aligned} \tag{9}$$

- Analisi dell'errore: consideriamo la fattorizzazione completa di Schur di A^T: $A^T [Q_k \, W_{n-k}] = [Q_k \, W_{n-k}] R$ ed esprimiamo l'errore in tale base: $e^{(i)} = Q_k \tilde{e}^{(i)} + W_{n-k} \bar{e}^{(i)}$. Allora:

$$\begin{aligned} (\tilde{e}^{(i+1)})^T = (e^{(i+1)})^T Q_k &= (\tilde{e}^{(i)})^T Q_k^T \Gamma^T Q_k + (\bar{e}^{(i)})^T W_{n-k}^T \Gamma \\ (\bar{e}^{(i+1)})^T = (e^{(i+1)})^T W_{n-k} &= (e^{(i)})^T A^T W_{n-k} \end{aligned} \tag{10}$$

Proiezione su uno spazio A-invariante

- Si può dimostrare che il controllore G lascia inalterati gli autovalori relativi a $\mathcal{Q}_k^{\perp}$ e che quindi l'azione di controllo non ha alcun effetto sulle componenti di errore relative a tale sottospazio.

Proiezione su uno spazio A-invariante

- Si può dimostrare che il controllore G lascia inalterati gli autovalori relativi a $\mathcal{Q}_k^{\perp}$ e che quindi l'azione di controllo non ha alcun effetto sulle componenti di errore relative a tale sottospazio.
- Inoltre, dato che la matrice $A = I_h$ ha solo un autovalore $\lambda = 1$ di molteplicità n, le componenti d'errore relative a $\mathcal{Q}_k^{\perp}$ rimangono costanti durante le iterazioni.

Proiezione su uno spazio A-invariante

- Si può dimostrare che il controllore G lascia inalterati gli autovalori relativi a $\mathcal{Q}_k^{\perp}$ e che quindi l'azione di controllo non ha alcun effetto sulle componenti di errore relative a tale sottospazio.
- Inoltre, dato che la matrice $A = I_h$ ha solo un autovalore $\lambda = 1$ di molteplicità n, le componenti d'errore relative a $\mathcal{Q}_k^{\perp}$ rimangono costanti durante le iterazioni.
- Allora, applichiamo ν passi di pre-rilassamento; es. da uno splitting additivo $K_h = M_h - N_h$, si ottiene:

$$A = (M_h^{-1} N_h)^{\nu} \tag{11}$$

Proiezione su uno spazio A-invariante

- Si può dimostrare che il controllore G lascia inalterati gli autovalori relativi a $\mathcal{Q}_k^{\perp}$ e che quindi l'azione di controllo non ha alcun effetto sulle componenti di errore relative a tale sottospazio.
- Inoltre, dato che la matrice $A = I_h$ ha solo un autovalore $\lambda = 1$ di molteplicità n, le componenti d'errore relative a $\mathcal{Q}_k^{\perp}$ rimangono costanti durante le iterazioni.
- Allora, applichiamo ν passi di pre-rilassamento; es. da uno splitting additivo $K_h = M_h - N_h$, si ottiene:

$$A = (M_h^{-1} N_h)^{\nu} \tag{11}$$

- Il calcolo della base Q_k può essere svolto dagli algoritmi di ARPACK ed è piuttosto oneroso.

Proiezione su uno spazio A-invariante

k	CPU time	Relaxation steps
1	0.571	822
2	0.556	394
4	0.515	388
6	0.784	421
8	1.168	399
10	1.491	394

Table: CPU time and iteration counts for DB-Schur.

Come atteso, il metodo è computazionalmente troppo oneroso, confrontato con PCG, che richiede 0.043s, o perfino un metodo stazionario come Jacobi, che richiede 0.155s con 822 iterazioni.

Proiezione su uno spazio di Krylov

- Proiettiamo allora il sistema nello spazio di Krylov $\mathcal{K}_k(A^T, r^{(0)}) = Span\{r^{(0)}, A^T r^{(0)}, \ldots, (A^T)^{k-1} r^{(0)}\}$. Mediante l'algoritmo di Arnoldi, costruiamo una base ortonormale V_k di $\mathcal{K}_k(A^T, r^{(0)})$ la quale soddisfa:

$$A^T V_k = V_k H_k + h_{k+1,k}\, v_{k+1}\, e_k^T \tag{12}$$

Proiezione su uno spazio di Krylov

- Proiettiamo allora il sistema nello spazio di Krylov $\mathcal{K}_k(A^T, r^{(0)}) = Span\{r^{(0)}, A^T r^{(0)}, \ldots, (A^T)^{k-1} r^{(0)}\}$. Mediante l'algoritmo di Arnoldi, costruiamo una base ortonormale V_k di $\mathcal{K}_k(A^T, r^{(0)})$ la quale soddisfa:

$$A^T V_k = V_k H_k + h_{k+1,k}\, v_{k+1}\, e_k^T \tag{12}$$

- Analogamente alla (9), si ha:

$$\begin{aligned} V_k^T \Gamma^T V_k &= V_k^T \left[A^T - C^T G^T\right] V_k \\ &= H_k - V_k^T C^T V_k \tilde{G}^T V_k^T V_k \\ &= \left[H_k - S\tilde{G}^T\right] \end{aligned} \tag{13}$$

Notare che V_k approssima Q_k, ad un costo inferiore.

Proiezione su uno spazio di Krylov

Caso di K_h simmetrica:

ψ	DB-Krylov		PCG		GMRES		Jacobi	
	CPU	k	CPU	Iter	CPU	k	CPU	Iter
0.02	1.142	4	0.646	206	1.890	30	9.681	>5000
0.05	0.505	2	0.458	146	0.564	25	2.107	1066
0.10	0.384	2	0.411	131	0.464	15	1.349	680
0.20	0.212	2	0.218	68	0.172	10	0.296	144
0.39	0.127	1	0.093	27	0.069	5	0.056	22

Table: $Pe = 0$

Proiezione su uno spazio di Krylov

Caso di K_h non simmetrica:

ψ	DB-Krylov		GMRES	
	CPU	k	CPU	k
0.02	0.380	1	0.478	5
0.05	0.352	1	0.543	3
0.10	0.345	1	0.462	3
0.20	0.341	1	0.450	3
0.39	0.258	1	0.419	3

Table: $Pe = 0.5$

Proiezione su uno spazio di Krylov

Caso di K_h non simmetrica:

ψ	DB-Krylov		GMRES	
	CPU	k	CPU	k
0.02	0.109	1	0.447	1
0.05	0.106	1	0.502	1
0.10	0.107	1	0.561	5
0.20	0.106	1	0.556	3
0.39	0.105	1	0.571	3

Table: $Pe = 0.99$

Proiezione su un insieme di direzioni A-coniugate

Sia p_i la i-esima direzione coniugata, si ha:

$$p_i^T A^T p_i = \alpha_T \tag{14}$$

$$\begin{aligned} p_i^T \Gamma^T p_i &= p_i^T \left[A^T - C^T G^T\right] p_i \\ &= \alpha_T - p_i^T K_h^T p_i \tilde{\gamma} p_i^T p_i \\ &= \alpha_T - \kappa_T \tilde{\gamma} p_i^T p_i \\ &= \alpha_T - \sigma\tilde{\gamma} \end{aligned}$$

dove $\kappa_T = p_i^T K_h^T p_i$ and $\sigma = \kappa_T p_i^T p_i$. Cercando un $\tilde{\gamma}$ tale che $p_i^T \Gamma^T p_i = 0$ (che è un deadbeat in una dimensione), l'espressione di G diventa:

$$G = p_i \frac{\alpha_T}{\sigma} \tag{15}$$

e nel caso $A = I_h$ and K_h simmetrica, si ha il Gradiente Coniugato.

Proiezione su una multi-griglia

Otteniamo:

$$\begin{aligned}
\left(I_{2h}^{h}\right)^{T}\Gamma^{T}\left(I_{h}^{2h}\right)^{T} &= I_{h}^{2h}\Gamma^{T}I_{2h}^{h} \\
&= I_{h}^{2h}\left[A^{T}-C^{T}G^{T}\right]I_{2h}^{h} \\
&= A_{2h}^{T}-I_{h}^{2h}\tilde{C}^{T}\tilde{G}^{T}\left(I_{2h}^{h}\right)^{T}I_{2h}^{h} \\
&= A_{2h}^{T}-S\hat{G}^{T}
\end{aligned}$$

Conclusioni

DB-Krylov:

- scelta del "metodo di rilassamento"

Conclusioni

DB-Krylov:

- scelta del "metodo di rilassamento"
- scelta di ν_p

Conclusioni

DB-Krylov:

- scelta del "metodo di rilassamento"
- scelta di ν_p
- scelta del riutilizzo di v_k

Conclusioni

DB-Krylov:

- scelta del "metodo di rilassamento"
- scelta di ν_p
- scelta del riutilizzo di v_k
- scelta di k