

Richiami sul GC

Dato il sistema $A\mathbf{x} = \mathbf{b}$, con A **matrice simmetrica definita positiva**, applichiamo lo schema del Gradiente Coniugato:

- si parte da un'approssimazione iniziale \mathbf{x}_0

- si applica la formula $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$

- con α_k che minimizza il funzionale dell'errore

$$\Phi(\mathbf{x}_{k+1}) = \mathbf{e}_{k+1}^T A \mathbf{e}_{k+1} = (\mathbf{x} - \mathbf{x}_{k+1})^T A (\mathbf{x} - \mathbf{x}_{k+1})$$

per cui $\alpha_k = \frac{\mathbf{p}_k^T \mathbf{r}_k}{\mathbf{p}_k^T A \mathbf{p}_k}$ item dove $\mathbf{r}_k = \mathbf{b} - A\mathbf{x}_k$ è il residuo associato alla k -sima iterazione. Il nuovo residuo si calcola mediante

$$\mathbf{r}_{k+1} = \mathbf{b} - A(\mathbf{x}_k + \alpha_k \mathbf{p}_k) = \mathbf{r}_k - \alpha_k A \mathbf{p}_k$$

- La direzione di ricerca \mathbf{p}_{k+1} viene calcolata imponendo la A -ortogonalizzazione rispetto a \mathbf{p}_k mediante la formula:

$$\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k$$

da cui

$$\beta_k = -\frac{\mathbf{r}_{k+1}^T A \mathbf{p}_k}{\mathbf{p}_k^T A \mathbf{p}_k}$$

Algoritmo

- Si sceglie \mathbf{x}_0 iniziale;
- si pone $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$ e $\mathbf{p}_0 = \mathbf{r}_0$
- (k=0)
- fino a convergenza si applica:
 - $k=k+1$
 - $\alpha_k = \frac{\mathbf{p}_k^T \mathbf{r}_k}{\mathbf{p}_k^T A \mathbf{p}_k}$
 - $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k$
 - $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A$
 - $\beta_k = -\frac{\mathbf{r}_{k+1}^T A \mathbf{p}_k}{\mathbf{p}_k^T A \mathbf{p}_k}$
 - $\mathbf{p}_{k+1} = \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k$

Sulla convergenza

Si considera una opportuna norma del residuo (per esempio, la norma euclidea) e si controlla se questa è inferiore rispetto ad una prefissata tolleranza data in input.

Osservazione: Risulta più conveniente controllare il residuo relativo, scritto in funzione del termine noto: $r_r = \frac{\|\mathbf{r}_{k+1}\|}{\|\mathbf{b}\|}$

Quindi si controlla se $r_r = \frac{\|\mathbf{r}_{k+1}\|}{\|\mathbf{b}\|} < \text{toll}$

Attenzione: Per effetto degli errori di arrotondamento, se il problema è mal condizionato la formula ricorsiva utilizzata per l'aggiornamento del residuo può portare ad un accumulo di tali errori. Quindi il residuo "vero" - calcolato mediante la formula della definizione - potrebbe essere diverso dal residuo calcolato tramite la formula ricorsiva.

In tal caso, la soluzione approssimata che ricaviamo tramite il criterio di convergenza stabilito prima ($r_r = \frac{\|r_{k+1}\|}{\|b\|} < \text{toll}$) può non essere la soluzione giusta!

Cosa fare? Se risulta per un certo indice \bar{k} che $x_{\bar{k}}$ soddisfa il criterio di convergenza per il residuo relativo, e quindi usciamo dal ciclo dell'algoritmo del Gradiente Coniugato, successivamente calcoliamo il residuo relativo “vero” e verificiamo se anche per questo è verificata la condizione di convergenza, cioè se:

$$\frac{\|b - Ax_{\bar{k}}\|}{\|b\|} < \text{toll}$$

Se è verificata la disuguaglianza, allora consideriamo $x_{\bar{k}}$ una buona approssimazione della soluzione. Altrimenti, se il residuo relativo “vero” è molto maggiore della tolleranza richiesta, la soluzione approssimata non è “buona”. Quanto non sia buona lo si vede caso per caso.

Gradiente Coniugato Precondizionato

Tuttavia, sappiamo dalla teoria che non conviene utilizzare lo schema del GC così come è. Occorre **precondizionare**. Si passa al sistema equivalente $B\mathbf{y} = \mathbf{c}$ dove

$$B = X^{-1}AX^{-1}, \mathbf{y} = X\mathbf{x} \text{ e } \mathbf{c} = X^{-1}\mathbf{b}$$

La matrice $K^{-1} = X^{-1}X^{-1}$ è la matrice di precondizionamento (ad esempio, precondizionamento diagonale o precondizionamento secondo Cholesky).

Con il precondizionamento, si ha lo schema del GCM

$$\begin{aligned}
\mathbf{x}_{k+1} &= \mathbf{x}_k + \alpha_k \mathbf{p}_k \\
\mathbf{r}_{k+1} &= \mathbf{r}_k - \alpha_k A \mathbf{p}_k \\
\mathbf{p}_{k+1} &= K^{-1} \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k
\end{aligned}$$

con

$$\alpha_k = \frac{\mathbf{p}_k^T \mathbf{r}_k}{\mathbf{p}_k^T A \mathbf{p}_k} \quad \beta_k = -\frac{\mathbf{r}_{k+1}^T K^{-1} A \mathbf{p}_k}{\mathbf{p}_k^T A \mathbf{p}_k}$$

Si prende $\mathbf{r}_0 = \mathbf{b} - A \mathbf{x}_0$ e $\mathbf{p}_0 = K^{-1} \mathbf{r}_0$.

Correzioni Residue

Si può rendere più veloce lo schema del GCM, andando a prendere come approssimazione iniziale \mathbf{x}_0 , il valore che si ottiene applicando lo schema delle Correzioni Residue allo stesso sistema lineare, vale a dire:

$$\begin{aligned}\mathbf{x}_0 &= K^{-1}\mathbf{b} \\ \mathbf{x}_1 &= \mathbf{x}_0 + K^{-1}\mathbf{r}_0 \\ &\vdots \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + K^{-1}\mathbf{r}_k\end{aligned}$$

Si effettuano, quindi, un certo numero n_{corr} di Correzioni Residue e l'approssimazione che si ottiene si utilizza come approssimazione iniziale per il GCM.

Computazionalmente

Da un punto di vista computazionale, lo schema del GCM richiede, ad ogni iterazione solo due prodotti matrice-vettore $A\mathbf{p}_k$ e $K^{-1}\mathbf{r}_{k+1}$ - che sono le operazioni più costose da fare.

Per sfruttare pienamente le potenzialità dello schema, conviene memorizzare la matrice del sistema non come una matrice piena (cioè evitando di memorizzare tutti gli elementi nulli della matrice) ma in una maniera particolare che tiene conto della sparsità (degli zeri) della matrice stessa, vale a dire in forma compatta. Un tipo di memorizzazione utilizzato a tal proposito è la memorizzazione CRS.

Memorizzazione di matrici in forma compatta

Sistema di memorizzazione chiamato **Compressed Row Storage (CRS)**

Vediamo per prima il caso di una matrice A $n \times n$, sparsa, **non simmetrica**.

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 5 & 0 \\ 0 & 0 & 6 & 7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 8 & 0 \\ 0 & 0 & 9 & 0 & 0 & 10 \end{pmatrix}$$

Sia nt il numero di coefficienti non nulli di A . In questo caso $nt = 10$, $n = 6$.

Elemento di A	1	2	3	4	5	6	7	8	9	10
Indice di colonna	1	1	2	3	5	3	4	5	3	6
Puntatore di riga	1	2	4	6	8	9	11			

Chiamiamo con SYSMAT il vettore che contiene i coefficienti non nulli di A , presi in successione riga per riga:

$\text{SYSMAT} =$	1	2	3	4	5	6	7	8	9	10
-------------------	---	---	---	---	---	---	---	---	---	----

Sia JA il vettore degli interi che contiene gli nt indici di colonna dei corrispondenti elementi di SYSMAT

$\text{JA} =$	1	1	2	3	5	3	4	5	3	6
---------------	---	---	---	---	---	---	---	---	---	---

Sia IA il vettore di $n + 1$ componenti, che contiene le posizioni in cui si trova in SYSMAT il primo elemento di ciascuna riga di A .

L'ultimo elemento di IA è posto uguale a $nt + 1$

$\text{IA} =$	1	2	4	6	8	9	11
---------------	---	---	---	---	---	---	----

A — SYSMAT

Quindi se $\text{SYSMAT}(k) = a_{ij}$, allora

$$\text{JA}(k) = j$$

Poichè IA è il vettore che dà le locazioni del primo elemento di ogni riga di A in SYSMAT, allora

$$\text{IA}(i) \leq k \leq \text{IA}(i + 1) - 1.$$

Si pone $\text{IA}(n + 1) = nt + 1$ per non avere problemi quando si indicano gli elementi dell'ultima riga di A .

SYSMAT — A

Quindi, un elemento della matrice A viene individuato tramite i vettori SYSMAT, JA, IA.

a_{ij} elemento di riga i e colonna j di A occupa una posizione k del vettore SYSMAT compresa nell'intervallo

$$IA(i) \leq k \leq IA(i + 1) - 1$$

e tale che $JA(k) = j$.

I valori di k compresi tra $IA(i)$ e $IA(i + 1) - 1$ individuano tutti gli elementi della riga i -sima di A nel vettore SYSMAT.

Poichè $JA(k)$ fornisce l'indice di colonna, se $JA(k) = j$ allora ho individuato l'elemento a_{ij} nel vettore SYSMAT.

Esempio

$$A = \begin{pmatrix} 4 & 0 & 2 & 0 & 0 & 0 \\ 2 & 3 & 0 & 0 & 0 & 0 \\ 0 & 1 & 5 & 0 & 0 & 2 \\ 0 & 0 & 6 & 7 & 8 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 \\ 0 & 2 & 0 & 0 & 0 & 9 \end{pmatrix}$$

SYSMAT =	4	2	2	3	1	5	2	6	7	8	5	2	9
JA =	1	3	1	2	2	3	6	3	4	5	5	2	6
IA=	1	3	5	8	11	12	14						

$$a_{4,5} = 8$$

Considero $IA(4) = 8$ e $IA(5) - 1 = 10$. Per $k = 8, 9, 10$ in JA ho i valori $JA(8) = 3$, $JA(9) = 4$ e $JA(10) = 5$

$JA(10) = 5$ fornisce l'indice di colonna di $a_{4,5}$, perciò in SYSMAT(10) è memorizzato il valore di $a_{4,5}$. Difatti $SYSMAT(10) = 8$.

Matrici simmetriche

Quando la matrice A è simmetrica, nel sistema CRS viene memorizzata solo la parte triangolare alta di A , includendo la diagonale principale della matrice stessa.

Si tiene conto del fatto che $a_{ij} = a_{ji}$ e si memorizzano solo gli elementi a_{ij} con $j \geq i$.

Nel vettore SYSMAT andremo a memorizzare solo gli elementi della parte alta di A e quindi nt non rappresenta più il numero di coefficienti non nulli di A (come nel caso non simmetrico) ma rappresenta il numero di coefficienti non nulli della parte alta di A .

Esempio

$A =$

$$\begin{pmatrix} 10 & 0 & 0 & 0 & 1 & 0 & 2 \\ 0 & 1 & 0 & 3 & 0 & -1 & 0 \\ 0 & 0 & 4 & 0 & 0 & 0 & 1 \\ 0 & 3 & 0 & 2 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 5 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 20 \end{pmatrix} = \begin{pmatrix} \boxed{10} & & & & \boxed{1} & & \boxed{2} \\ & \boxed{1} & & & & & \\ & & \boxed{4} & & & & \\ & & & \boxed{3} & & & \\ & & & & \boxed{2} & & \\ & & & & & \boxed{1} & \\ & & & & & & \boxed{5} \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & & \\ & & & & & \boxed{1} & \\ & & & & & & \\ & & & & & & \boxed{20} \end{pmatrix}$$

$n = 7$, elementi non nulli: 19, elementi non nulli da memorizzare: $nt = 13$

SYSMAT =	10	1	2	1	3	-1	4	1	2	1	5	1	20
JA =	1	5	7	2	4	6	3	7	4	5	5	6	7
IA=	1	4	7	9	11	12	13	14					

Ora IA individua gli elementi della diagonale principale di ogni riga.

Poichè si tratta di matrici simmetriche si vede anche che

$$IA(1) = 1, IA(n) = nt,$$

$$JA(1) = 1, JA(nt) = n.$$

Prodotto matrice-vettore: caso non simmetrico

$$A\mathbf{v} = \mathbf{w}$$

A matrice quadrata non simmetrica di ordine n ,
 \mathbf{v} e \mathbf{w} vettori di dimensione n .

La componente i -sima del vettore \mathbf{w} è dato da:

$$w_i = \sum_{j=1}^n a_{ij} v_j$$

Algoritmo per matrici non simmetriche

```
001      Per  $i = 1, n$ 
002          azzero  $w_i$ 
003          Per  $k = \text{IA}(i), \text{IA}(i + 1) - 1$ 
004               $j := \text{JA}(k)$ 
005               $w_i := w_i + \text{SYSMAT}(k) \cdot v_j$ 
006          Fine Per
007      Fine Per
```

Osservazione: conviene azzerare le componenti del vettore w per evitare di usare impropriamente valori precedentemente contenuti nel medesimo vettore.

Prodotto matrice-vettore: caso simmetrico

Noi abbiamo memorizzati solo gli elementi della parte alta di A , a_{ij} , $j \geq i$, quindi la componente i -sima di w si può scrivere:

$$\begin{aligned}w_i &= \sum_{j=1}^n a_{ij} v_j \\ &= \sum_{j=1}^{i-1} a_{ij} v_j + \sum_{j=i}^n a_{ij} v_j \\ &= \sum_{j=1}^{i-1} a_{ji} v_j + \sum_{j=i}^n a_{ij} v_j\end{aligned}$$

La seconda sommatoria può essere implementata senza problemi come nel caso non simmetrico. Per la prima sommatoria, non conviene andare a cercare in SYSMAT le componenti che appartengono alla colonna i -sima per ottenere a_{ji} perchè dovremmo cercare su nt componenti e applicare la procedura n volte, con un dispendio notevole dal punto di vista computazionale.

Conviene osservare che gli elementi della parte alta della riga i -sima di A contribuiscono non solo al calcolo di w_i ma, per la simmetria, anche al calcolo di w_j , con j uguale all'indice di colonna considerato nella sommatoria.

Fisso infatti due indici i e l , con $l \geq i$.

$$w_i = \sum_{j=1}^{i-1} a_{ji} v_j + \sum_{j=i}^n a_{ij} v_j$$

$$w_l = \sum_{j=1}^{l-1} a_{jl} v_j + \sum_{j=l}^n a_{lj} v_j$$

Nel calcolo di w_i , nella seconda sommatoria, ci sarà un indice $j = l$ che darà il contributo $a_{il} v_l$.

Ma $a_{il} = a_{li}$: questo termine si trova nel calcolo di w_l , nella prima sommatoria, per $j = i$: $a_{il} v_i$.

Quindi possiamo aggiornare la sommatoria relativa al calcolo di w_i e, nello stesso tempo, aggiornare le sommatorie relative al calcolo di w_j con $j = i, \dots, n$.

Algoritmo per matrici simmetriche

```
001     Per  $i = 1, n$ 
002         azzero  $w_i$ 
003     Fine Per
004     Per  $i = 1, n$ 
005          $k := \text{IA}(i)$ 
006          $w_i := w_i + \text{SYSMAT}(k) \cdot v_i$ 
007         Per  $k = \text{IA}(i) + 1, \text{IA}(i + 1) - 1$ 
008              $j := \text{JA}(k)$ 
009              $w_i := w_i + \text{SYSMAT}(k) \cdot v_j$ 
010              $w_j := w_j + \text{SYSMAT}(k) \cdot v_i$ 
011         Fine Per
012     Fine Per
```


Osservazioni:

Il vettore w deve essere azzerato in un ciclo esterno rispetto al calcolo di w .

Il contributo a w_i dato dall'elemento diagonale a_{ii} è fatto a parte (riga 6) per evitare di considerarlo due volte nel ciclo successivo.