3. Risoluzione di Sistemi Lineari

Definizione problema

Sia $A \in \mathbb{R}^{m \times n}$ e $b \in \mathbb{R}^m$ vogliamo trovare un vettore $x \in \mathbb{R}^n$ che soddisfi l'equazione $Ax = b$.

Tratteremo solo sistemi quadrati, in cui esiste un'unica soluzione x se e solo se:

- esiste l'inversa A^{-1} della matrice A
- oppure il rango della matrice A è n
- oppure $Ax = 0$ e in tal caso $x = 0$

Teorema di Cramer

Se il determinante della matrice è non nullo, allora esiste unica una soluzione del sistema, ed è data da:

$$x_i = \frac{\det(\Delta_i)}{\det(A)} \tag{1}$$

Dove Δ_i è A a cui sostituiamo l'i-esima riga con il vettore dei termini noti.

Numero di condizionamento nei sistemi lineari

Sia A una matrice quadrata, il numero di condizionamento $k(A)$ della matrice è un valore che misura la sensibilità della soluzione di $Ax = b$ alle perturbazioni nei dati. Idealmente vorremmo lavorare con matrici con numero di condizionamento basso. Supponendo l'esistenza dell'inversa, $k(A)$ si calcola come segue:

$$k(A) = \|A\| \cdot \|A^{-1}\| \tag{2}$$

> La matrice identità ha il più basso numero di condizionamento (1) ed è l'esempio perfetto di matrice ben condizionata nel caso della risoluzione di un sistema lineare associato.

In generale, in numero di condizionamento è correlato all'errore sui dati come segue:

$$k(A) \geq \frac{\text{errore sul risultato}}{\text{errore sui dati}} \tag{3}$$

Problemi di calcolo del numero di condizionamento

La complessità del calcolo della matrice inversa cresce insieme alla crescita del numero di condizionamento. Ma per calcolare il numero di condizionamento è necessaria la matrice inversa! Affermiamo che:

1. Se $k(A)$ è vicino ad 1 allora $k(A)$ è facilmente calcolabile
2. Se $k(A)$ è grande, allora è difficilmente ricavabile

Correlazione tra numero di condizionamento e raggio spettrale

Supponendo l'esistenza dell'inversa, è presente la seguente relazione tra i due valori:

$$k(A) \geq \rho(A)\rho(A^{-1}) \tag{4}$$

Questo implica che:

$$k(A) \geq \frac{\max \lambda}{\min \lambda} \tag{5}$$

Banalmente al numeratore è presente il raggio spettrale di A. Sappiamo che se esiste λ autovalore di A, allora λ^{-1} è autovalore di A^{-1}. Dato che invertiamo tutti gli autovalori, l'autovalore più piccolo di A, una volta invertito, diventa l'autovalore più grande di A^{-1}:

$$\rho(A)\rho(A^{-1}) = \max \lambda \cdot \frac{1}{\min \lambda} = \frac{\max \lambda}{\min \lambda} \tag{6}$$

Matrice di Hilbert

La matrice di Hilbert è un esempio di matrice mal condizionata per i sistemi lineari (ma è ben condizionata per altri problemi). In generale, l'elemento $h_{ij} = (i + j - 1)^{-1}$, approfondire qui.

Metodi diretti

I metodi diretti trovano la soluzione del sistema lineare in un numero finito di passi. Sono adatti a sistemi con matrici piene, essendo che tendono a rimpire gli zeri della matrice.

Metodo delle sostituzioni in avanti

Supponiamo che la matrice A del sistema sia triangolare inferiore. Il metodo delle sostituzioni in avanti risolve il sistema una soluzione per volta. Partendo da x_1 come segue:

$$x_1 = \frac{b_1}{l_{11}} \tag{7}$$

Si calcola il generico x_i

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} l_{ij}x_j}{l_{ii}} \tag{8}$$

Metodo delle sostituzioni indietro

Supponiamo che la matrice A del sistema sia triangolare superiore. Il metodo delle sostituzioni indietro risolve il sistema una soluzione per volta, partendo dal basso. Partendo da x_n come segue:

$$x_n = \frac{b_n}{u_{nn}} \tag{9}$$

Si calcola il generico x_i

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} u_{ij}x_j}{u_{ii}} \tag{10}$$

Complessità metodi di sostituzione e considerazioni

I metodi di sostituzione presentati eseguono circa $\frac{n(n+1)}{2}$ divisioni e moltiplicazioni e $\frac{n(n-1)}{2}$ somme algebriche, per un costo computazionale $\Theta(n^2)$. Per derivare la complessità basta osservare che nella prima iterazione facciamo 1 div/mul, nella seconda 2, nella terza 3 e così via. Avendo n righe totali basta calcolare la somma dei primi n numeri naturali, che corrisponde proprio alla prima formula mostrata.

Questi metodi sono molto leggeri, ma necessitano di matrici triangolari. Gli altri metodi presentati cercheranno di triangolarizzare la matrice, per poi utilizzare i metodi di sostituzione per risolvere il sistema lineare. Se dell'istanza del problema cambia solo il vettore dei coefficienti, è possibile precalcolarsi alcuni dei parametri per rendere il metodo più rapido.

Metodo di eliminazione naive di Gauss (MEG)

Supponendo che la matrice A sia non degenere, e sia $a_{11} \neq 0$ (altrimenti scambia righe), allora è possibile applicare il metodo di Gauss (a meno di un'altra condizione che enunceremo dopo). Alla prima iterazione si calcolano i moltiplicatori:

$$m_{i1}^{(1)} = -\frac{a_{i1}}{a_{11}} \qquad i = 2, \dots, n \tag{11}$$

Aggiungiamo alla i-esima equazione la prima equazione moltiplicata per m_{i1}. Così facendo andremo ad annullare tutti gli elementi della prima colonna meno che il primo. In generale, durante la i-esima iterazione lo scopo è a nnullare tutti gli elementi della i-esima colonna al di sotto dell'i-esimo, quindi si calcolano i moltiplicatori tramite l'elemento a_{ii}, che prende il nome di pivot:

$$m_{ji}^{(i)} = -\frac{a_{ji}}{a_{ii}} \qquad j = i+1, \dots, n \tag{12}$$

E si aggiunge l'i-esima equazione alle successive moltiplicata per il rispettivo moltiplicatore. Al passo $n-1$ si ottiene un sistema triangolare superiore che può essere risolto con la sostituzione all'indietro. Il costo computazionale del metodo è circa $\frac{4}{3}n^3$.

Affinché il metodo funzioni è necessario che gli elementi della diagonale a_{ii} siano non nulli ad ogni iterazione. Questo è garantito se tutti i minori principali di A sono non nulli.

MEG con Pivot Parziale

La tecnica del pivot parziale evita le divisioni per zero o per numeri prossimi allo zero nel calcolo dei moltiplicatori. Alla i-esima iterazione si cerca la riga $i \leq k \leq n$ con l'elemento maggiore nella colonna i.

$$a_{ki} = \max_{i \leq s \leq n} |a_{si}| \tag{13}$$

Si sostituisce la riga k-esima con la riga i-esima. Il beneficio sta nel fatto che viene minimizzata la grandezza del moltiplicatore, minimizzando anche l'errore amplificato durante le moltiplicazioni. La complessità totale del metodo è $O(n^2)$.

MEG con Pivot Totale

Il pivot totale ha gli stessi benefici del pivot parziale, ma amplificati: il metodo può scegliere tra tutti gli elementi della matrice (incompleta) e non solo quelli della colonna analizzata. Il drawback sta nella complessità implementativa e nel cambio di variabile necessario.

Supponiamo che il pivot corrente sia a_{ii} e che l'elemento più grande della matrice sia a_{rs}. Allora prima si effettua una permutazione delle colonne $i \leftrightarrow s$ e si memorizza il cambio di variabile, e dopodiché si scambiano le righe $i \leftrightarrow r$. Quando si ottiene la soluzione, per ottenere la soluzione rispetto al problema originale bisogna applicare i cambi di variabile a ritroso alla soluzione.

Fattorizzazione LU

Sia $Ax = b$ il sistema lineare da risolvere, un metodo di fattorizzazione matriciale consiste nei seguenti passi:

1. Si trova una matrice S non singolare tale che $SAx = Sb$ e $SA = U$ triangolare superiore.
2. Se S è triangolare inferiore, lo sarà anche S^{-1} e poniamo $L = S^{-1}$
3. Osserviamo che $A = LU = S^{-1}SA = A$ quindi LU è una fattorizzazione di A

La fattorizzazione non dipende dai termini noti, quindi se nel sistema lineare in analisi variano solo i termini noti, precalcolando la fattorizzazione si ha un risparmio in efficienza. Attraverso la fattorizzazione LU riformuliamo il metodo di eliminazione di Gauss. Siano:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \ddots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \qquad L_1 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ m_{21} & 1 & \cdots & 0 \\ \vdots & & \ddots & \\ m_{n1} & 0 & \cdots & 1 \end{bmatrix} \tag{14}$$

dove m_{i1} per $i = 2, \ldots, n$ sono i moltiplicatori mostrati in precedenza. Il prodotto $L_1 A$ equivale al primo passo di Gauss. In generale, la matrice L_i è definita come segue:

$$L_i = \begin{bmatrix} 1 & \ldots & 0 & \ldots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \ldots & 1 & \ldots & 0 \\ 0 & \ldots & m_{i+1,i} & \ldots & 0 \\ \vdots & & \vdots & & \vdots \\ 0 & \ldots & m_{n,i} & \ldots & 1 \end{bmatrix} \tag{15}$$

Alla fine si ha:

$$U = L_{n-1} L_{n-2} \ldots L_2 L_1 A \tag{16}$$

Poniamo $\tilde{L} = L_{n-1} L_{n-2} \ldots L_2 L_1$, quindi $U = \tilde{L} A$. Poniamo $L = \tilde{L}^{-}1$ e osserviamo che:

$$LU = \tilde{L}^{-1} U = \tilde{L}^{-1} \tilde{L} A = A \tag{17}$$

Adesso, la soluzione del sistema lineare $Ax = b$ si risolve in due passaggi:

1. $Ly = b$ e si risolve per y
2. $Ux = y$ e si risolve per x

Questo poiché con il passo (1) si trova $y = L^{-1}b$, mentre con il passo 2 si trova $x = U^{-1}L^{-1}b$, che è sicuramente soluzione del sistema $Ax = LUx = b$, ed entrambi possono essere risolti con sostituzioni in avanti ed indietro, essendo matrici triangolari inferiori (L) e superiori (U).

Non sempre esiste una fattorizzazione LU della matrice. Condizione necessaria di esistenza è che la matrice sia non degenere, quindi che abbia il determinante non nullo. Se vale tale condizione, allora esiste sicuramente una matrice di permutazione P tale che $PA = LU$. L'esistenza è garantita per le matrici **diagonalmente dominanti** e **simmetriche definite positive**, senza dover permutare la matrice.

Matrici di permutazione

Le tecniche del pivot parziale e totale possono essere implementate nella forma matriciale del metodo di eliminazione di Gauss attraverso delle matrici di permutazione. Una matrice di permutazione è una matrice ottenuta scambiando righe o colonne della matrice identità. In particolare, scambiando la riga i con la riga j di I e

- premoltiplicandola per A si scambiano le righe
- postmoltiplicandola per A si scambiano le colonne

Esempio: scambiamo la riga 1 e la riga 3 della matrice identità e vediamo cosa succede se la premoltiplichiamo ad una matrice qualunque:

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 4 & 2 & 7 \\ 6 & 2 & 9 \\ 7 & 3 & 5 \end{bmatrix} = \begin{bmatrix} 7 & 3 & 5 \\ 6 & 2 & 9 \\ 4 & 2 & 7 \end{bmatrix} \tag{18}$$

L'effetto è stato uno scambio di righe. Se invece postmoltiplichiamo la matrice di permutazione:

$$\begin{bmatrix} 4 & 2 & 7 \\ 6 & 2 & 9 \\ 7 & 3 & 5 \end{bmatrix} \cdot \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 7 & 2 & 4 \\ 9 & 2 & 6 \\ 5 & 3 & 7 \end{bmatrix} \tag{19}$$

L'effetto è stato uno scambio di colonne.

Metodo di Doolittle e di Croud

Se esplicitassimo il sistema lineare $LU = A$ avremmo n^2 equazioni (una per ogni elemento di A) ed $n^2 + n$ incognite (elementi di L ed U supponendo che siano rispettivamente triangolari inferiore e superiore), questo implica che abbiamo n gradi di libertà, ovvero la fattorizzazione LU non è unica.

I seguenti metodi eliminano i gradi di liberta imponendo dei vincoli:

- **Doolittle**: si fissa $l_{ii} = 1$ in L (equivalente a eliminazione gaussiana senza pivot)
- **Crout**: si fissa $u_{ii} = 1$ in U

Effetto Fill-in

I metodi di fattorizzazione modificano la matrice iniziale causando un effetto fill-in, ovvero gli zeri diventano elementi non nulli. Se la matrice è inizialmente sparsa, conviene optare per metodi iterativi.

Metodo di Cholesky

Sia $A \in \mathbb{R}^{n\times n}$ una matrice simmetrica ($A = A^T$) e definita positiva, allora esiste almeno una matrice L triangolare inferiore tale che $A = LL^T$. Inoltre, se si impone che $l_{ii} > 0$ la fattorizzazione è unica.

Dimostrazione (costruttiva).

Per il criterio di Sylvester, la matrice ha determinante strettamente positivo, e quindi esiste una fattorizzazione LU. Proviamo a costruire la fattorizzazione U^TU come segue:

$$U^TU = \begin{bmatrix} u_{11} & & 0 \\ \vdots & \ddots & \\ u_{1n} & \dots & u_{nn} \end{bmatrix} \begin{bmatrix} u_{11} & \dots & u_{1n} \\ \vdots & \ddots & \\ 0 & \dots & u_{nn} \end{bmatrix} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \\ a_{n1} & \dots & a_{nn} \end{bmatrix} = A \tag{20}$$

Dato che le incognite sono gli elementi di U, che sono $n(n+1)/2$, possiamo strutturare n^2 equazioni (non lineari) tramite il prodotto riga-colonna.

Esplicitiamo il calcolo dell'elemento a_{kj} supponendo che $k > j$ (le moltiplicazioni per indici maggiori di j sono nulle per la triangolarità delle matrici), quindi abbiamo:

$$a_{kj} = \sum_{i=1}^{j} u_{ki}u_{ij} = u_{kj}u_{jj} + \sum_{i=1}^{j-1} u_{ki}u_{ij} \tag{21}$$

Estraiamo l'elemento u_{kj} dall'equazione:

$$u_{kj} = \frac{a_{kj} - \sum_{i=1}^{j-1} u_{ki}u_{ij}}{u_{jj}} \tag{22}$$

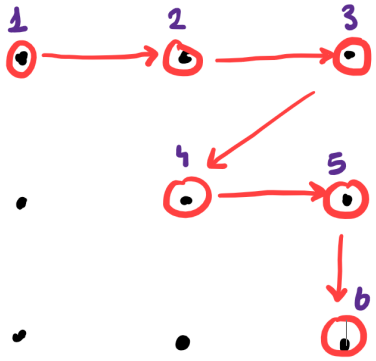
Vediamo invece come calcolare un elemento della diagonale u_{kk}. Tramite la seguente equazione cerchiamo di calcolare l'elemento a_{kk}:

$$a_{kk} = \sum_{p=1}^{k} u_{pk}^2 = u_{kk}^2 + \sum_{p=1}^{k-1} u_{pk}^2 \tag{23}$$

Oss. la sommatoria termina a k e non ad n perché per la struttura triangolare della matrice gli elementi $a_{k,k+1}, \dots, a_{k,n}$ sono nulli. Ora possiamo estrapolare l'elemento u_{kk} come:

$$u_{kk} = \sqrt{a_{kk} - \sum_{p=1}^{k-1} u_{pk}^2} \tag{24}$$

Calcolando u_{11} la sommatoria verrebbe annullata, quindi $u_{11} = \sqrt{a_{11}}$. Bisogna seguire un certo andamento per poter calcolare tutti gli elementi di U, dato che un calcolo potrebbe richiedere dei calcoli di altri elementi della matrice. Nella seguente foto è illustrato tale andamento su una matrice 3×3:



Algoritmo di Thomas

L'algoritmo di Thomas è un TDMA (tridiagonal matrix algorithm), ovvero un algoritmo che opera su matrici tridiagonali. Una matrice tridiagonale di dimensione $n \times n$ ha $3n - 2$ elementi. Scriviamola come prodotto di due matrici particolari, in cui le incognite sono α_i per $i = 1, \dots, n$ e γ_i per $i = 1, \dots, n - 1$.

$$\begin{bmatrix} a_1 & c_1 & 0 & & & \\ b_2 & a_2 & c_2 & 0 & & 0 \\ & b_3 & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & \ddots & \ddots & \ddots \\ & 0 & & & b_{n-1} & a_{n-1} & c_{n-1} \\ & & & & & b_n & a_n \end{bmatrix} = \begin{bmatrix} \alpha_1 & & & & \\ b_2 & \alpha_2 & & 0 & \\ & b_3 & \alpha_3 & & \\ & 0 & \ddots & \ddots & \\ & & & b_n & \alpha_n \end{bmatrix} \begin{bmatrix} 1 & \gamma_1 & & & \\ & 1 & \gamma_2 & 0 & \\ & & \ddots & & \\ & 0 & & \ddots & \gamma_{n-1} \\ & & & & 1 \end{bmatrix}$$

I coefficienti sono determinabili come segue:

- $\alpha_1 = a_1$ e $\alpha_i = a_i - b_i \gamma_{i-1}$ per $i = 2, \dots, n$
- $\gamma_1 = c_1 / \alpha_1$ e $\alpha_i = c_i / \alpha_i$ per $i = 2, \dots, n - 1$

Per una trattazione completa consultare il seguente documento.

Metodi iterativi

Un metodo iterativo produce una successione di soluzioni che, sotto opportune condizioni, converge alla soluzione reale. Comunemente si parte da una soluzione randomica $x^{(0)}$, in quanto si dimostra che il metodo converge comunque.

Descrizione generale di un algoritmo iterativo

Sia $A \in \mathbb{R}^{n \times n}$ una matrice non degenere, si impone che:

$$\begin{cases} Ax = b \\ A = M - N \end{cases} \tag{25}$$

Quindi si ha:

$$Ax = b$$

$$\begin{aligned} (M-N)x &= b \\ Mx - Nx &= b \\ Mx^{(k+1)} &= Nx^{(k)} + b \end{aligned} \tag{26}$$

E quindi si determina la soluzione al passo $k+1$ dalla soluzione al passo precedente:

$$x^{(k+1)} = M^{-1}Nx^{(k)} + M^{-1}b \tag{27}$$

L'equazione $A = M - N$ prende il nome di decomposizione, e si dice regolare se M non è degenere e gli elementi dell'inversa M^{-1} e di N sono tutti maggiori di zero. Spesso la matrice $M^{-1}N$ viene chiamata "matrice del metodo" a cui si fa riferimento, poiché studiandola è possibile studiarne la convergenza.

Criteri d'arresto

Fissata una tolleranza ϵ, allora un criterio d'arresto standard utilizzabile per fermare un algoritmo iterativo è il seguente:

$$\frac{||x^{(k+1)} - x^{(k)}||_\infty}{||x^{(k)}||_\infty} \tag{28}$$

Metodo iterativo convergente

Un metodo iterativo si dice convergente se per qualunque vettore iniziale $x^{(0)}$ il metodo sarà convergente.

Teorema sulla convergenza (1)

Sia $A = M - N$ una decomposizione regolare di A e sia $||M^{-1}N|| \leq \lambda < 1$. Allora:

1. A non è singolare
2. Il metodo iterativo associato alla decomposizione è convergente
3. $||x^{(k)} - x||$ ci dà un limite all'errore commesso

Teorema sulla convergenza (2)

Condizione necessaria e sufficiente affinché il metodo iterativo sia convergente è che il raggio spettrale ρ della matrice $M^{-1}N$ sia minore di 1:

$$\rho(M^{-1}N) < 1 \tag{29}$$

Corollario sulla convergenza (3)

Dato che il determinante di una matrice è anche definito come il prodotto dei suoi autovalori, allora condizione necessaria affinché il metodo converga è che

$$|\det\left(M^{-1}N\right)| < 1 \tag{30}$$

Se il determinante è maggiore di 1 allora esiste almeno un autovalore ≥ 1, e quindi il metodo non può convergere per il teorema precedente.

Corollario sulla convergenza (4)

La traccia di una matrice è definita come la somma dei suoi autovalori. Condizione necessaria affinché il metodo converga è che:

$$|t_r(M^{-1}N)| < n \tag{31}$$

Se vale il contrario, allora almeno uno degli autovalori è ≥ 1, quindi il metodo non può convergere per il teorema precedente.

Correlazione tra raggio spettrale e convergenza

Il raggio spettrale ρ relativo alla matrice di uno specifico metodo è correlato alla convergenza dello stesso come segue:

- Se $\rho \geq 1$ il metodo diverge
- Se $\rho < 1$ il metodo converge
- Tanto più ρ si avvicina ad 1, tanto più il metodo convergerà lentamente

Velocità di convergenza di un metodo iterativo

Tramite la seguente valutazione è possibile ricavare il numero di iterazioni k affinché l'errore e generato da un certo metodo scenda al di sotto di un errore assegnato m. Siano $e^{(k)}$ ed $x^{(k)}$ rispettivamente l'errore e la soluzione generati al passo k. L'errore è ricavato come segue:

$$e^{(k+1)} = x^{(k+1)} - x^{(k)} \tag{32}$$

Che deriva dalla formula di approssimazione dell'errore, dove l'x soluzione reale viene approssimato con la soluzione calcolata allo step successivo. È possibile calcolare un lower bound del numero di iterazioni k necessarie a far scendere l'errore sotto m, come segue:

$$k \geq -\frac{m}{\log_{10} \rho(B)} \tag{33}$$

Dove B è la matrice del metodo. Sia R il **fattore di convergenza** del metodo iterativo, definito come segue:

$$R = \log_{10} \rho(B) \tag{34}$$

Allora si avrà:

$$k \geq -\frac{m}{R} \tag{35}$$

Più grande è R, più la convergenza sarà veloce. Da questo deriva che più $\rho(B)$ è vicino ad 1, più l'intero denominatore si annulla e più le esecuzioni necessarie crescono. Se $0 < \rho(B) < 1$ il logaritmo assumerà un valore negativo che è più grande tanto più è vicino allo zero.

Metodo di Jacobi

Partendo da una soluzione iniziale $x^{(0)}$, la soluzione $x^{(k)}$ si determina dalla soluzione $x^{(k-1)}$ come segue:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{13}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{cases}, \qquad a_{ii} \neq 0, \forall i \qquad \begin{cases} x_1^{(k+1)} = \dfrac{b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)}}{a_{11}} \\ x_2^{(k+1)} = \dfrac{b_2 - a_{21}x_1^{(k)} - a_{23}x_3^{(k)}}{a_{22}} \\ x_3^{(k+1)} = \dfrac{b_3 - a_{31}x_1^{(k)} - a_{32}x_2^{(k)}}{a_{33}} \end{cases}$$

Condizione sufficiente di convergenza per Jacobi

Sia $Ax = b$ il sistema da risolvere, condizione sufficiente affinché Jacobi converga è che A sia strettamente diagonalmente dominante.

Formulazione matriciale

La decomposizione presa in considerazione nel metodo è la seguente:

$$A = D - E - F \tag{36}$$

Se si pone $M = D$ ed $N = E + F$ allora si ottiene la decomposizione classica:

$$A = M - N \tag{37}$$

Le tre matrici rappresentano:

- D la diagonale principale di A
- E gli elementi al di sotto della diagonale principale di A, cambiati di segno.
- F gli elementi al di sopra della diagonale principale di A, cambiati di segno.

Dalle matrici, ritroviamo l'equazione di aggiornamento del metodo di Jacobi:

$$\begin{aligned} Ax &= b \\ (D - E - F)x &= b \\ Dx - (E + F)x &= b \\ Dx &= (E + F)x + b \\ x &= D^{-1}(E + F)x + D^{-1}b \end{aligned} \tag{38}$$

Applichiamo gli indici all'ultimo passaggio per indicare l'aggiornamento:

$$x^{(k+1)} = D^{-1}(E + F)x^{(k)} + D^{-1}b \tag{39}$$

> Ricordiamo che l'inversa di una matrice diagonale è pari al reciproco dei singoli elementi non nulli.

La matrice del metodo è $D^{-1}(E + F)$ e studiandola è possibile studiare la convergenza.

Metodo di Gauss-Seidel (GS)

È una variante del metodo di Jacobi che sfrutta il calcolo delle componenti precedenti della soluzione corrente per calcolare le successive. Si vede empiricamente che il metodo converge spesso più velocemente rispetto a quello di Jacobi (quindi il raggio spettrale di GS è minore). Nell'esempio sottostante, gli apici rossi sono quelli di Gauss-Seidel, mentre quelli neri sono di Jacobi.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{13}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{cases}, \quad a_{ii} \neq 0, \forall i \quad \begin{cases} x_1^{(k+1)} = \dfrac{b_1 - a_{12}x_2^{(k)} - a_{13}x_3^{(k)}}{a_{11}} \\ x_2^{(k+1)} = \dfrac{b_2 - a_{21}x_1^{(k)(k+1)} - a_{23}x_3^{(k)}}{a_{22}} \\ x_3^{(k+1)} = \dfrac{b_3 - a_{31}x_1^{(k)(k+1)} - a_{32}x_2^{(k)(k+1)}}{a_{33}} \end{cases}$$

Condizione sufficiente di convergenza per GS

Sia $Ax = b$ il sistema da risolvere, condizione sufficiente affinché Jacobi converga è che A sia strettamente diagonalmente dominante o che A sia simmetrica definita positiva.

Formulazione matriciale

La formulazione è analoga a quella di Jacobi, ma viene isolata la matrice $(D - E)$ anziché isolare solo la matrice D. Se volessimo affibiare le matrici alla trattazione generale, avremmo che $M = (D - E)$ e $N = F$. Vediamo i passaggi:

$$\begin{aligned} Ax &= b \\ (D - E - F)x &= b \\ (D - E)x - Fx &= b \\ (D - E)x &= Fx + b \\ x &= (D - E)^{-1}Fx + (D - E)^{-1}b \end{aligned} \tag{40}$$

Supponendo l'esistenza dell'inversa. Applichiamo gli indici all'ultimo passaggio per indicare l'aggiornamento:

$$x^{(k+1)} = (D - E)^{-1}Fx^{(k)} + (D - E)^{-1}b \tag{41}$$

La matrice del metodo in questo caso è $(D - E)^{-1}F$ e studiandola è possibile studiare la convergenza.

Metodo Successive Over-Relaxation (SOR)

SOR è un metodo basato su GS che applica un rilassamento al calcolo della soluzione. Si fissa un certo parametro ω e si calcola la soluzione utilizzata come la media pesata tra la soluzione di GS e la soluzione precedente:

$$x^{(k+1)} = \omega x^{(k+1)} + (1 - \omega)x^{(k)} \tag{42}$$

Formulazione matriciale

Ricaviamo l'equazione di aggiornamento del metodo, partendo dal sistema lineare:

$$\begin{aligned} Ax &= b \\ \omega Ax &= \omega b \\ \omega(D-E-F)x &= \omega b \\ Dx + \omega(D-E-F)x &= Dx + \omega b \\ Dx + \omega Dx - \omega Ex - \omega Fx &= Dx + \omega b \\ (D-\omega E)x &= \omega Fx + Dx - \omega Dx + \omega b \\ (D-\omega E)x &= [\omega F + (1-\omega)D]x + \omega b \\ x &= (D-\omega E)^{-1}[\omega F + (1-\omega)D]x + (D-\omega E)^{-1}\omega b \end{aligned} \tag{43}$$

Introduciamo la matrice $H(w) = (D-\omega E)^{-1}[\omega F + (1-\omega)D]$, sostituiamo nell'ultima equazione, e introduciamo gli apici del metodo iterativo:

$$x^{(k+1)} = H(\omega)x^{(k)} + (D-\omega E)^{-1}\omega b \tag{44}$$

Poniamo:

$$\begin{aligned} L &= D^{-1}E \\ R &= D^{-1}F \end{aligned} \tag{45}$$

E manipoliamo la matrice $H(\omega)$ come segue:

$$\begin{aligned} H(\omega) &= (D-\omega E)^{-1}[(1-\omega)D + \omega F] \\ &= [D(I-\omega D^{-1}E)]^{-1}D[(1-\omega)I + \omega D^{-1}F] \\ &= [D(I-\omega L)]^{-1}D[(1-\omega)I + \omega R] \\ &= (I-\omega L)^{-1}D^{-1}D[(1-\omega)I + \omega R] \\ &= (I-\omega L)^{-1}[(1-\omega)I + \omega R] \end{aligned} \tag{46}$$

La matrice finale è data da:

$$H(\omega) = (I-\omega L)^{-1}[(1-\omega)I + \omega R] \tag{47}$$

E studiando $\rho(H(\omega))$ non solo è possibile studiare la convergenza del metodo, ma dato il parametro ω arbitrario sarà possibile aggiustare la matrice per manipolare la sua convergenza.

Teorema di condizionamento del parametro di SOR

Si ha la seguente relazione sempre verificata

$$\rho(H(\omega)) \geq |\omega - 1| \quad \forall \omega \in \mathbb{R} \tag{48}$$

Da cui osserviamo che:

$$\begin{cases} \text{SOR diverge} & \omega \leq 0 \vee \omega \geq 2 \\ \text{SOR converge} & 0 < \omega < 2 \end{cases} \tag{49}$$

Fissato ω nel range convergente, è possibile muoversi all'interno del range di convergenza per tarare l'algoritmo e renderlo più rapido.

Variante SSOR

Symmetric SOR (SSOR) è una variante di SOR in cui viene preservata la possibile simmetria della matrice.

Metodo di discesa del gradiente

Quando un sistema $Ax = b$ è molto grande, è possibile risolverlo riconducendolo ad un problema di minimizzazione. Si considera la forma quadratica del problema:

$$\phi(x) = \frac{1}{2}x^T Ax + x^T b \tag{50}$$

Si calcola il gradiente della funzione:

$$\Delta\phi = \left(\frac{\partial\phi}{\partial x_1}, \ldots, \frac{\partial\phi}{\partial x_n} \right) \tag{51}$$

Utilizziamo il l'opposto del gradiente come direzione di discesa:

$$d^{(k)} = -\Delta\phi(x^{(k)}) \tag{52}$$

Sia a_k la lunghezza del passo di aggiornamento all'k-esimo passo, aggiorniamo la soluzione:

$$x^{(k+1)} = x^{(k)} + a_k d^{(k)} \tag{53}$$