Una Libreria di Algebra Lineare per il Calcolo Scientifico

Sandro Tosi

Introduzione
Il Lavoro di Tesi

Il Metodo di Wassing
Introduzione al Metodo
Descrizione Teorica del Metodo
Ridurre l'Occupazione di Memoria
Metodo di Memorizzazione degli Elementi

Sperimentazione e Risultati
Risultati Attesi
Verifica delle Performance
Verifica dell'Accuratezza

Conclusioni

Introduzione

- La risoluzione di sistemi lineari ormai è una pratica molto comune
- Sistemi di grandi dimensioni, però, possono creare problemi di occupazione di memoria

Il Lavoro di Tesi

Lo scopo del lavoro di tesi è stato lo sviluppo di una libreria di software matematico

- in Fortran e C
- in aritmetica reale e complessa
- in singola e doppia precisione

per la risoluzione efficiente, dal punto di vista dell'occupazione di memoria, di sistemi densi di equazioni lineari, tramite il *metodo di Wassing*.

Il Metodo di Wassing

Questo metodo consente di

- risolvere sistemi lineari
- utilizzare $\frac{1}{4}$ della memoria necessaria a metodi classici (Gauss)

Questo guadagno di occupazione di memoria è ottenuto tramite

- il non richiedere la memorizzazione esplicita della matrice dei coefficienti
- un metodo opportuno di memorizzazione degli elementi

Descrizione Teorica del Metodo

Il problema da risolvere è

$$Ax = b$$

Il metodo di Wassing costruisce una nuova matrice:

$$Z = \left[\begin{array}{c} A^T \\ \hline -b^T \end{array}\right]$$

che viene fattorizzata LU

$$Z = \left[\begin{array}{c} A^T \\ \hline -b^T \end{array}\right] = \left[\begin{array}{c|c} L & \underline{0} \\ \hline \ell^T & 1 \end{array}\right] \cdot \left[\begin{array}{c} U \\ \hline \underline{0}^T \end{array}\right] \equiv \hat{L}\hat{U}$$

da cui

$$x^T = -\ell^T L^{-1}$$

Descrizione Teorica del Metodo

Noto che la soluzione è $x^T = -\ell^T L^{-1}$

si osservi che

$$\hat{L} = \left[\begin{array}{c|c} L & \underline{0} \\ \hline \ell^T & 1 \end{array}\right] \quad \Rightarrow \quad \hat{L}^{-1} = \left[\begin{array}{c|c} L^{-1} & \underline{0} \\ \hline -\ell^T L^{-1} & 1 \end{array}\right]$$

I primi n elementi dell'ultima riga di $\hat{L}$ contengono la soluzione x^T

È quindi possibile calcolare solo la matrice $\hat{L}$, e non anche $\hat{U}$, per ottenere la soluzione al nostro problema iniziale.

Descrizione Teorica del Metodo

I passi che devono essere svolti per risolvere il sistema lineare sono i seguenti:

1. $\forall i = 2, 3, \ldots, n, n+1$
   $$z_{i1} = -z_{i1}/z_{11}$$
2. $\forall k = 2, 3, \ldots, n,$
   2.1 $\forall i = k, k+1, \ldots, n, n+1$
   $$z_{ik} = z_{ik} + \sum_{j=1}^{k-1} z_{ij} z_{jk}$$
   2.2 $\forall i = k+1, k+2, \ldots, n, n+1$
   $$z_{ik} = -z_{ik}/z_{kk}$$
   2.3 $\forall j = 1, 2, \ldots, k-1$ e $\forall i = k+1, k+2, \ldots, n, n+1$
   $$z_{ij} = z_{ij} + z_{ik} z_{kj}$$
3. Il vettore soluzione, sarà
   $$x_j = z_{n+1,j} \; \forall j = 1, 2, \ldots, n$$

Descrizione Teorica del Metodo

Il metodo può essere facilmente generalizzato al caso di più termini noti

$$AX = B$$

con A $n \times n$ e X, B $n \times m$ costruendo la matrice Z come

$$Z = \left[\frac{A^T}{-B^T} \right]$$

Inoltre, la complessità risulta di $\mathcal{O}(\frac{2}{3}n^3)$, uguale, cioè, al metodo di Gauss

Ridurre l'Occupazione di Memoria

Vediamo ora come sono organizzati i dati necessari all'applicazione del metodo di Wassing.

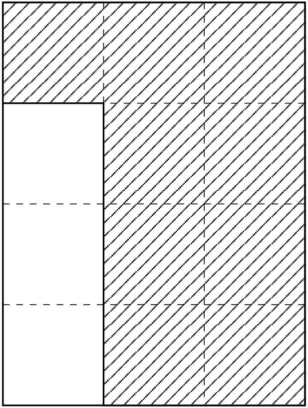
Verrà presentato un esempio di un sistema lineare 3×3, mostrando quali elementi della matrice Z sono necessari nei vari passi.

Nei grafici vengono usate le convenzioni che

- le aree evidenziate indicano degli elementi che non sono mai stati utilizzati
- gli spazi barrati, invece, indicano elementi non più necessari nel proseguio dell'algoritmo

Contenuto della matrice Z al passo 1

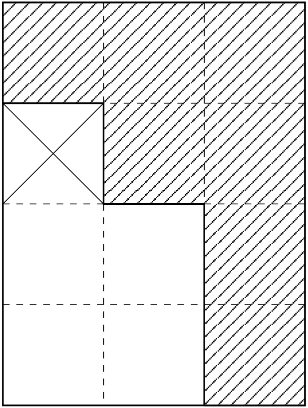
Al termine del passo 1 si ha bisogno di memorizzare solo gli elementi sottodiagonali della prima colonna della matrice Z



Contenuto della matrice Z al passo 2

Al termine del passo 2 si devono memorizzare gli elementi sottodiagonali della seconda colonna e gli elementi omologhi sulla prima colonna

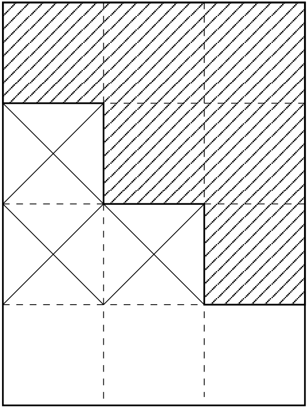
un elemento precedentemente memorizzato non sarà più utilizzato



Contenuto della matrice Z al passo 3

Al termine del terzo ed ultimo passo, devono essere memorizzati solo gli elementi sull'ultima riga, la quale contiene il vettore soluzione x^T

altri 2 elementi non verranno più utilizzati



Ridurre l'Occupazione di Memoria

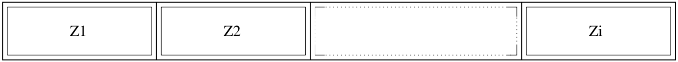
In precedenza, si è evidenziato come alcuni elementi non venissero più utilizzati per il proseguio dell'algoritmo. Allora,

- eliminando questi elementi
- memorizzando solo quelli necessari in un vettore invece che in una matrice

è possibile ridurre l'occupazione di memoria ad $\dfrac{(n+1)^2}{4}$ locazioni di memoria, contro le consuete n^2

Metodo di Memorizzazione degli Elementi

La matrice Z viene memorizzata, per colonne, all'interno di un vettore:



dove, appunto, Zi rappresenta l'i-esima colonna di Z.

In questo modo, le operazioni vengono eseguite su elementi contigui in memoria, e risultano dunque ottimizzate.

Sperimentazione e Risultati

La valutazione di performance ed accuratezza dei codici numerici sviluppati è stata una parte fondamentale del lavoro di tesi:

- è preferibile utilizzare un codice performante
- un codice poco accurato è inutile

Risultati Attesi

Vediamo quali risoltati possiamo attenderci dal metodo di Wassing:

- a causa dei molti spostamenti di dati in memoria, ci si aspetta che le performance siano minori del metodo di Gauss (utilizzato nelle routine LAPACK), sebbene la complessità risulti la stessa
- ci si attende, però, che l'accuratezza del metodo sia confrontabile con quella di altri metodi

Verifica delle Performance

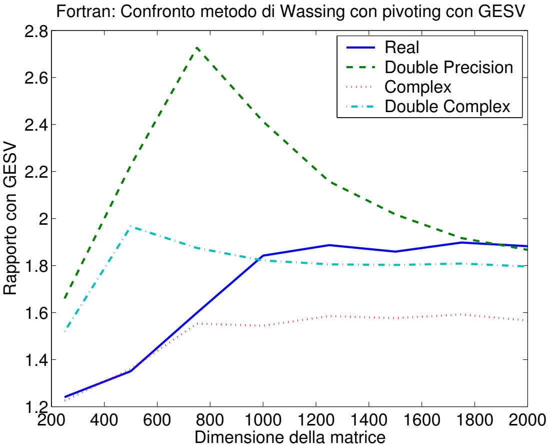
La verifica delle performance è stata eseguita facendo variare

- la dimensione del sistema tra $n = 250$ ed $n = 2000$ con incrementi di 250
- il linguaggio di programmazione
- il tipo di dato

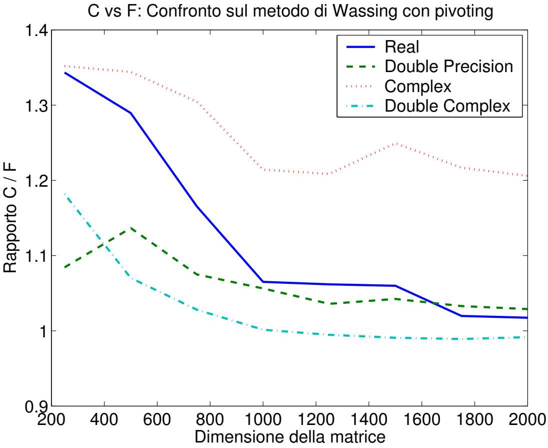
I risultati sono stati confrontati con le routine LAPACK

- `GESV`, per matrici dei coefficienti generali
- `POSV`, per matrici dei coefficienti simmetriche e definite positive

Confronto del Metodo di Wassing con GESV



Confronto tra le Implementazioni Fortran e C



Verifica dell'Accuratezza

Una caratteristica fondamentale di un codice numerico è la sua accuratezza

Risulta, pertanto, necessario stimare l'accuratezza del codice sviluppato

Per far questo si è utilizzato il vettore *errore*:

$$err = x - \hat{x}$$

la differenza tra la soluzione calcolata e quella *esatta*

Stima dell'Accuratezza

Il sistema lineare utilizzato per la verifica dell'accuratezza era tale per cui la soluzione esatta fosse nota per via algebrica e composta da tutti 1

Per stimare l'accuratezza si sono utilizzati due indicatori:

- la norma 2 "pesata"

$$\sqrt{\frac{1}{n}\sum_{i=1}^{n} err_i^2} = \frac{\|err\|_2}{\sqrt{n}}$$

- la norma infinito

$$\max_{i=1,\dots,n} |x_i - \hat{x}_i| = \|err\|_\infty$$

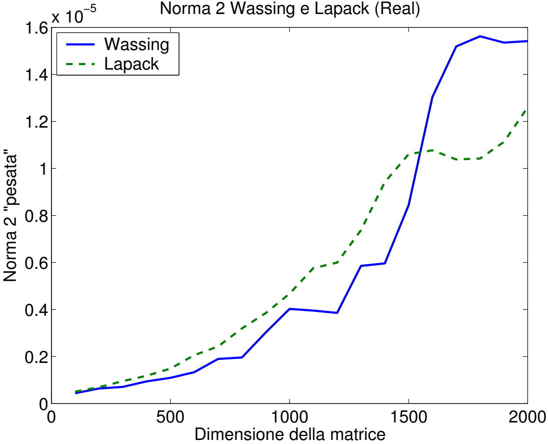
Verifica dell'Accuratezza

La verifica dell'accuratezza è stata eseguita facendo variare

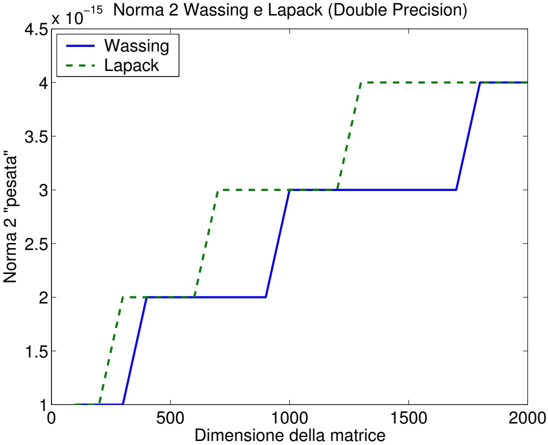
- la dimensione del sistema tra $n = 100$ ed $n = 2000$ con incrementi di 100
- il linguaggio di programmazione
- il tipo di dato

I risultati sono stati confrontati con quelli della routine LAPACK `GESV`

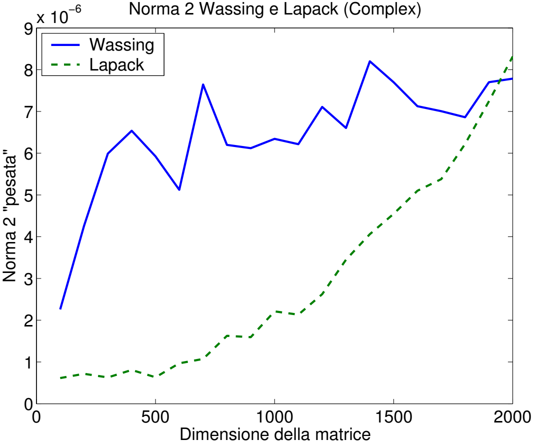
Norma 2 "Pesata"; Numeri Reali in S.P.



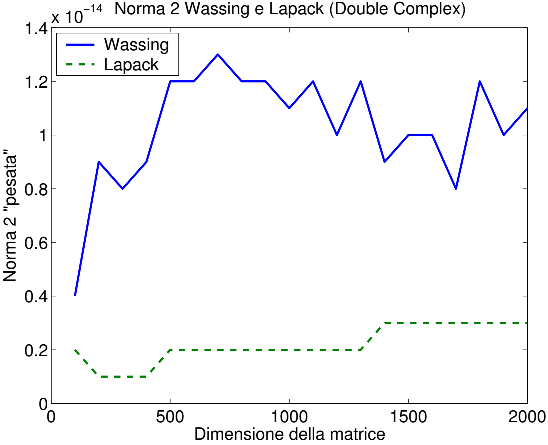
Norma 2 "Pesata"; Numeri Reali in D.P.



Norma 2 "pesata"; Numeri Complessi in S.P.



Norma 2 "pesata"; Numeri Complessi in D.P.



Conclusioni

È possibile diminuire di un fattore 4 l'occupazione di memoria pagando, al massimo, il raddoppio del tempo di esecuzione del codice.

L'accuratezza nel risolvere sistemi lineari, inoltre, rimane paragonabile a quella di metodi classici, come l'eliminazione di Gauss.