

*Dottorato di Ricerca in Ingegneria Industriale*

*Corso: NUMERICAL LINEAR ALGEBRA: TOOLS AND METHODS*

*Docente: PROF. CLAUDE BREZINSKI*

## **PROJECTION METHODS**

*DOTTORANDI*

*Ing. Andrea Carta*

*Ing. Nicola Cau*

*Ing. Simone Pani*

*Ing. Alessandro Serpi*

*Ing. Gian Giuseppe Soma*

# Indice

<i>1. Introduzione</i> .....	3
<i>2. Projection Methods</i> .....	5
<i>2.1 – Metodo del Gradiente</i> .....	6
<i>2.2 – Metodo del Gradiente Coniugato</i> .....	7
<i>2.3 – Metodo del Gradiente Coniugato Precondizionato</i> .....	8
<i>2.4 – Metodo del Residuo Minimo</i> .....	8
<i>2.5 – Metodi di tipo gradiente per sistemi non simmetrici</i> .....	9
<i>3. Simulazioni</i> .....	11
<i>3.1 – Sistemi simmetrici e definiti positivi</i> .....	12
<i>3.2 – Sistemi non simmetrici</i> .....	19
<i>4. Conclusioni</i> .....	25

# 1 – INTRODUZIONE

Assegnato un sistema lineare di ordine  $n$ , questo può essere espresso in forma compatta mediante la seguente relazione:

$$A \cdot x = b \quad (1)$$

nella quale  $A$  rappresenta la matrice del sistema,  $x$  è il vettore soluzione da determinare, mentre  $b$  è il vettore termine noto:

$$A \in \mathbb{R}^{n \times n}, \quad x \in \mathbb{R}^{n \times 1}, \quad b \in \mathbb{R}^{n \times 1}$$

Per determinare il vettore  $x$  è possibile impiegare:

- i *metodi diretti*, i quali consentono di determinare la soluzione in un unico passo con un costo computazionale dell'ordine di  $n^3$ ;
- i *metodi iterativi*, che consentono il raggiungimento della soluzione esatta del sistema in un numero teoricamente infinito di passi, ciascuno dei quali richiede un costo computazionale dell'ordine di  $n^2$ .

Pertanto, quando l'ordine  $n$  del sistema è sufficientemente grande, l'impiego dei metodi iterativi risulta preferibile rispetto a quelli diretti, il cui costo computazionale risulterebbe eccessivamente oneroso; in particolare, nel caso in cui la matrice  $A$  sia sparsa, i vantaggi nell'applicazione dei metodi iterativi risultano ancora più evidenti per effetto della riduzione del costo computazionale di ogni singola iterazione.

Una strategia generale per costruire metodi iterativi lineari consiste nel decomporre la matrice  $A$  nel modo seguente:

$$A = P - N$$

dove  $P$ , definita *matrice di preconditionamento*, non deve essere singolare; l'algoritmo di iterazione risulta dunque espresso dalla seguente relazione:

$$P \cdot x^{(k+1)} = N \cdot x^{(k)} + b \quad (2)$$

Introducendo l'espressione del residuo  $r^{(k)}$ , definito dalla seguente relazione:

$$r^{(k)} = b - A \cdot x^{(k)} \quad (3)$$

l'Eq. (2) può essere riscritta nella seguente forma:

$$x^{(k+1)} = x^{(k)} + P^{-1} \cdot r^{(k)} \quad (4)$$

Di conseguenza, nel caso si decida di utilizzare un metodo iterativo, è necessario definire:

- un *algoritmo di iterazione*, il quale, partendo da un generico vettore iniziale  $x^{(0)}$ , consenta di determinare una successione di vettori  $x^{(k)}$  convergente alla soluzione esatta del sistema:

$$\lim_{k \rightarrow +\infty} x^{(k)} = x$$

- un *criterio di arresto*, il quale prevede generalmente l'interruzione dell'algoritmo una volta che la distanza fra due iterati successivi o la norma del residuo  $r^{(k+1)}$  risultino inferiori ad un limite  $\varepsilon$  prefissato:

$$\begin{aligned} \|x^{(k+1)} - x^{(k)}\| &\leq \varepsilon \\ \|r^{(k)}\| &\leq \varepsilon \end{aligned} \tag{5}$$

Inoltre, il criterio di arresto pone in generale anche una limitazione al numero massimo di iterazioni  $k_{max}$  dell'algoritmo.

## 2 – PROJECTION METHODS

Assegnato il sistema lineare espresso dall'Eq.(1), la seguente relazione:

$$f(u) = (z, A \cdot u - b) = 0$$

definisce un iperpiano  $\mathcal{P}$  nello spazio  $\mathbb{R}^n$  contenente la soluzione  $x$  ricercata; pertanto, considerati due generici vettori  $v_1$  e  $v_2$  appartenenti a  $\mathcal{P}$ , valgono le seguenti relazioni:

$$\begin{cases} (z, A \cdot v_1 - b) = (z, A \cdot v_1) - (z, b) = 0 \\ (z, A \cdot v_2 - b) = (z, A \cdot v_2) - (z, b) = 0 \end{cases}$$

Sottraendo membro a membro le precedenti equazioni:

$$(z, A \cdot (v_1 - v_2)) = (A^T \cdot z, (v_1 - v_2)) = 0 \quad \forall v_1, v_2 \in \mathbf{P}$$

In conclusione:

$$\forall v_1, v_2 \in \mathbf{P} \quad \rightarrow \quad (A^T \cdot z, (v_1 - v_2)) = 0 \quad \Rightarrow \quad (A^T \cdot z) \perp \mathbf{P}$$

Considerato quindi il seguente metodo iterativo:

$$x^{(k+1)} = x^{(k)} + \lambda^{(k)} \cdot A^T \cdot z^{(k)} \quad (6)$$

il valore di  $\lambda^{(k)}$  può essere determinato imponendo che il residuo  $r^{(k+1)}$  sia ortogonale al piano  $\mathcal{P}_k$  mediante la seguente condizione:

$$-(z^{(k)}, r^{(k+1)}) = (z^{(k)}, A \cdot x^{(k+1)} - b) = 0$$

Sviluppando, si ottiene la seguente espressione:

$$\lambda^{(k)} = \frac{(z^{(k)}, r^{(k)})}{(A^T \cdot z^{(k)}, A^T \cdot z^{(k)})} \quad (7)$$

Pertanto, partendo dal generico vettore  $x^{(k)}$ , è necessario eseguire ad ogni passo le seguenti operazioni:

- 1) scegliere una direzione  $z^{(k)}$ ;
- 2) calcolare il residuo  $r^{(k)}$  mediante l'Eq. (3);
- 3) calcolare il valore di  $\lambda^{(k)}$  mediante l'Eq. (7);
- 4) determinare il nuovo vettore  $x^{(k+1)}$  mediante l'Eq. (6).

## 2.1 – METODO DEL GRADIENTE

Se la matrice  $A$  del sistema è simmetrica e definita positiva:

$$A = A^T$$

$$\forall v \in \mathbb{R}^{n \times 1}, v \neq 0, v^T \cdot A \cdot v > 0$$

si consideri il seguente funzionale quadratico:

$$\mathbf{J}(u) = \frac{1}{2} \cdot (u, A \cdot u) - (u, b) \quad (8)$$

Dato che:

$$\nabla \mathbf{J}(u) = A \cdot u - b$$

la soluzione  $x$  del sistema lineare rappresenta un punto di minimo per il funzionale  $\mathcal{J}$ ; infatti, è possibile dimostrare che:

$$\forall u \in \mathbb{R}^{n \times 1} \quad \mathbf{J}(u) = \mathbf{J}(x) + \frac{1}{2} \cdot ((u-x), A \cdot (u-x)) > \mathbf{J}(x)$$

Di conseguenza, scegliendo, come direzione  $A^T z^{(k)}$  nell'Eq. (6), quella individuata dal gradiente di  $\mathcal{J}$ , si ottiene il metodo del *gradiente* o della *steepest descent*:

$$x^{(k+1)} = x^{(k)} + \lambda^{(k)} \cdot r^{(k)} \quad (9)$$

Il valore di  $\lambda^{(k)}$  è determinato imponendo che  $\mathcal{J}$  in  $x^{(k+1)}$  sia minimo; pertanto, utilizzando l'Eq. (8):

$$\mathbf{J}(x^{(k+1)}) = \frac{1}{2} \cdot (x^{(k)}, A \cdot x^{(k)}) - (x^{(k)}, b) - \lambda^{(k)} \cdot (r^{(k)}, r^{(k)}) + \frac{(\lambda^{(k)})^2}{2} \cdot (r^{(k)}, A \cdot r^{(k)})$$

Derivando rispetto a  $\lambda^{(k)}$  ed uguagliando a zero:

$$-(r^{(k)}, r^{(k)}) + \lambda^{(k)} \cdot (r^{(k)}, A \cdot r^{(k)}) = 0 \quad \rightarrow \quad \lambda^{(k)} = \frac{(r^{(k)}, r^{(k)})}{(r^{(k)}, A \cdot r^{(k)})} \quad (10)$$

Inoltre, considerando l'espressione del residuo  $r^{(k+1)}$ , si ottiene la seguente relazione:

$$r^{(k+1)} = b - A \cdot x^{(k+1)} = r^{(k)} - \lambda^{(k)} \cdot A \cdot r^{(k)} \quad (11)$$

Moltiplicando scalarmene per  $r^{(k)}$  e sostituendo il valore di  $\lambda^{(k)}$  fornito dall'Eq. (10), si dimostra che il residuo al passo (k+1) risulta ortogonale al precedente:

$$\left( r^{(k+1)}, r^{(k)} \right) = 0$$

In conclusione, i risultati ottenuti sono riassunti di seguito:

$$\begin{cases} r^{(k)} = b - A \cdot x^{(k)} \\ x^{(k+1)} = x^{(k)} + \lambda^{(k)} \cdot r^{(k)} \end{cases}, \quad \lambda^{(k)} = \frac{\left( r^{(k)}, r^{(k)} \right)}{\left( r^{(k)}, A \cdot r^{(k)} \right)} \quad (12)$$

## 2.2 – METODO DEL GRADIENTE CONIUGATO

La convergenza del metodo del gradiente risulta lenta nel caso in cui vi sia molta differenza fra il massimo ed il minimo autovalore della matrice  $A$ ; pertanto, considerando una generica direzione  $p^{(k)}$ :

$$x^{(k+1)} = x^{(k)} + \lambda^{(k)} \cdot p^{(k)}$$

è possibile scegliere il valore di  $\lambda^{(k)}$  in modo tale da minimizzare il funzionale  $\mathcal{J}$  lungo tale direzione:

$$\left( r^{(k+1)}, p^{(k)} \right) = 0 \quad \rightarrow \quad \lambda^{(k)} = \frac{\left( p^{(k)}, r^{(k)} \right)}{\left( p^{(k)}, A \cdot p^{(k)} \right)} \quad (13)$$

A questo punto, imponendo la seguente condizione:

$$\begin{cases} \left( p^{(k+1)}, A \cdot p^{(k)} \right) = 0 \\ p^{(k+1)} = r^{(k+1)} - \mu^{(k+1)} \cdot p^{(k)} \end{cases} \quad \rightarrow \quad \mu^{(k+1)} = \frac{\left( A \cdot p^{(k)}, r^{(k+1)} \right)}{\left( p^{(k)}, A \cdot p^{(k)} \right)} \quad (14)$$

si ottiene il metodo del *gradiente coniugato* (CG); tale metodo assicura la convergenza in un numero finito di iterazioni (al massimo  $n$ ), pertanto può essere considerato un metodo diretto; tuttavia, esso può essere favorevolmente utilizzato come metodo iterativo mediante l'impiego di opportuni criteri di arresto. Di conseguenza, scegliendo come direzione iniziale  $p^{(0)}$  il valore del residuo  $r^{(0)}$ , il metodo prevede l'applicazione ricorsiva delle seguenti relazioni:

$$\begin{cases} x^{(k+1)} = x^{(k)} + \lambda^{(k)} \cdot p^{(k)} \\ r^{(k+1)} = r^{(k)} - \lambda^{(k)} \cdot A \cdot p^{(k)} \\ p^{(k+1)} = r^{(k+1)} - \mu^{(k+1)} \cdot p^{(k)} \end{cases}, \quad \begin{cases} \lambda^{(k)} = \frac{\left( p^{(k)}, r^{(k)} \right)}{\left( p^{(k)}, A \cdot p^{(k)} \right)} \\ \mu^{(k+1)} = \frac{\left( A \cdot p^{(k)}, r^{(k+1)} \right)}{\left( p^{(k)}, A \cdot p^{(k)} \right)} \end{cases} \quad (15)$$

## 2.3 – METODO DEL GRADIENTE CONIUGATO PRECONDIZIONATO

Per migliorare la convergenza del metodo del gradiente coniugato, è possibile introdurre una matrice  $P$  di preconditionamento, simmetrica e definita positiva; pertanto, definendo il vettore  $z^{(k+1)}$  mediante la seguente relazione:

$$z^{(k+1)} = P^{-1} \cdot r^{(k+1)}$$

le Eq. (15) possono essere modificate come riportato di seguito:

$$\begin{cases} x^{(k+1)} = x^{(k)} + \lambda^{(k)} \cdot p^{(k)} \\ r^{(k+1)} = r^{(k)} - \lambda^{(k)} \cdot A \cdot p^{(k)} \\ p^{(k+1)} = z^{(k+1)} - \mu^{(k+1)} \cdot p^{(k)} \end{cases}, \quad \begin{cases} \lambda^{(k)} = \frac{(p^{(k)}, r^{(k)})}{(p^{(k)}, A \cdot p^{(k)})} \\ \mu^{(k+1)} = \frac{(A \cdot p^{(k)}, z^{(k+1)})}{(p^{(k)}, A \cdot p^{(k)})} \end{cases} \quad (16)$$

ottenendo il metodo del *gradiente coniugato preconditionato* (PCG).

## 2.4 – METODO DEL RESIDUO MINIMO

Combinando opportunamente le Eq. (16), è possibile eliminare la dipendenza dalla direzione  $p^{(k)}$ , ottenendo la seguente relazione:

$$A \cdot r^{(k)} = -\frac{1}{\lambda_k} \cdot r^{(k+1)} + \left( \frac{1}{\lambda_k} - \frac{\mu^{(k)}}{\lambda_{k-1}} \right) \cdot r^{(k)} + \frac{\mu^{(k)}}{\lambda_{k-1}} \cdot r^{(k-1)} \quad (17)$$

Inoltre, è possibile esprimere i coefficienti  $\lambda^{(k)}$  e  $\mu^{(k)}$  nel modo seguente:

$$\lambda^{(k)} = \frac{(r^{(k)}, r^{(k)})}{(p^{(k)}, A \cdot p^{(k)})}, \quad \mu^{(k+1)} = -\frac{(r^{(k+1)}, r^{(k+1)})}{(r^{(k)}, r^{(k)})} \quad (18)$$

Utilizzando quindi le Eq. (17) e (18) ed imponendo la minimizzazione della norma 2 del residuo, si ottiene il metodo del *Minimo Residuo* (MINRES).

## 2.5 – METODI DI TIPO GRADIENTE PER SISTEMI NON SIMMETRICI

Nel caso la matrice  $A$  non sia simmetrica, è possibile costruire il seguente sistema lineare:

$$G \cdot x = c \quad , \quad \begin{cases} G = A^H \cdot A \\ c = A^H \cdot b \end{cases} \quad (19)$$

Applicando al nuovo sistema il metodo del gradiente coniugato, si ottengono le seguenti relazioni:

$$\begin{cases} x^{(k+1)} = x^{(k)} + \lambda^{(k)} \cdot p^{(k)} \\ \rho^{(k+1)} = \rho^{(k)} - \lambda^{(k)} \cdot G \cdot p^{(k)} \\ p^{(k+1)} = \rho^{(k+1)} - \mu^{(k+1)} \cdot p^{(k)} \end{cases}$$

Inoltre, dato che:

$$\rho^{(k)} = c - G \cdot x^{(k)} = A^H \cdot (b - A \cdot x^{(k)}) = A^H \cdot r^{(k)}$$

Sostituendo:

$$\begin{cases} x^{(k+1)} = x^{(k)} + \lambda^{(k)} \cdot p^{(k)} \\ r^{(k+1)} = r^{(k)} - \lambda^{(k)} \cdot A \cdot p^{(k)} \\ p^{(k+1)} = A^H \cdot r^{(k+1)} + \mu^{(k+1)} \cdot p^{(k)} \end{cases} \quad (20)$$

dove i coefficienti  $\lambda^{(k)}$  e  $\mu^{(k+1)}$  possono essere determinati imponendo la minimizzazione della norma 2 del residuo, ottenendo in questo caso il metodo del *gradiente coniugato normale residuo* (CGNR):

$$\begin{cases} x^{(k+1)} = x^{(k)} + \lambda^{(k)} \cdot p^{(k)} \\ r^{(k+1)} = r^{(k)} - \lambda^{(k)} \cdot A \cdot p^{(k)} \\ p^{(k+1)} = A^H \cdot r^{(k+1)} + \mu^{(k+1)} \cdot p^{(k)} \end{cases} \quad , \quad \begin{cases} \lambda^{(k)} = \frac{(A^H \cdot r^{(k)}, A^H \cdot r^{(k)})}{(A \cdot p^{(k)}, A \cdot p^{(k)})} \\ \mu^{(k+1)} = \frac{(A^H \cdot r^{(k+1)}, A^H \cdot r^{(k+1)})}{(A^H \cdot r^{(k)}, A^H \cdot r^{(k)})} \end{cases} \quad (21)$$

Se invece  $\lambda^{(k)}$  e  $\mu^{(k+1)}$  risultano determinati in modo tale da minimizzare la norma 2 dell'errore, si ottiene il metodo del *gradiente coniugato normale errore* (CGNE):

$$\begin{cases} x^{(k+1)} = x^{(k)} + \lambda^{(k)} \cdot p^{(k)} \\ r^{(k+1)} = r^{(k)} - \lambda^{(k)} \cdot A \cdot p^{(k)} \\ p^{(k+1)} = A^H \cdot r^{(k+1)} + \mu^{(k+1)} \cdot p^{(k)} \end{cases} \quad , \quad \begin{cases} \lambda^{(k)} = \frac{(r^{(k)}, r^{(k)})}{(p^{(k)}, p^{(k)})} \\ \mu^{(k+1)} = \frac{(r^{(k+1)}, r^{(k+1)})}{(r^{(k)}, r^{(k)})} \end{cases} \quad (22)$$

Nel caso si costruisca il seguente sistema esteso:

$$\begin{bmatrix} 0 & A \\ A^H & 0 \end{bmatrix} \cdot \begin{bmatrix} \underline{x} \\ \underline{x} \end{bmatrix} = \begin{bmatrix} \underline{b} \\ \underline{b} \end{bmatrix} \quad (23)$$

e si applichi il metodo del gradiente coniugato, si ottengono le seguenti relazioni:

$$\begin{cases} \underline{x}^{(k+1)} = \underline{x}^{(k)} + \lambda^{(k)} \cdot \underline{p}^{(k)} \\ \underline{r}^{(k+1)} = \underline{r}^{(k)} - \lambda^{(k)} \cdot A \cdot \underline{p}^{(k)} \\ \underline{p}^{(k+1)} = \underline{r}^{(k+1)} + \mu^{(k+1)} \cdot \underline{p}^{(k)} \end{cases}, \quad \begin{cases} \underline{x}_0^{(k+1)} = \underline{x}_0^{(k)} + \lambda^{(k)} \cdot \underline{p}_0^{(k)} \\ \underline{f}_0^{(k+1)} = \underline{f}_0^{(k)} - \lambda^{(k)} \cdot A^H \cdot \underline{p}_0^{(k)} \\ \underline{p}_0^{(k+1)} = \underline{p}_0^{(k+1)} + \mu^{(k+1)} \cdot \underline{p}_0^{(k)} \end{cases} \quad (24)$$

Dove i coefficienti  $\lambda^{(k)}$  e  $\mu^{(k+1)}$  assumono i seguenti valori:

$$\lambda^{(k)} = \frac{(\underline{f}_0^{(k)}, \underline{r}^{(k)})}{(\underline{p}_0^{(k)}, A \cdot \underline{p}^{(k)})}, \quad \mu^{(k+1)} = \frac{(\underline{r}^{(k+1)}, \underline{p}_0^{(k+1)})}{(\underline{r}^{(k)}, \underline{p}_0^{(k)})} \quad (25)$$

Le Eq. (24) e (25) definiscono il metodo del *gradiente bi-coniugato* (BiCG).

In alternativa ai metodi precedenti, è possibile ricorrere al metodo del *minimo residuo generalizzato* (GMRES); questo metodo, analogo al CG per sistemi non simmetrici, richiede tuttavia la conoscenza di tutti gli iterati precedenti per calcolare  $\underline{x}^{(k+1)}$ , di conseguenza, superato un certo numero di iterazioni, risulta necessario interrompere il metodo (Truncated GMRES) o farlo ripartire dall'ultimo valore ottenuto (Restarted GMRES).

### 3 – SIMULAZIONI

Utilizzando il software MATLAB, sono state effettuate diverse simulazioni al calcolatore allo scopo di confrontare alcuni dei metodi iterativi precedentemente citati impiegandoli nella risoluzione di sistemi lineari generati casualmente. In particolare, sono stati assunti come parametri di confronto:

- il numero di iterazioni *iter* necessario per determinare la soluzione del sistema con la precisione *tol* assegnata;
- il tempo *time\_iter* impiegato da ciascun metodo per eseguire la singola iterazione;
- il tempo totale *time\_tot* impiegato per eseguire l'intero algoritmo.

Pertanto, una volta fissato l'ordine *n* del sistema, si è provveduto a generare una matrice casuale *A* (eventualmente simmetrica e definita positiva), inoltre si è scelto di utilizzare come soluzione un vettore *x* noto a priori, avente le seguenti caratteristiche:

$$x_i = 1 \quad (i = 1..n)$$

Di conseguenza, il vettore termine noto *b* è stato calcolato semplicemente utilizzando l'Eq. (1). Una volta definito il sistema, sono stati fissati i parametri relativi al criterio di arresto di ciascun metodo, ossia il grado di precisione *tol* ed il numero massimo di iterazioni *k<sub>max</sub>*. Le simulazioni effettuate hanno riguardato i seguenti aspetti:

- valutazione dell'effetto del preconditionamento sulla convergenza dei metodi applicati;
- influenza del grado di precisione *tol* richiesto e dell'ordine *n* del sistema sui metodi.

In conclusione, sono stati confrontati fra loro:

- i metodi CG, PCG e MINRES applicati a sistemi simmetrici e definiti positivi;
- i metodi BiCG e GMRES applicati a sistemi qualsiasi.

### 3.1 – SISTEMI SIMMETRICI E DEFINITI POSITIVI

Per generare sistemi simmetrici e definiti positivi è stato utilizzato il seguente comando:

```
A = sprandsym(n,delta,rc,1)
```

In questo modo, è stato possibile generare delle matrici quadrate di ordine  $n$  aventi le seguenti caratteristiche:

- sparse, di densità  $\delta$ ;
- simmetriche;
- definite positive (1);
- avente numero di condizionamento pari al reciproco di  $rc$ .

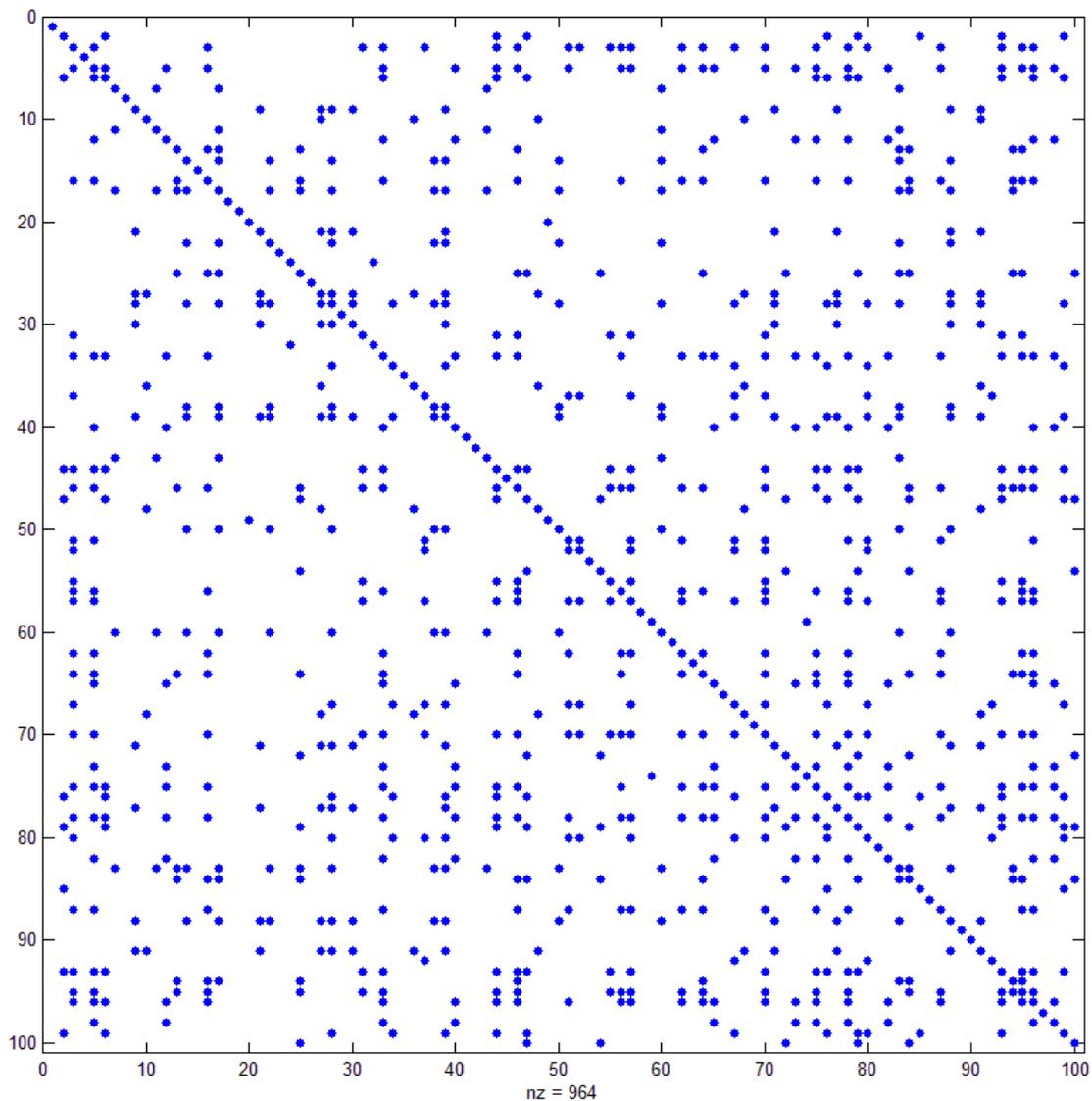


Fig. 1. Rappresentazione grafica della matrice A generata mediante il comando `sprandsym`

Successivamente, dopo aver definito il vettore soluzione  $x$  ed il termine noto  $b$  come illustrato in precedenza, per applicare il metodo CG è stato utilizzato il seguente comando:

```
[y_cg flag_cg relres_cg iter_cg resvec_cg] = pcg(A,b,tol,k_MAX,[],[],x0)
```

Per quanto riguarda gli ingressi:

- $A$  e  $b$  rappresentano rispettivamente la matrice ed il termine noto del sistema lineare;
- $tol$  (precisione desiderata) e  $k\_max$  (numero massimo di iterazioni) definiscono il criterio di arresto del metodo;
- $x0$  è il vettore iniziale.

La funzione *pcg* restituisce:

- $y\_cg$ , che rappresenta il vettore soluzione finale ottenuto dall'algoritmo;
- $flag\_cg$ , ossia un indice dell'efficacia del metodo;
- $relres\_cg$ , ossia il valore finale del residuo;
- $iter\_cg$ , che indica il numero di iterazioni effettuate;
- $resvec\_cg$ , ossia il vettore le cui componenti rappresentano le norme del residuo in corrispondenza di ogni iterazione (in seguito tale parametro non sarà considerato).

Per applicare il metodo PCG, è necessario definire la matrice di preconditionamento: date le caratteristiche del sistema (simmetrico, definito positivo), si è scelto di utilizzare come matrici di preconditionamento quelle ottenute mediante la fattorizzazione incompleta di Cholesky della matrice  $A$ :

```
R = cholinc(A,eps)
```

Infatti, calcolando la fattorizzazione completa di  $A$ , si potrebbe risolvere il sistema in un'unica iterazione, utilizzando di fatto un metodo diretto, il cui costo computazionale risulterebbe però eccessivamente elevato; per questo motivo, è stata introdotta una soglia *eps* che consente di ridurre il costo computazionale della fattorizzazione a scapito della precisione. In conclusione, il comando utilizzato per implementare il metodo PCG risulta:

```
[y_pcg flag_pcg relres_pcg iter_pcg] = pcg(A,b,tol,k_MAX,R.',R,x0)
```

Per quanto riguarda il metodo MINRES, i comandi utilizzati sono riportati di seguito:

```
[y_minres flag_minres relres_minres iter_minres] = minres(A,b,tol,k_MAX,[],[],x0)
```

```
[y_pminres flag_pminres relres_pminres iter_pminres] = minres(A,b,tol,k_MAX,R.',R,x0)
```

La prima simulazione è stata effettuata allo scopo di evidenziare l'influenza del preconditionamento sulla rapidità di convergenza dei metodi iterativi; pertanto, fissati i seguenti parametri:

`n = 100`      `rc = 0.01`      `tol = 1e-12`      `k_MAX = 500`      `x0 = zeros(n,1)`

sono stati confrontati i risultati ottenuti dai metodi PCG e MINRES:

- in assenza di preconditionamento;
- utilizzando come preconditionamento la fattorizzazione incompleta di Cholesky della matrice  $A$  per diversi valori di  $eps$ .

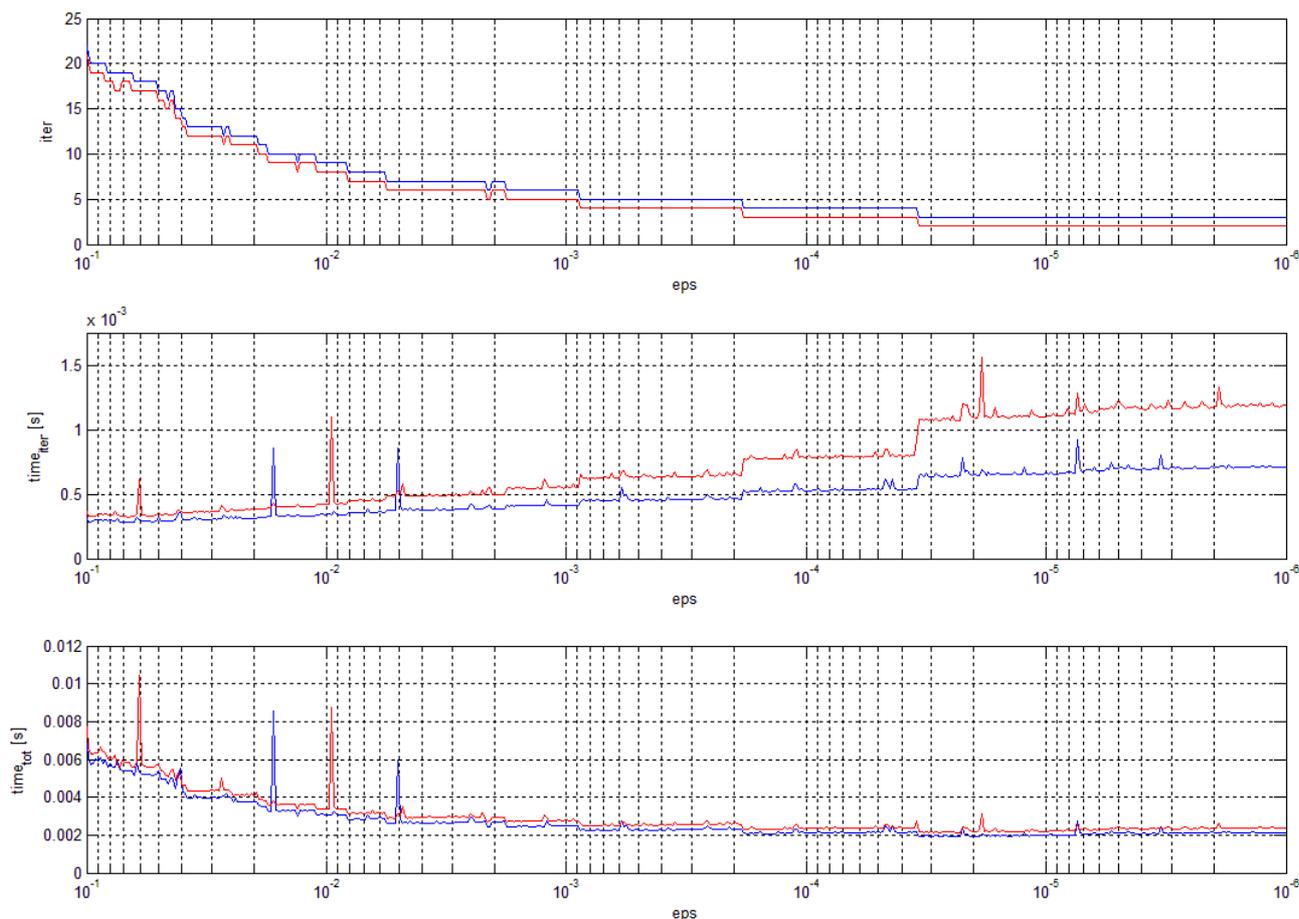


Fig. 2. Andamento dei parametri di controllo al variare di  $eps$  per i metodi PCG e MINRES

I risultati ottenuti hanno evidenziato come l'introduzione del preconditionamento consenta un notevole abbattimento dei tempi di

Tab. 1		PCG	MINRES
Not conditioned	time_tot [s]	0.0608	0.0453
	iter	97	95
Cholesky Factorization	time_tot [s]	0.0176	0.0342
	iter	1	1

calcolo; in particolare, facendo riferimento a dati riportati in Tab. 1, è possibile notare come, in assenza di preconditionamento, il numero di iterazioni effettuate dai due algoritmi sia molto elevato, mentre utilizzando la fattorizzazione completa di Cholesky i costi computazionali risultano decisamente inferiori. Inoltre, utilizzando una fattorizzazione incompleta con grado di precisione  $eps$ , è possibile ottenere tempi di calcolo ulteriormente inferiori, come riportato in Fig. 2. Tale figura mostra inoltre come ridurre  $eps$  al di sotto di  $10^{-3}$  non comporti significative variazioni dei tempi di calcolo complessivi.

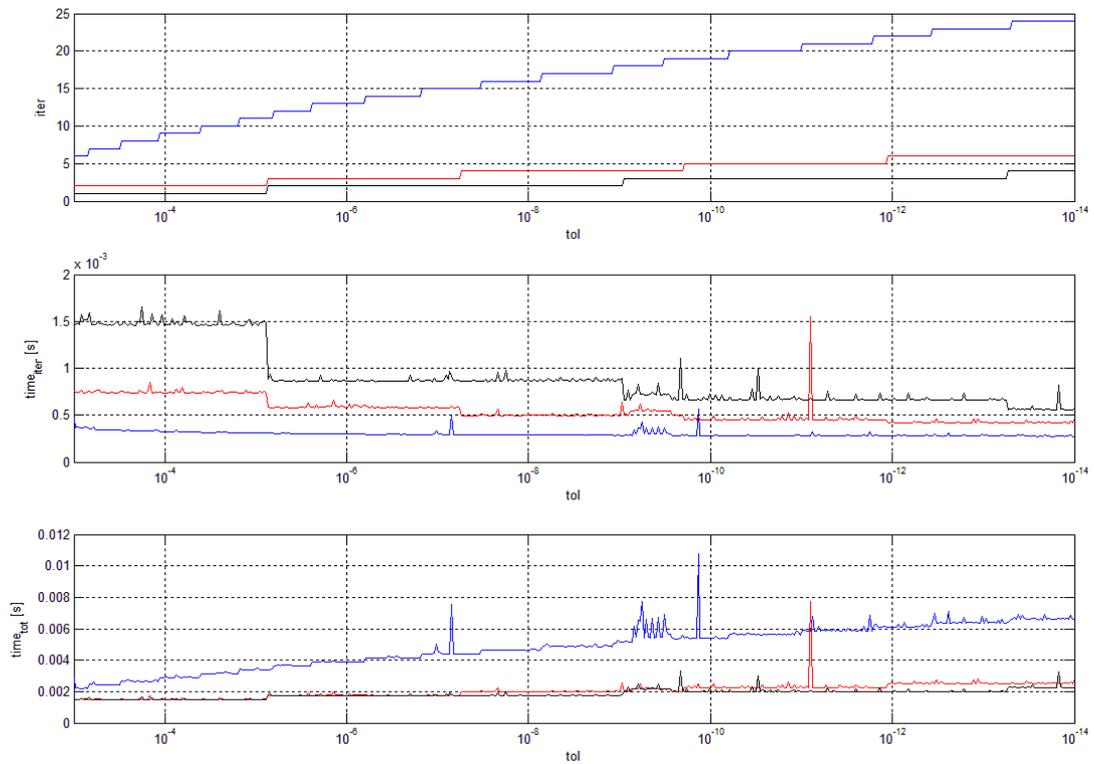


Fig. 3. Andamento dei parametri di controllo al variare di  $tol$  per il metodo PCG con  $eps = 10^{-1}$ ,  $eps = 10^{-3}$  e  $eps = 10^{-5}$

Nella seconda simulazione sono state confrontate le prestazioni fornite dai metodi PCG e MINRES al variare del grado di precisione  $tol$  della soluzione; facendo riferimento alle Fig. 3 e 4, si nota come utilizzando un valore di  $eps$  troppo alto il numero di iterazioni ed il costo computazionale complessivo risultino molto elevati; per contro, utilizzare valori di  $eps$  troppo piccoli non comporta una significativa riduzione del costo computazionale complessivo.

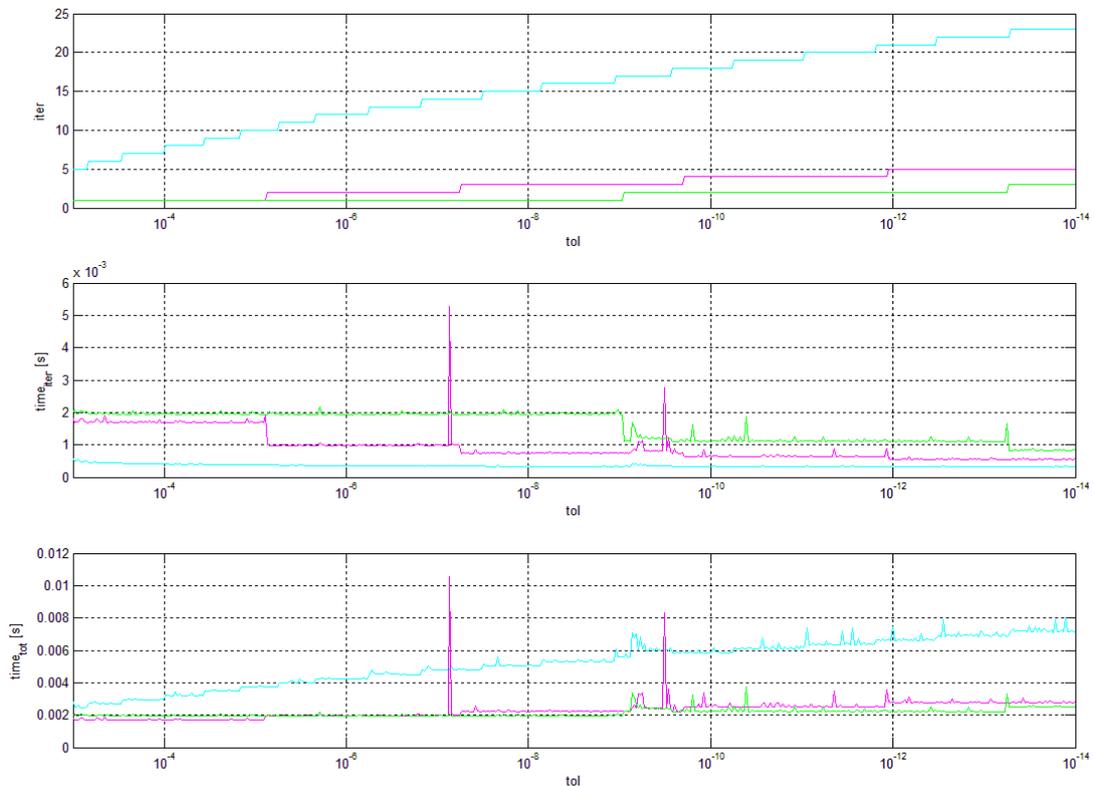


Fig. 4. Andamento dei parametri di controllo al variare di  $tol$  per il metodo MINRES con  $eps=10^{-1}$ ,  $eps=10^{-3}$  e  $eps=10^{-5}$

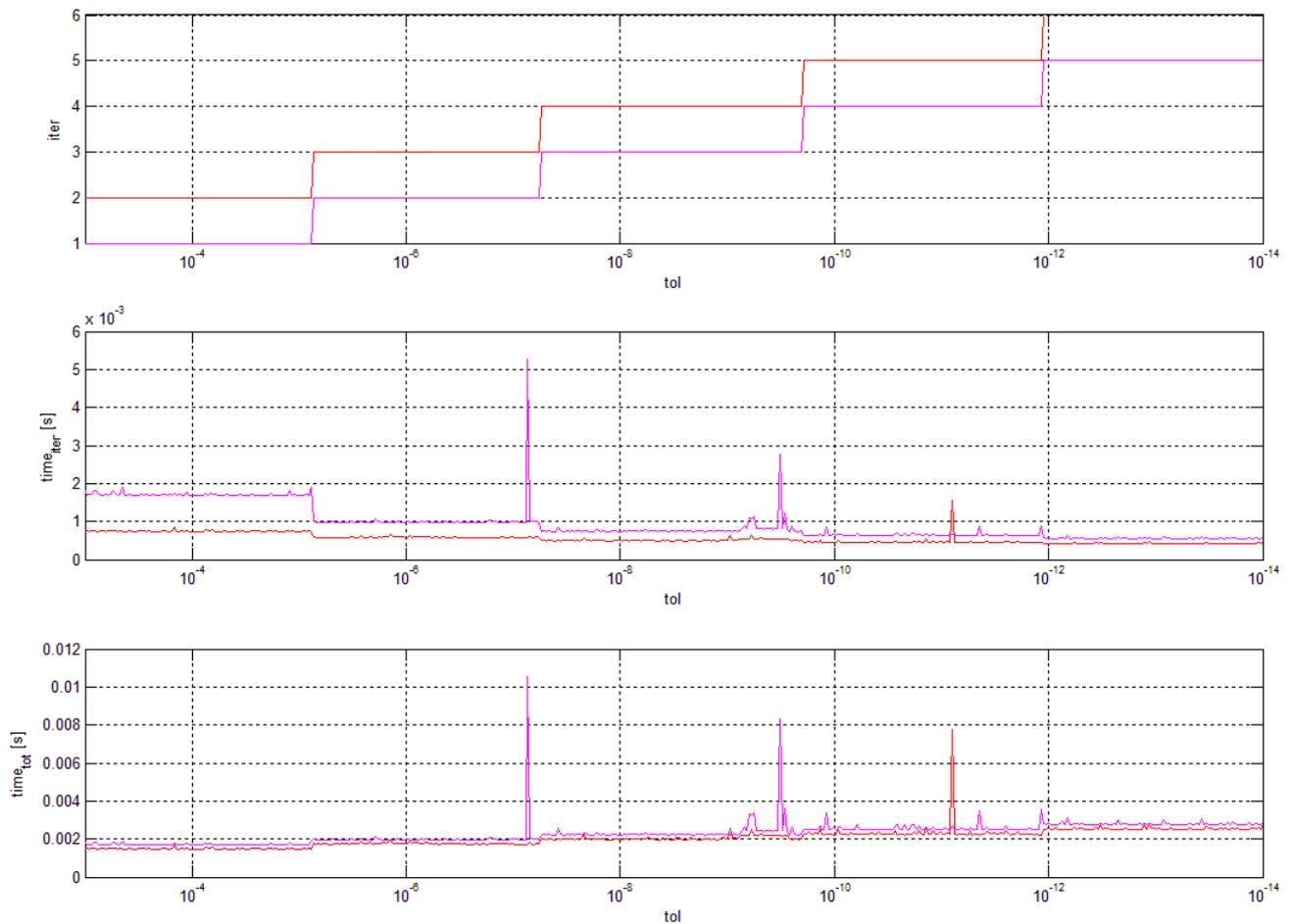


Fig. 5. Confronto fra i parametri di controllo al variare di  $tol$  per i metodi PCG e MINRES con  $eps = 10^{-3}$

Confrontando quindi i risultati ottenuti dai due algoritmi per il valore intermedio di  $eps$ , riportati in Fig. 5, si nota come mediante il metodo MINRES si ottengano un minor numero di iterazioni rispetto al metodo PCG, ma con un costo computazionale complessivo leggermente superiore.

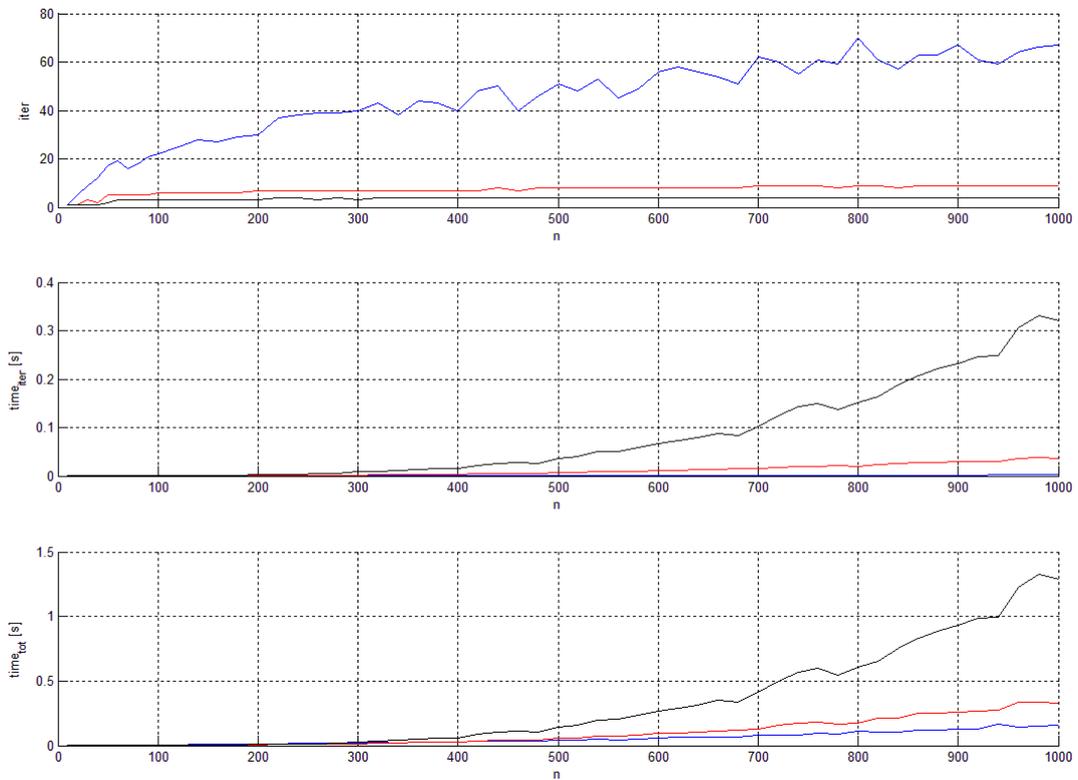


Fig. 6. Andamento dei parametri di controllo al variare di  $n$  per il metodo PCG con  $\text{eps} = 10^{-1}$ ,  $\text{eps} = 10^{-3}$  e  $\text{eps} = 10^{-5}$

In conclusione, la terza simulazione è stata effettuata allo scopo di verificare l'efficacia degli algoritmi al variare dell'ordine  $n$  del sistema. Dalle Fig. 6 e 7 è possibile notare come l'impiego di un valore di  $\text{eps}$  eccessivamente piccolo comporti, per elevati valori di  $n$ , costi computazionali complessivi molto elevati.

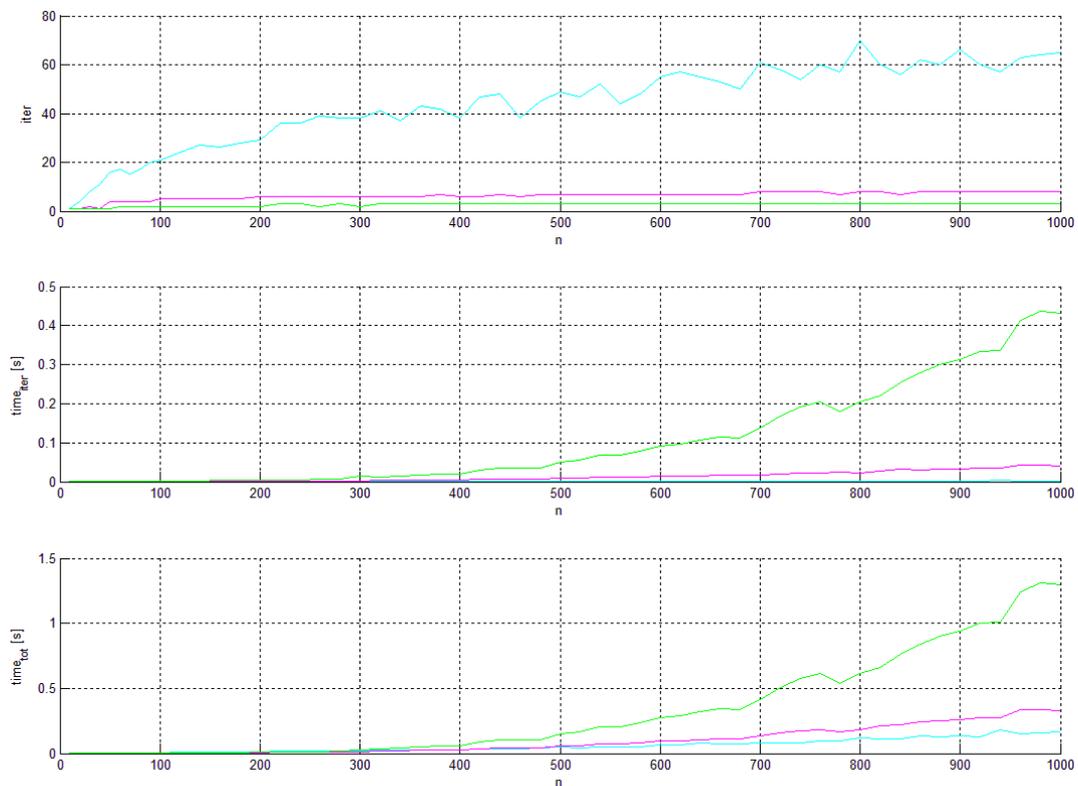


Fig. 7. Andamento dei parametri di controllo al variare di  $n$  per il metodo MINRES con  $\text{eps}=10^{-1}$ ,  $\text{eps}=10^{-3}$  e  $\text{eps}=10^{-5}$

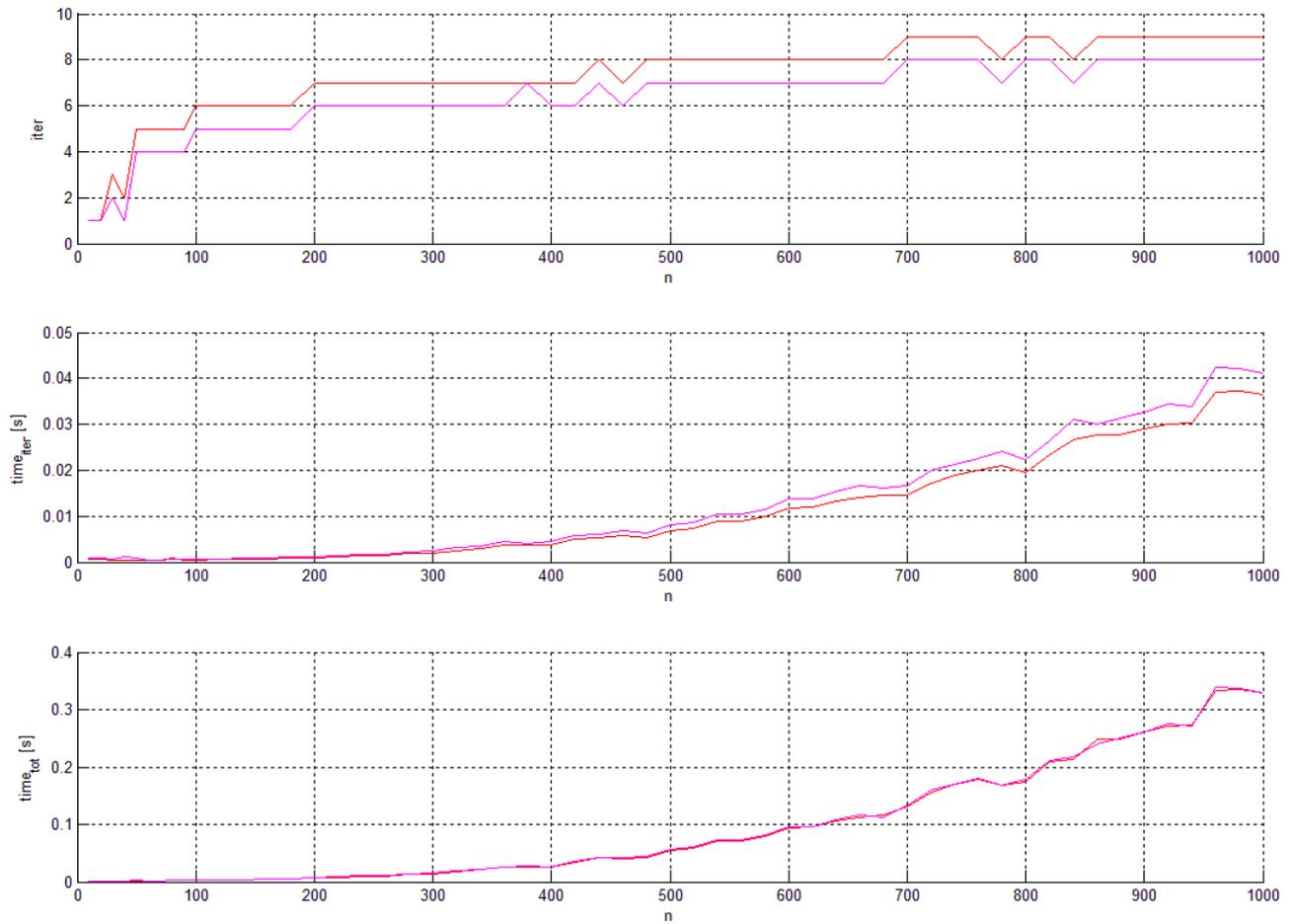


Fig. 8. Confronto fra i parametri di controllo al variare di  $n$  per i metodi **PCG** e **MINRES** con  $\text{eps} = 10^{-3}$

Infine, dal confronto fra i due algoritmi per il valore intermedio di  $\text{eps}$ , non sono emerse differenze significative, come riportato in Fig. 8.

### 3.1 – SISTEMI NON SIMMETRICI

Per generare sistemi non simmetrici è stato utilizzato il seguente comando:

```
A = sprand(n,n,delta,rc)
```

In questo modo, è stato possibile generare delle matrici sparse di ordine  $n$ , di densità  $delta$  ed aventi numero di condizionamento pari al reciproco di  $rc$ .

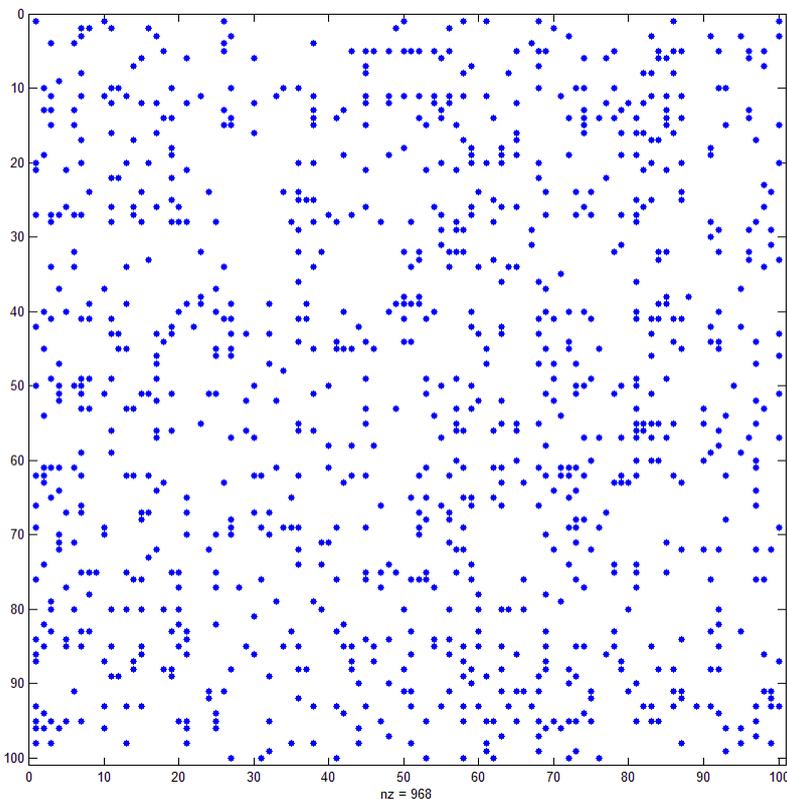


Fig. 9. Rappresentazione grafica della matrice A generata mediante il comando `sprand`

Successivamente, dopo aver definito il vettore soluzione  $x$  ed il termine noto  $b$  come precedentemente descritto, per applicare il metodo BiCG è stato utilizzato il seguente comando:

```
[y_bicg flag_bicg relres_bicg iter_bicg resvec_bicg] = bicg(A,b,tol,k_MAX,[],[],x0)
```

Per quanto riguarda il significato delle variabili ingresso-uscita, valgono le stesse considerazioni fatte per il metodo CG nel precedente paragrafo. Inoltre, date le caratteristiche del sistema (non simmetrico), si è scelto di impiegare come matrici di preconditionamento quelle ottenute mediante la fattorizzazione LU incompleta della matrice A:

```
[L,U] = luinc(A,eps)
```

```
[y_pbicg flag_pbicg relres_pbicg iter_pbicg] = bicg(A,b,tol,k_MAX,L,U,x0)
```

Per quanto riguarda il metodo GMRES, i comandi utilizzati risultano:

```
[y_gmres flag_gmres relres_gmres iter_gmres] = gmres(A,b,[],tol,n,[],[],x0);
```

```
[y_pgmres flag_pgmres relres_pgmres iter_pgmres] = gmres(A,b,[],tol,n,L,U,x0);
```

La prima simulazione è stata effettuata allo scopo di evidenziare anche in questo caso l'influenza del preconditionamento sulla rapidità di convergenza dei metodi iterativi; pertanto, fissati i seguenti parametri:

$n = 100$        $rc = 0.01$        $tol = 1e-12$        $k\_MAX = 500$        $x0 = zeros(n,1)$

sono stati confrontati i risultati ottenuti dai metodi BiCG e GMRES:

- in assenza di preconditionamento;
- utilizzando come preconditionamento la fattorizzazione incompleta LU della matrice  $A$  per diversi valori di  $eps$ .

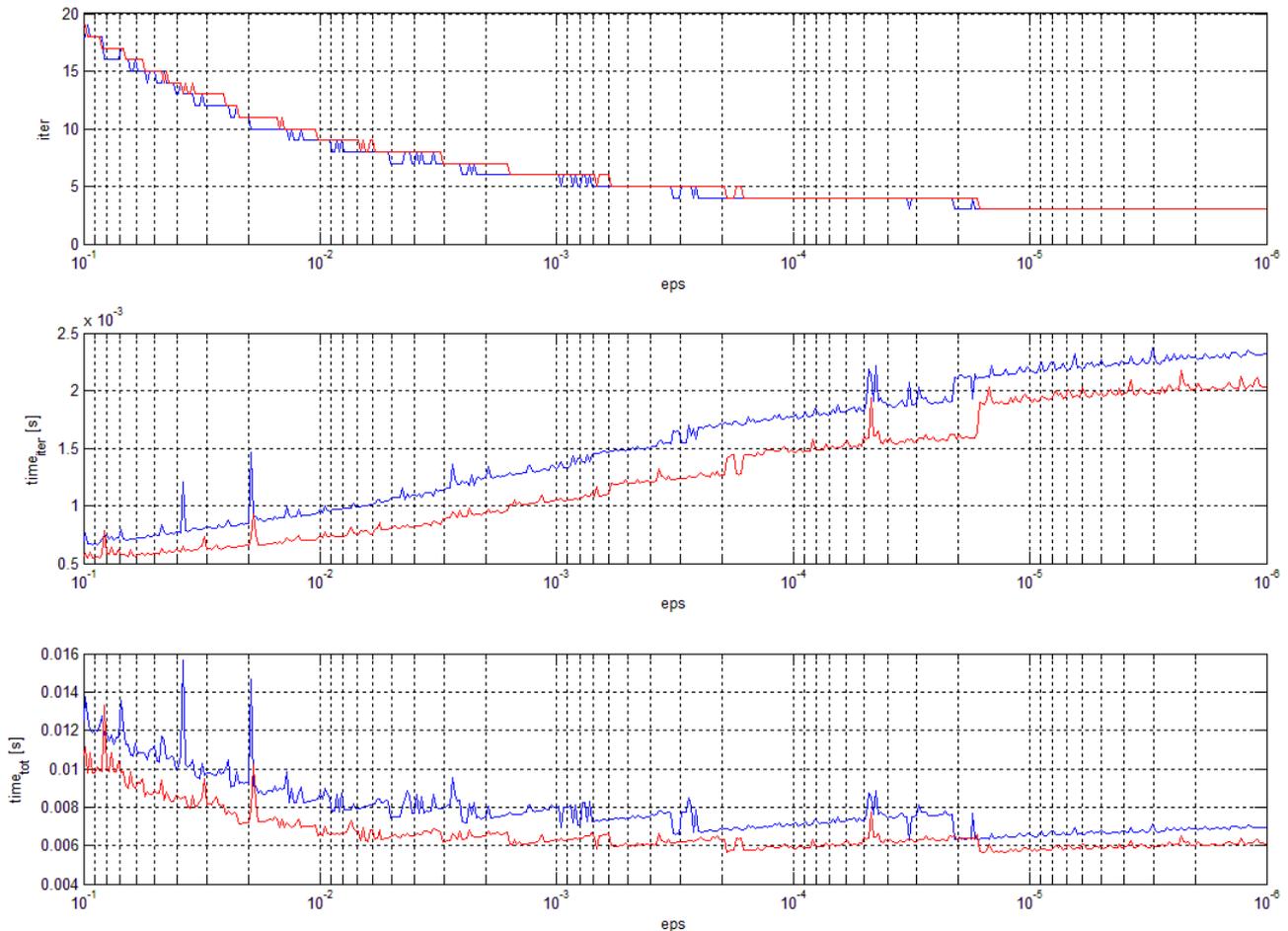


Fig. 10. Andamento dei parametri di controllo al variare di  $eps$  per i metodi BiCG e GMRES

I risultati ottenuti hanno evidenziato come anche in questo caso l'introduzione del preconditionamento consenta un notevole abbattimento dei tempi di calcolo; in particolare, facendo riferimento a dati riportati in Tab. 2 ed in Fig. 10, è possibile effettuare considerazioni analoghe a quelle del precedente paragrafo, riferite questa volta ai metodi BiCG e GMRES.

Tab. 2		BiCG	GMRES
Not conditioned	time_tot [s]	0,1549	0,1154
	iter	496	100
LU Factorization	time_tot [s]	0,0259	0,0377
	iter	1	1

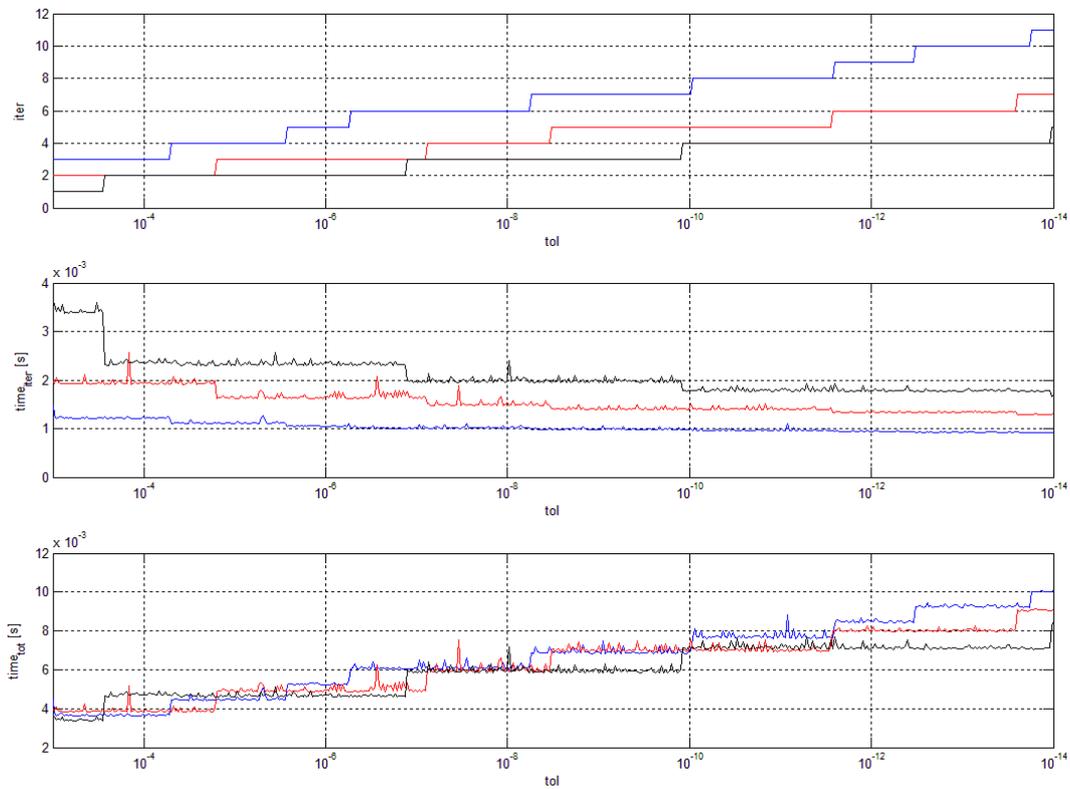


Fig. 11. Andamento dei parametri di controllo al variare di  $tol$  per il metodo BiCG con  $\epsilon = 10^{-2}$ ,  $\epsilon = 10^{-3}$  e  $\epsilon = 10^{-4}$

Nella seconda simulazione sono state confrontate le prestazioni fornite dai metodi BiCG e GMRES al variare del grado di precisione  $tol$  della soluzione; dalle Fig. 11 e 12 si nota come all'aumentare di  $\epsilon$  crescano sia il numero di iterazioni sia i costi computazionali complessivi dei due metodi; inoltre, anche in questo caso utilizzare valori di  $\epsilon$  troppo piccoli non comporta un significativo miglioramento delle prestazioni.

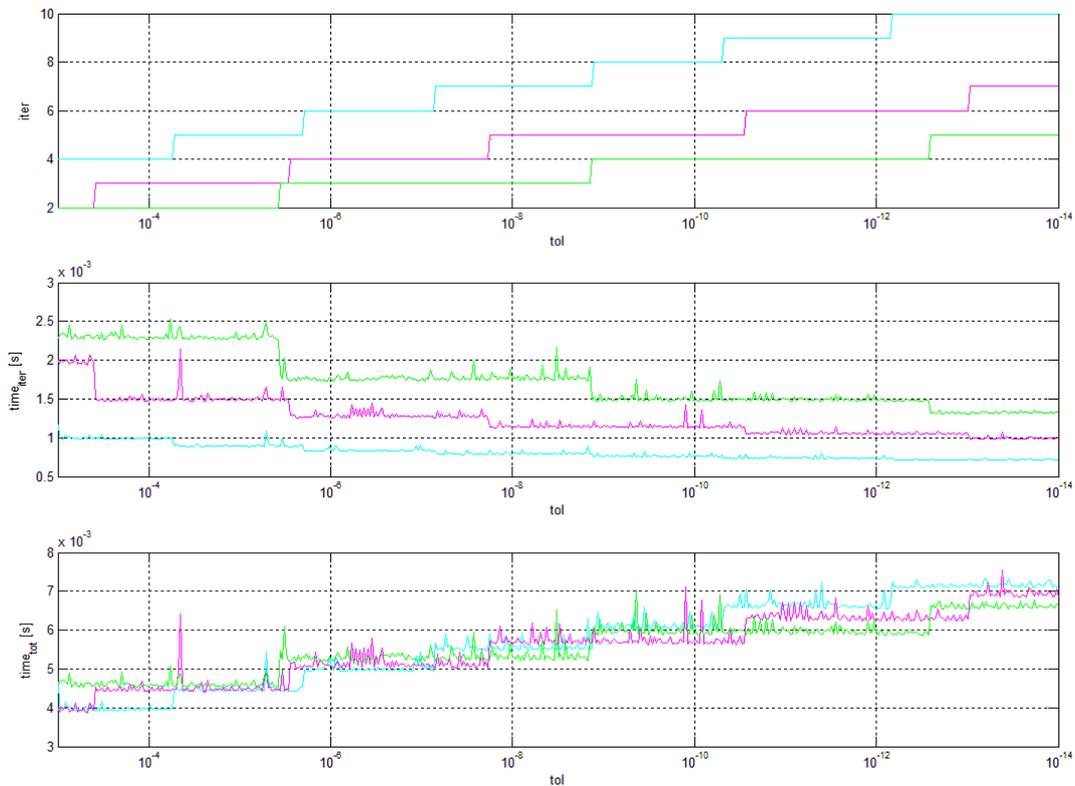


Fig. 12. Andamento dei parametri di controllo al variare di  $tol$  per il metodo GMRES con  $\epsilon = 10^{-2}$ ,  $\epsilon = 10^{-3}$  e  $\epsilon = 10^{-4}$

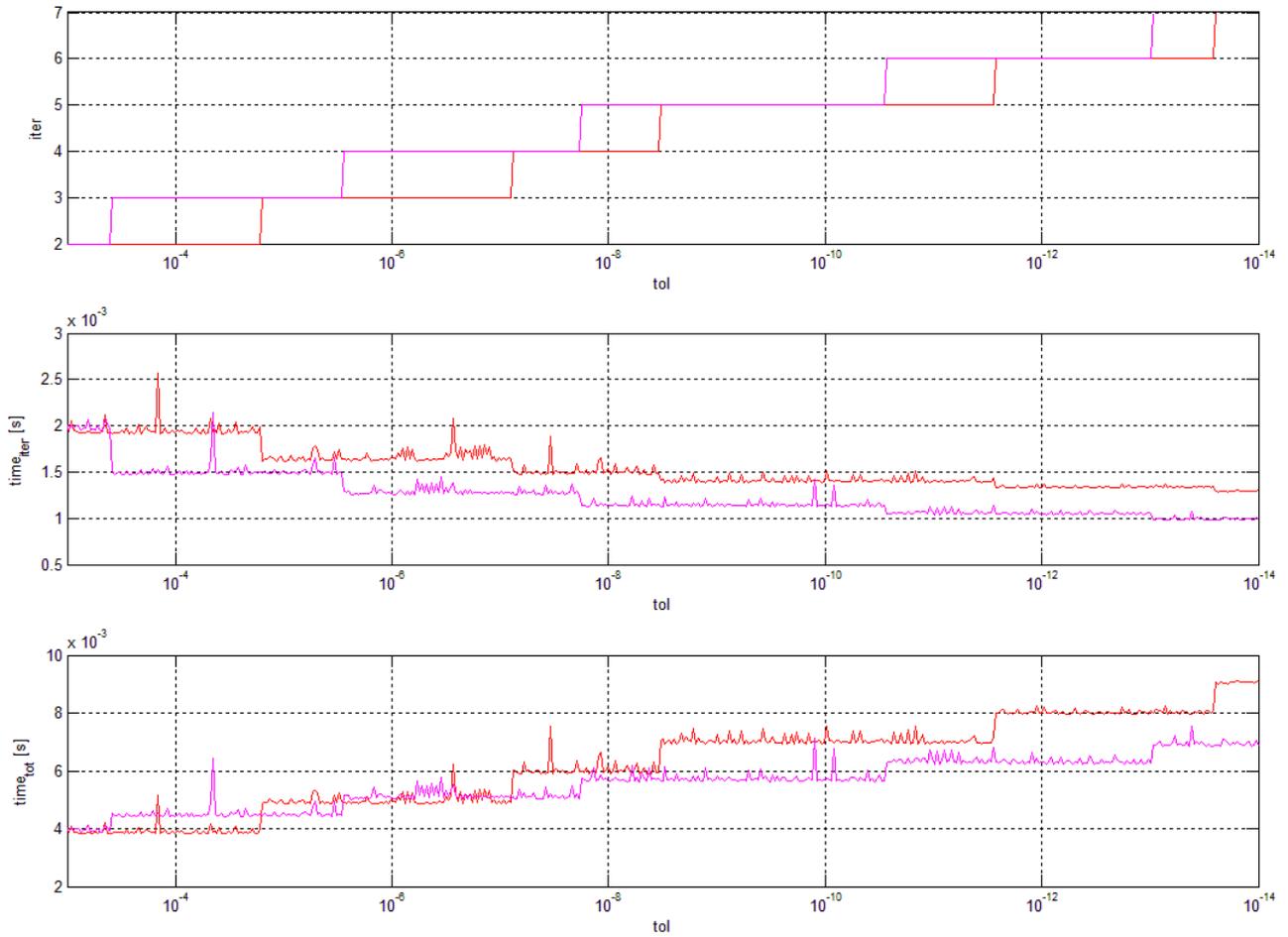


Fig. 13. Confronto fra i parametri di controllo al variare di  $tol$  per i metodi **BiCG** e **GMRES** con  $eps = 10^{-3}$

Confrontando i risultati ottenuti dai due algoritmi per il valore intermedio di  $eps$ , si nota come, nonostante il numero di iterazioni effettuate dai due metodi sia molto simile, i costi computazionali complessivi del metodo GMRES siano inferiori rispetto a quelli ottenuti con il metodo BiCG, come riportato in Fig. 13.

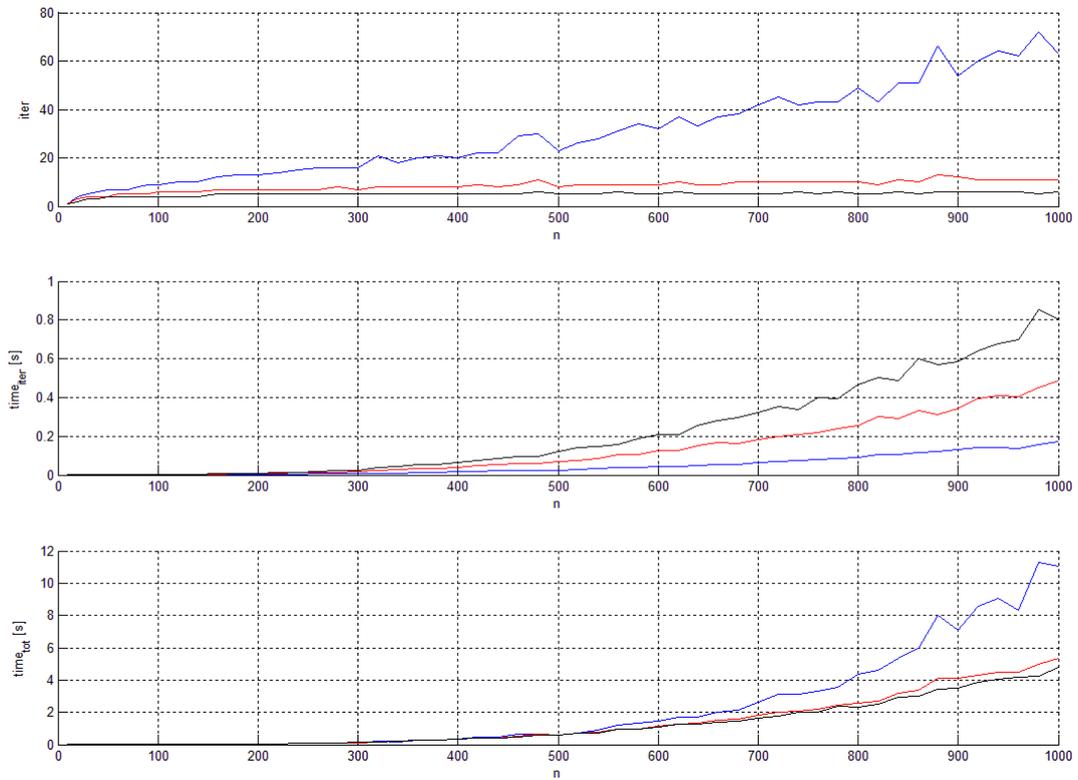


Fig. 14. Andamento dei parametri di controllo al variare di  $n$  per il metodo BiCG con  $\text{eps} = 10^{-2}$ ,  $\text{eps} = 10^{-3}$  e  $\text{eps} = 10^{-4}$

In conclusione, la terza simulazione è stata effettuata allo scopo di verificare l'efficacia degli algoritmi al variare dell'ordine  $n$  del sistema. Facendo quindi riferimento alle Fig. 14 e 15, è possibile notare come, per elevati valori di  $n$  ed al crescere di  $\text{eps}$ , i costi computazionali complessivi in questo caso aumentino.

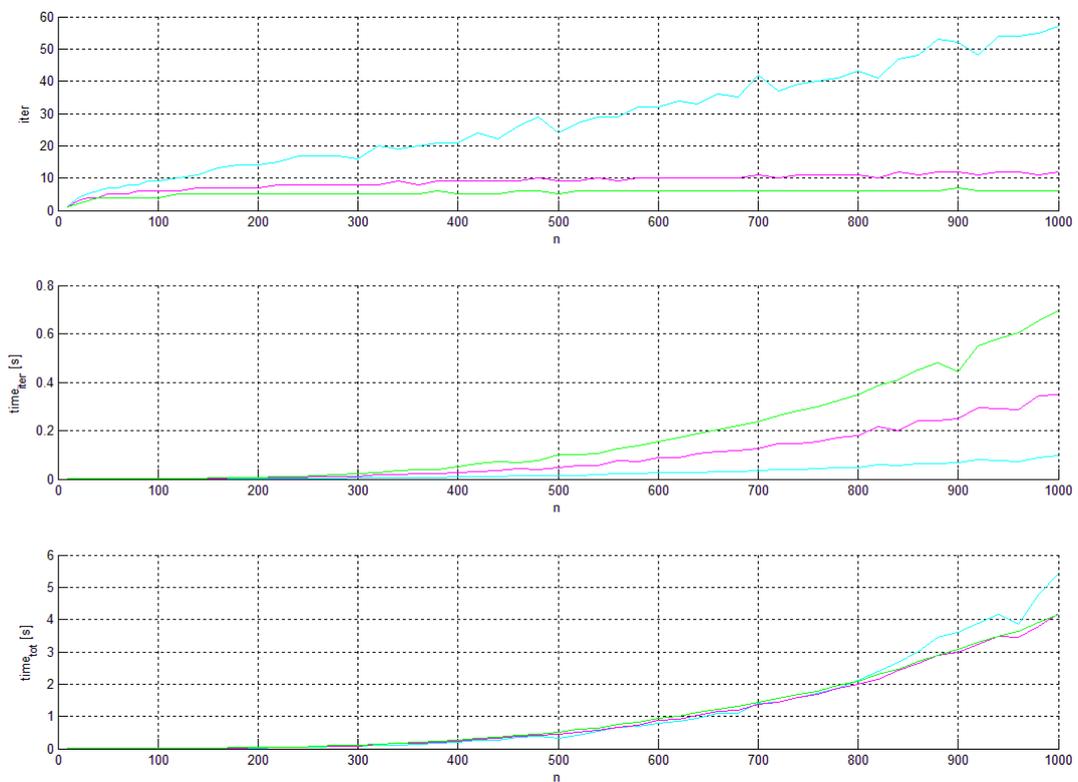


Fig. 15. Andamento dei parametri di controllo al variare di  $n$  per il metodo GMRES con  $\text{eps}=10^{-2}$ ,  $\text{eps}=10^{-3}$  e  $\text{eps}=10^{-4}$

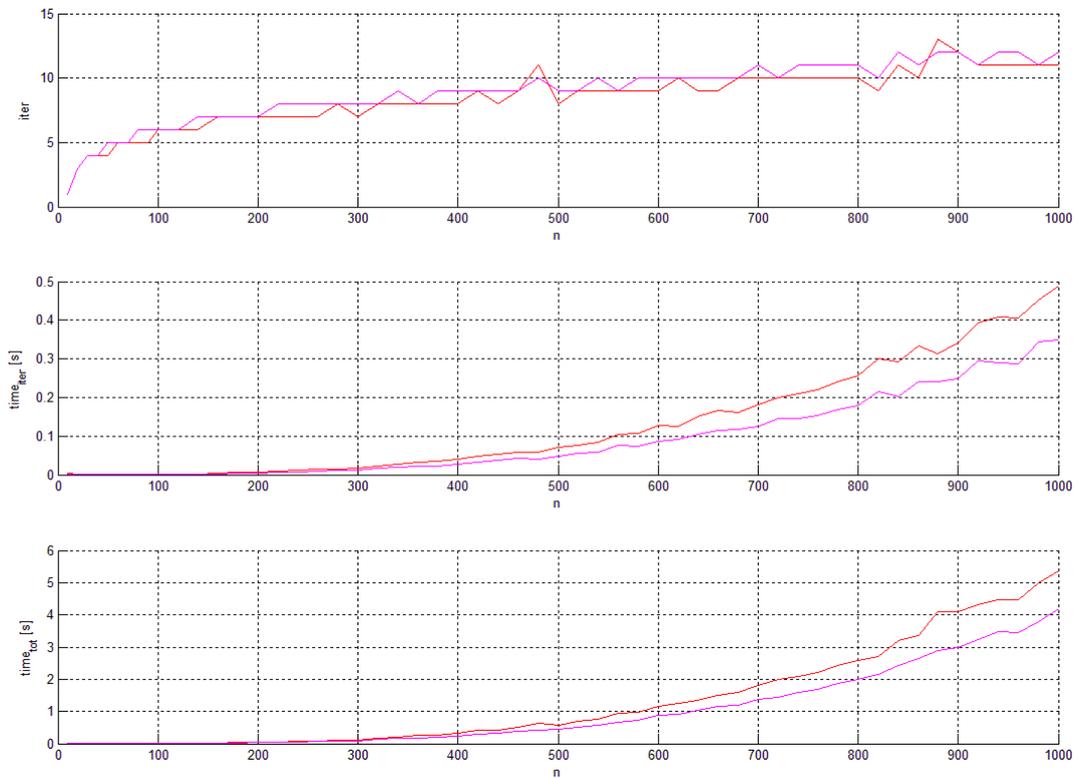


Fig. 16. Confronto fra i parametri di controllo al variare di  $n$  per i metodi **BiCG** e **GMRES** con  $\text{eps} = 10^{-3}$

Infine, il confronto dei due algoritmi per il valore intermedio di  $\text{eps}$  mostra come l'algoritmo GMRES offra migliori prestazioni rispetto al BiCG al crescere dell'ordine  $n$  del sistema, come riportato in Fig. 16.

## 4 – CONCLUSIONI

Nel presente lavoro è stata effettuata un'analisi dei *Projection Methods*: in particolare, è stata fornita una breve descrizione dei principali metodi iterativi utilizzati sia per sistemi simmetrici e definiti positivi (CG, PCG, MINRES) sia per sistemi qualsiasi (CGNR, CGNE, BiCG, GMRES).

Successivamente, mediante l'ausilio del software MATLAB, sono stati confrontati fra loro i risultati ottenuti mediante l'impiego dei metodi CG, PCG e MINRES applicati a sistemi simmetrici e definiti positivi; in modo analogo, è stato effettuato uno studio del comportamento dei metodi BiCG e GMRES applicati a sistemi non simmetrici.

I risultati delle simulazioni effettuate hanno evidenziato l'importanza del preconditionamento, il quale consente un notevole incremento della velocità di convergenza di tutti i metodi esaminati. Per contro, i risultati ottenuti al variare del grado di precisione *tol* e soprattutto dell'ordine del sistema  $n$  hanno fornito indicazioni contraddittorie: infatti, se per i metodi PCG e MINRES l'impiego di preconditionatori eccessivamente precisi (caratterizzati quindi da valori di *eps* molto piccoli) ha mostrato un eccessivo sforzo computazionale all'aumentare di  $n$  (Fig. 6), per i metodi BiCG e GMRES è stato rilevato un comportamento diametralmente opposto, come mostrato in Fig. 14.