

Relazione sul Seminario di Metodi per la Ricerca Operativa

Autori: Samuele Lilliu, n.m. 30355, ing. Elettronica n.o., iscritto al terzo anno.
Matteo Tomasi, n.m. 30354, ing. Elettronica n.o., iscritto al terzo anno.

Sommario: Generalità numeri macchina, tipi di errori, teorema (eps), studio dell'errore, condizionamento di un problema, analisi degli errori nei sistemi lineari, condizionamento di una matrice, risoluzione di un sistema lineare con il metodo di Gauss, risoluzione di sistemi lineari con l'ausilio di MATLAB, Prove (rand, hilb, pascal, tridiag), errori relativi al calcolo degli autovalori.

Introduzione

Nella pratica della programmazione l'utente istruisce l'elaboratore elettronico con una descrizione della procedura necessaria a risolvere un problema, partendo dai dati necessari, tramite una sequenza finita di operazioni aritmetiche e decisioni logiche.

É fondamentale tener presenti le limitazioni imposte sulla precisione raggiungibile dalla modalità di rappresentazione dei numeri e dalle corrispondenti regole aritmetiche.

A causa della struttura fisica dell'hardware ad ogni numero deve necessariamente essere abbinato un numero finito e prefissato di cifre binarie e pertanto non tutti i numeri possono essere rappresentati esattamente in un elaboratore.

Questo ci fa capire che se A é l'insieme dei numeri rappresentabili con un elaboratore elettronico, allora $A \subset \mathfrak{R}$ e quindi A é solo un piccolo sottoinsieme dei numeri reali.

I numeri rappresentabili esattamente si chiamano *numeri macchina*. Ad esempio non potranno essere rappresentabili come numeri macchina i risultati delle operazioni aritmetiche, chiamate *operazioni macchina*.

Errori

Nelle applicazioni matematiche, non si verificano solo errori di arrotondamento, ma anche altri tipi, convenzionalmente individuati in tre categorie:

- *errori inerenti*: dovuti alle schematizzazioni che vengono fatte per ridurre in termini matematici i problemi che si presentano nella pratica o dovuti ad errori nelle misure sperimentali di certe grandezze;
- *errori di troncamento*: quando ad un procedimento ad infiniti passi se ne sostituisce uno costituito da un numero finito di passi;
- *errori di arrotondamento*: errore che si commette nel considerare il numero macchina nella sua rappresentazione in virgola mobile.

Gli errori inerenti sono strettamente connessi sia all'intuito di chi schematizza, in termini matematici, un fenomeno reale, sia nell'abilità dello sperimentatore. Gli errori di troncamento possono, in generale, essere trattati con i metodi dell'analisi numerica. Gli errori di arrotondamento dipendono quasi esclusivamente dallo strumento di calcolo che si ha a disposizione. Una loro valutazione rigorosa é spesso difficile. É tuttavia indispensabile fornire, metodi che permettano di valutare la loro influenza sulla precisione dei risultati.

Gli errori dovuti all'arrotondamento di numeri utilizzando il sistema decimale sono più piccoli di quelli commessi utilizzando un sistema di numerazione binario:

$$\begin{cases} |e_{r10}| < 5 \times 10^{-m} \\ |e_{r2}| < 2^{-m} \end{cases}$$

con valore arrotondato alla m-esima cifra (binaria o decimale).

Quando si rappresenta un numero reale x in un calcolatore (con t bit a disposizione per la mantissa) si presentano i seguenti 4 casi:

1. x può essere rappresentato in modo esatto; in tal caso si dice che x è un numero di macchina (come le potenze di 2);
2. x , in valore assoluto, è più piccolo del più piccolo numero rappresentabile; in questo caso si verifica un *underflow* e generalmente si assume 0 come valore approssimato di x (con eventuale segnalazione 'warning');
3. x , in valore assoluto, più grande del più grande numero rappresentabile; in questo caso di verifica un *overflow*, non si effettua nessuna approssimazione (x viene approssimato come $\pm\infty$ a secondo del segno) e il sistema arresta il calcolo;
4. la caratteristica è rappresentabile in modo esatto, ma la mantissa m di x non è rappresentabile con t cifre; il numero x è approssimato con il numero di macchina x' , la mantissa m viene approssimata con la mantissa m' .

Relativamente all'ultimo caso vale il seguente **teorema (eps)**:

Dato un numero reale x non nullo, e dato un numero x' dipendente dal numero di bit a disposizione per la mantissa, dal massimo e minimo numero rappresentabili dalla macchina, l'errore relativo (errore di arrotondamento o round error) che si commette approssimando x con x' è:

$$|\varepsilon_x| \equiv \frac{|x - x'|}{|x|} \leq eps$$

$$eps = \begin{cases} \beta^{1-t} & \text{caso del troncamento} \\ \frac{1}{2}\beta^{1-t} & \text{caso dell'arrotondamento} \end{cases}$$

in cui $x = \pm m \mathbf{b}^p$ dove $m = 0.a_1a_2\dots a_t a_{t+1}$ e a_i compreso tra 0 e $(\beta-1)$, con a_1 diverso da zero. Mentre per $x' = \pm m' \mathbf{b}^p$

$$m' = \begin{cases} 0.a_1a_2\dots a_t & \text{troncamento} \\ 0.a_1a_2\dots \bar{a}_t & \text{arrotondamento} \end{cases} \quad \bar{a}_t = \begin{cases} a_t + \beta^{-t} & \text{se } a_{t+1} \geq \frac{1}{2}\beta \\ a_t & \text{se } a_{t+1} < \frac{1}{2}\beta \end{cases}$$

La quantità **eps** è detta **precisione di macchina**, oppure unità di arrotondamento, oppure zero macchina.

Il valore di eps è ricavabile in ambiente MATLAB:

```
>> eps
```

```
ans =
```

```
2.2204e-016
```

```
>>
```

Dunque, la precisione di macchina è una maggiorazione dell'errore relativo che si commette quando si approssima un numero reale con un numero di macchina.

Studio dell'errore

Per studiare quanto si riflettono gli errori sulla soluzione esatta di un problema introduciamo due tipi di rappresentazione dell'errore:

Definizione : Sia \mathbf{x} la soluzione esatta di un modello matematico che risolve il problema considerato e \mathbf{x}' la soluzione ottenuta attraverso le computazioni dell'algoritmo risolutivo, allora $\mathbf{x}' - \mathbf{x}$ è l' **Errore Assoluto**.

Tale stima non è da sola esauritiva in quanto evidenzia di quanto si discosta la soluzione calcolata da quella esatta ma non di quanto si *rifletta* l'errore dalla soluzione esatta.

Definizione : Sia \mathbf{x} la soluzione esatta di un modello matematico che risolve il problema considerato e \mathbf{x}' la soluzione ottenuta attraverso le computazioni dell'algoritmo risolutivo,

allora $\mathbf{e}_x = \frac{\mathbf{x}' - \mathbf{x}}{\mathbf{x}}$ è chiamato **Errore Relativo**. Solitamente se ne considera il valore assoluto.

Nel nostro caso, trattandosi di vettori abbiamo preso l'errore relativo con norma 2.

Questo parametro è un indicatore di quanto si riflette l'errore sul problema. Se ne deduce che l'errore è tollerabile quando il valore si avvicina allo 0.

Condizionamento di un problema

Lo studio di problemi con valori di grandi dimensioni in *matematica discreta* porta a casistiche in cui gli errori sono inaccettabili ma non potendo conoscere la soluzione esatta non possiamo sapere che valori assumeranno questi ultimi.

Definizione: Condizionamento di un problema Dato un problema espresso tramite relazioni tra valori numerici, il condizionamento è un valore che quantifica di quanto viene amplificato l'errore in ogni computazione tra gli errori sui dati in ingresso e quelli sui dati in arrivo .

Analisi degli errori nei sistemi lineari

Nella risoluzione di sistemi lineari si usa la rappresentazione matriciale poichè vi sono dei metodi efficienti per la determinazione delle soluzioni (come il metodo di Gauss). Siccome si usano i calcolatori per sistemi di grandi dimensioni, prevedendo di avere un valore di deviazione dei valori molto elevato è chiaro che l'aspetto la stima dell' errore induce a considerare gli effetti che si producono sottoponendo la matrice alle computazioni che necessita l' algoritmo risolutivo .

Il **condizionamento di una matrice A** non singolare, relativo alla risoluzione di un sistema lineare, é dato dal seguente numero:

$$m_p(A) = \|A\| \times \|A\|^{-1}$$

In ambiente MATLAB il numero di condizionamento relativo all'operazione di inversione si trova mediante l'istruzione `cond(A)` dove A è una matrice quadrata, invertibile. **Un grande valore indica che la matrice è prossima ad essere singolare.**

Esempio di calcolo del condizionamento con la definizione:

```
>> A
A =
     2     3
     4     5
>> norm(A)*norm(inv(A))
ans =
    26.9629
>> cond(A)
ans =
    26.9629
>>
```

Risoluzione di sistemi lineari con il metodo di Gauss

Nel nostro caso è stato utilizzato il metodo di Gauss che consiste sostanzialmente nell'applicazioni di:

$$\begin{cases} y'_{ij} = y_{ij} - \frac{y_{pj}}{y_{pq}} y_{iq}, & i \neq p \\ y'_{pj} = \frac{y_{pj}}{y_{pq}} \end{cases}$$

Per ogni i diversa da p , e per ogni j . Dove y_{ij} , sono elementi della matrice A, con i e j rispettivamente indici di riga e colonna. Per una matrice quadrata $M \times M$ occorrerà iterare il blocco di operazioni M volte per ogni pivot corrispondente ad ogni elemento della diagonale.

Queste sequenze di divisioni, possono portare sicuramente ad errori nella loro rappresentazione macchina. Essendo poi divisioni occorre che tutti i suoi pivot siano non

nulli. I pivot di una matrice quadrata *non sono univocamente determinati*; i pivot cambiano a seconda di quali scambi di riga si vengono effettuati. Il *determinante* di una matrice corrisponde al valore assoluto del prodotto di tutti i pivot.

Questo metodo presenta comunque il vantaggio di ridurre notevolmente la propagazione degli errori.

Risoluzione di sistemi lineari con Matlab

MATLAB risolve il sistema lineare $\mathbf{Ax} = \mathbf{b}$ con il comando:

$x = A \backslash b$.

Se A è quadrata, questo comando implica i seguenti passi:

- Calcola la fattorizzazione LU (con pivoting parziale);
- Risolvi i due sistemi triangolari;

Se A è rettangolare (sistemi sovradeterminati), Matlab calcola la soluzione nel senso dei minimi quadrati, cioè:

- Calcola la fattorizzazione QR (con pivoting sulle colonne);
- Risolve il sistema triangolare superiore;

Se il condizionamento di A è elevato, stampa un messaggio di warning, ma calcola ugualmente la soluzione. Inoltre occorre prestare attenzione perché Matlab calcola una soluzione anche quando il sistema non ammette soluzione.

Prove

Sono state effettuate delle prove sulla valutazione della norma dell'**errore relativo** commesso nella risoluzione di un sistema lineare $\mathbf{Ax} = \mathbf{b}$, utilizzando una soluzione \mathbf{e} assegnata (vero valore), ricavando la colonna RHS con $\mathbf{b} = \mathbf{A} * \mathbf{e}$, quindi risolvendo $\mathbf{Ax} = \mathbf{b}$, per ottenere \mathbf{x} (valore misurato) con il metodo di Gauss. Si è quindi calcolato il valore $\mathbf{err} = \|\mathbf{x} - \mathbf{e}\| / \|\mathbf{e}\|$.

Sono state fatte quattro prove utilizzando come matrici-test:

- `rand(n)`
- `hilb(n)`
- `pascal(n)`
- `tridiag(n)`

che sono un esempio di matrici mal condizionate.

per ogni prova è stato utilizzata come soluzione **vera**, \mathbf{e} :

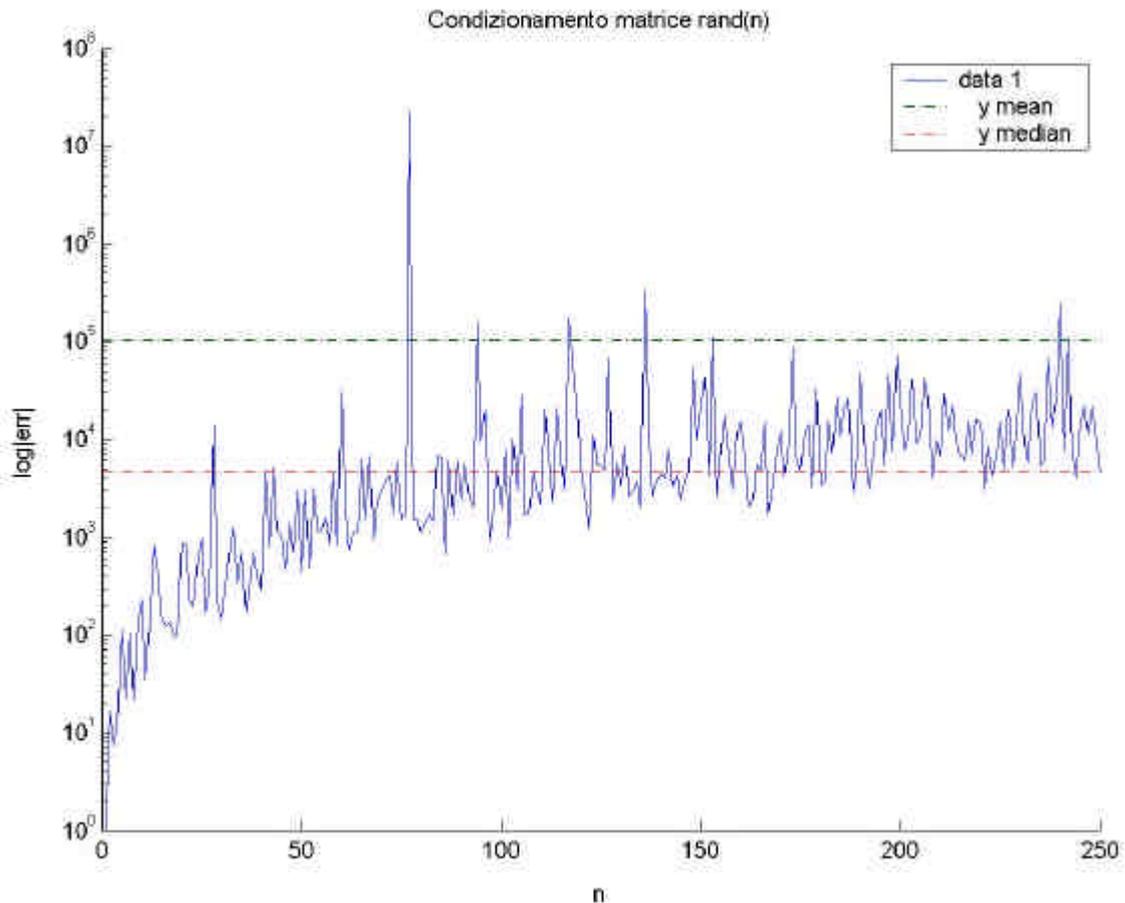
- $\mathbf{e} = \mathbf{ones}(n)$ (figura a)
- $\mathbf{e1} = 1 + 1/(2*j+1)$; (figura b, blu)
- $\mathbf{e2} = 1/(2*j+1)$; (figura b, magenta)
- $\mathbf{e3} = 1/(64*j+1)$; (figura b, verde)

in ciascuna prova è stato fatto variare n. Infine è stato calcolato il condizionamento in funzione della dimensione della matrice A (figura c). Il listato .m è stato modificato per le prove 2, 3, 4 solo nella riga in cui compare $\mathbf{A} = \dots$

In questo modo, confrontando i grafici di figura a, con quelli di figura b e c, è stato possibile osservare un incremento dell'errore relativo in funzione del condizionamento.

A = rand (n)

RAND(N) restituisce una matrice NxN con valori pseudo-casuali, scelti in una distribuzione uniforme nell'intervallo [0,1]. La sequenza dei numeri generati dipende dallo stato del generatore. Una verifica sul condizionamento della matrice rand con n variabile da 0 a 250 mostra un andamento che si mantiene entro un certo inviluppo che tende a saturare verso un valore costante. Non compaiono Warnings, dato che median = 4615. Ciò significa che le matrici calcolate con rand non sono mal condizionate.



Listato .m

```
clear all;
close all;
clf;
hold on;

for i = 1:100,
    xx(i)=i;

    e = ones (i,1);
    for j = 1:i,
        e1(j) = 1+1/(2*j+1);
        e2(j) = 1/(2*j+1);
        e3(j) = 1/(64*j+1);
    end
    A = rand (i);
```

```

b = A*e ;
b1 = A*e1';
b2 = A*e2';
b3 = A*e3';

x = A\b ;
x1 = A\b ;
x2 = A\b ;
x3 = A\b ;

err(i) = norm (x-e,2);
err1(i) = norm (x-e1',2)/norm (e1,2);
err2(i) = norm (x-e2',2)/norm (e2,2);
err3(i) = norm (x-e3',2)/norm (e3,2);
con(i) = cond (A,2);

```

```
end
```

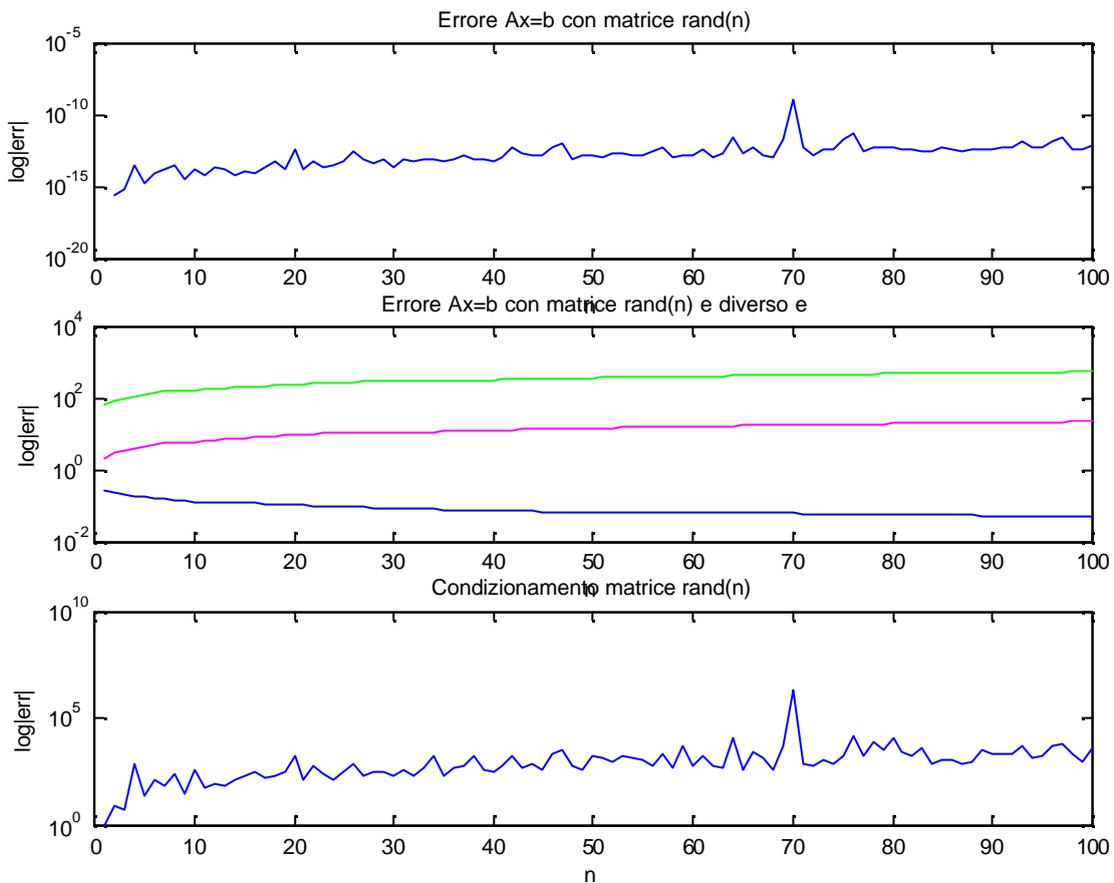
```

subplot(3,1,1);
semilogy(xx,err,'b')
title('Errore Ax=b con matrice rand(n)');
xlabel('n');
ylabel('log|err|');

subplot(3,1,3);
semilogy(xx,con);
title('Condizionamento matrice rand(n)');
xlabel('n');
ylabel('log|err|');
hold on
subplot(3,1,2);
semilogy(xx,err1,'b',xx,err2,'m',xx,err3,'g')
title('Errore Ax=b con matrice rand(n) e diverso e');
xlabel('n');
ylabel('log|err|');

```

Notiamo sotto, in **fig.a** un errore molto piccolo prossimo all'**eps**, che sale lentamente all'aumentare delle dimensioni della matrice. Nella **fig.b** relativamente alla curva blu osserviamo un decremento dell'errore relativo dovuto al fatto che per $j \rightarrow \infty$, $e1=1$; pertanto, al limite di $n \rightarrow \infty$, l'errore scende ai valori della **fig.a**. Sempre nella **fig.b** le diverse caratteristiche della curva verde rispetto a quella Magenta sono dovute alla maggiore rapidità della prima a tendere a zero rispetto alla seconda.



A=hilb(n)

Le matrici di Hilbert sono un esempio di mal condizionamento.

HILB(N) restituisce una matrice NxN con elementi $1/(i+j-1)$.

Il valore esatto della sua inversa si calcola con INVHILB(N).

```
>> hilb(3)
```

```
ans =
```

```
1.0000    0.5000    0.3333
0.5000    0.3333    0.2500
0.3333    0.2500    0.2000
```

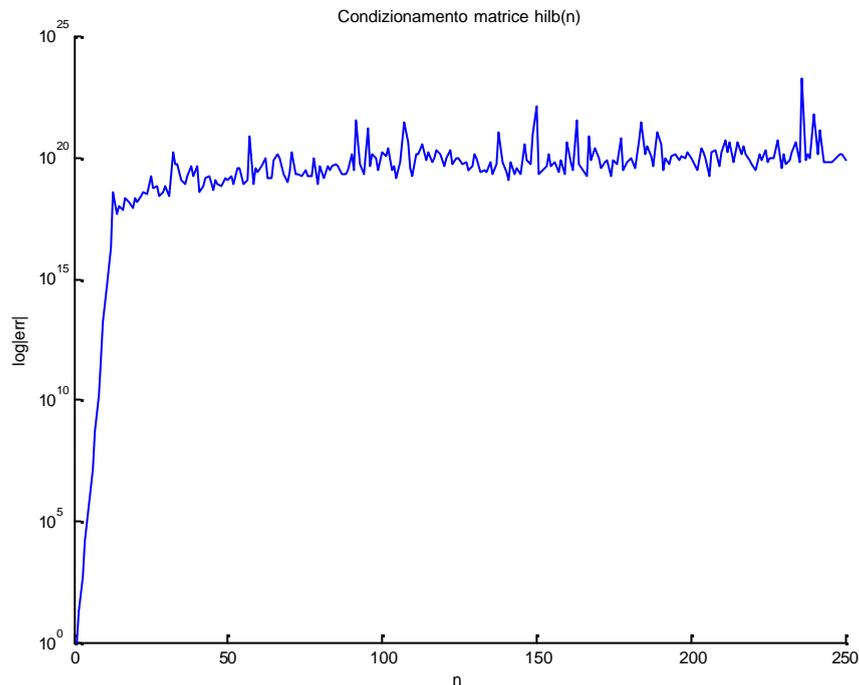
```
>> invhilb(3)
```

```
ans =
```

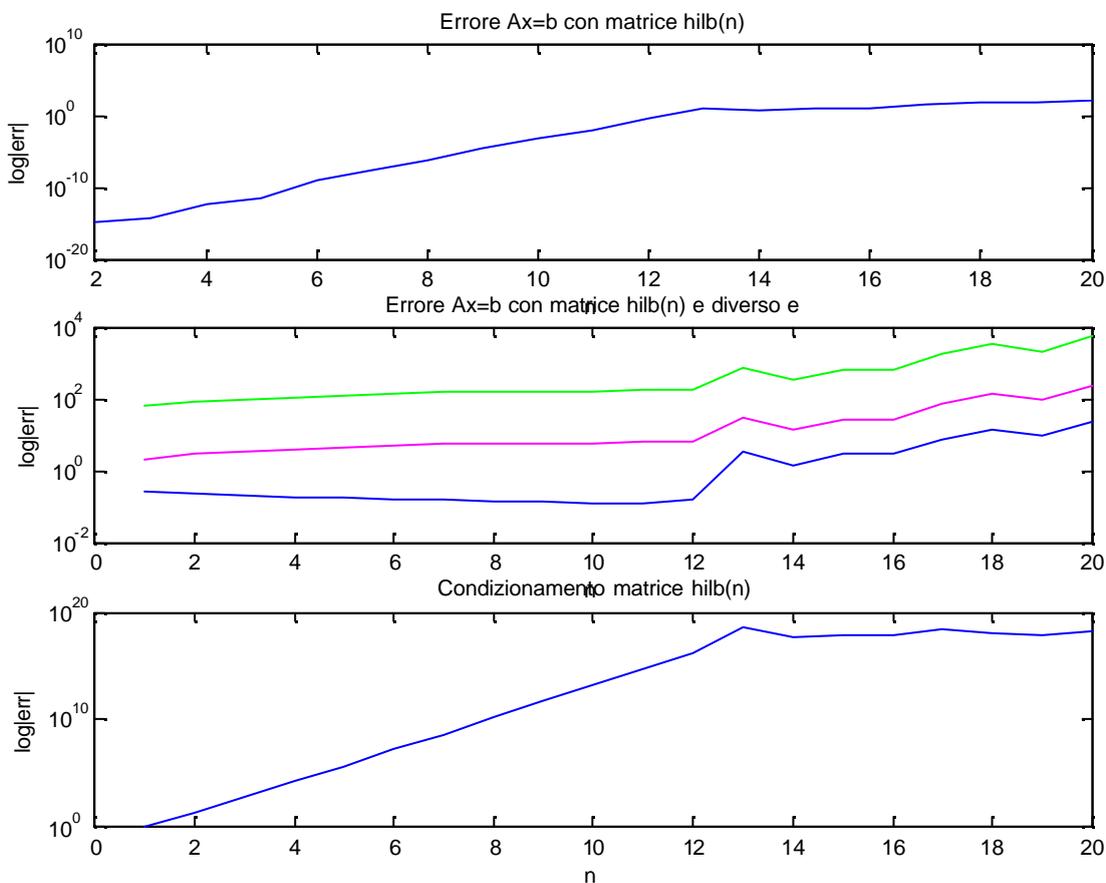
```
9    -36    30
-36   192  -180
30   -180   180
```

Mostriamo sotto l'andamento del condizionamento della matrice di Hilbert in funzione della sua dimensione. Si può osservare l'andamento esponenziale per n compreso tra 0 e 13. Dopo di che si manifesta un comportamento 'caotico', compendiatore nella risoluzione dei sistemi lineari da messaggi di Warning quali:

Warning: Matrix is close to singular or badly scaled.
 Results may be inaccurate. RCOND = 9.654750e-022.
 In cui $RCOND=1/\text{cond}(A)$.



Riportiamo sotto i risultati relativi all'errore relativo.



Si osserva che in fig.a l'errore relativo segue sostanzialmente l'andamento del condizionamento

assestandosi intorno al 100% quando $n = 13$. Per quanto riguarda la figura b, la curva in blu si impenna fortemente a partire da un minimo assunto in corrispondenza di $n=12$ giungendo al 100% quando $n=13$. Per le curve color Magenta e Verde valgono le considerazioni precedenti.

A=pascal(n)

PASCAL(N) è una matrice di ordine N simmetrica, positiva, a valori interi, costruita con il triangolo di Pascal.

pascal (4)

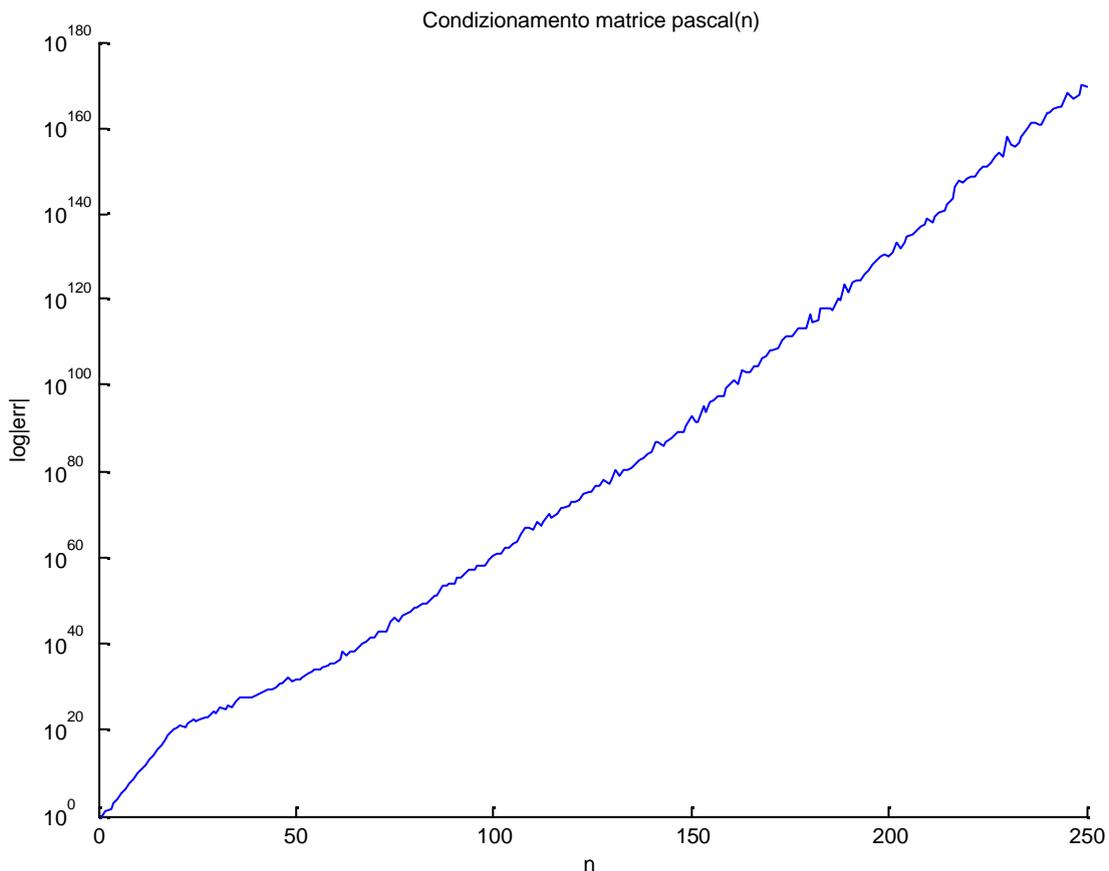
ans =

```

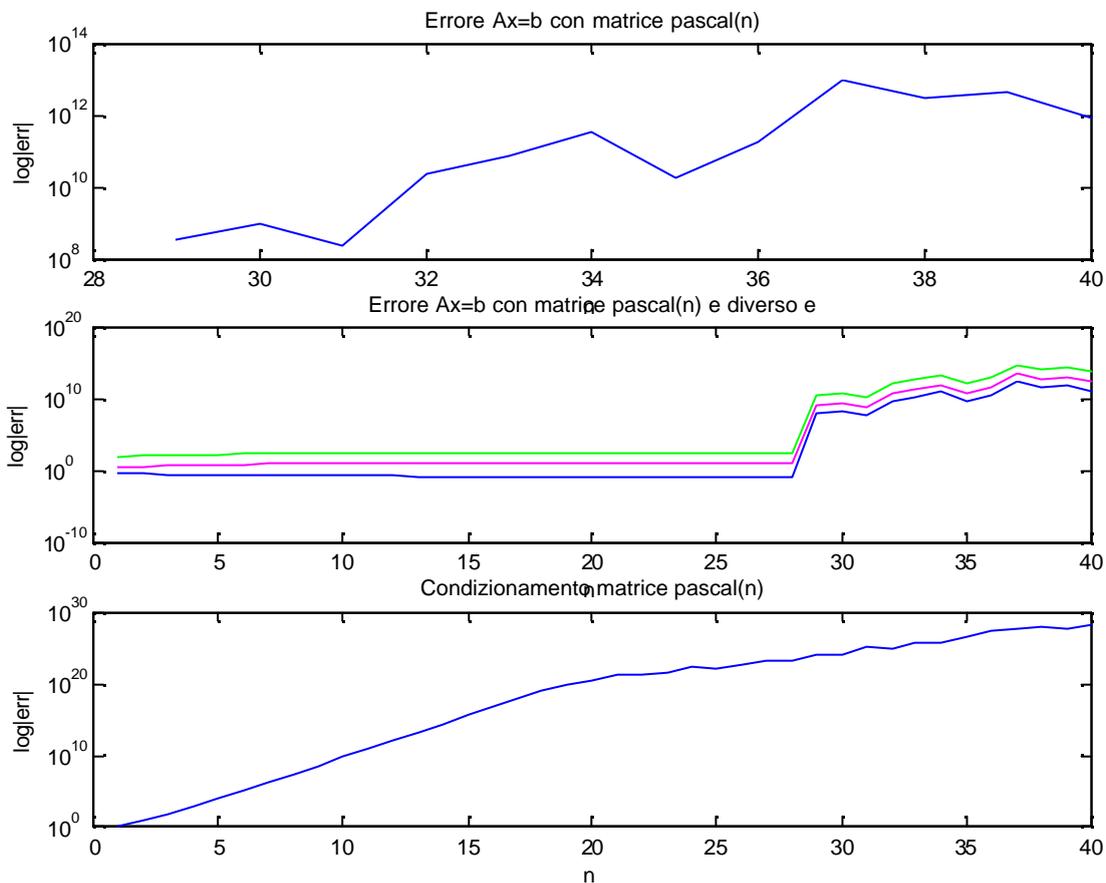
1  1  1  1
1  2  3  4
1  3  6 10
1  4 10 20

```

Si tratta di una matrice mal condizionata. Riportiamo di sotto l'andamento del condizionamento della matrice per n compreso tra 0 e 250. L'andamento è nettamente esponenziale fino ad $n=30$, poi continua a crescere con un andamento vagamente 'caotico' e con una minore pendenza.



Nella risoluzione del sistema si manifestano effetti simili a quelli precedenti ma su scala maggiore.



A=tridiag(n)

In questo caso nel listato compare in luogo di A=...

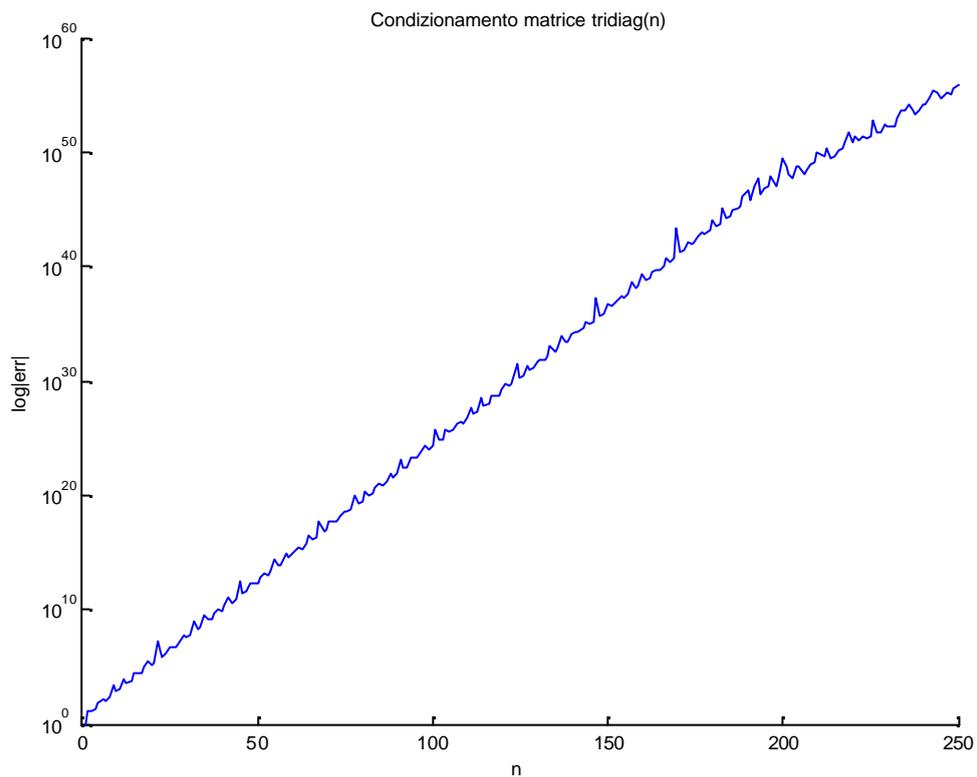
a=1;

b=2

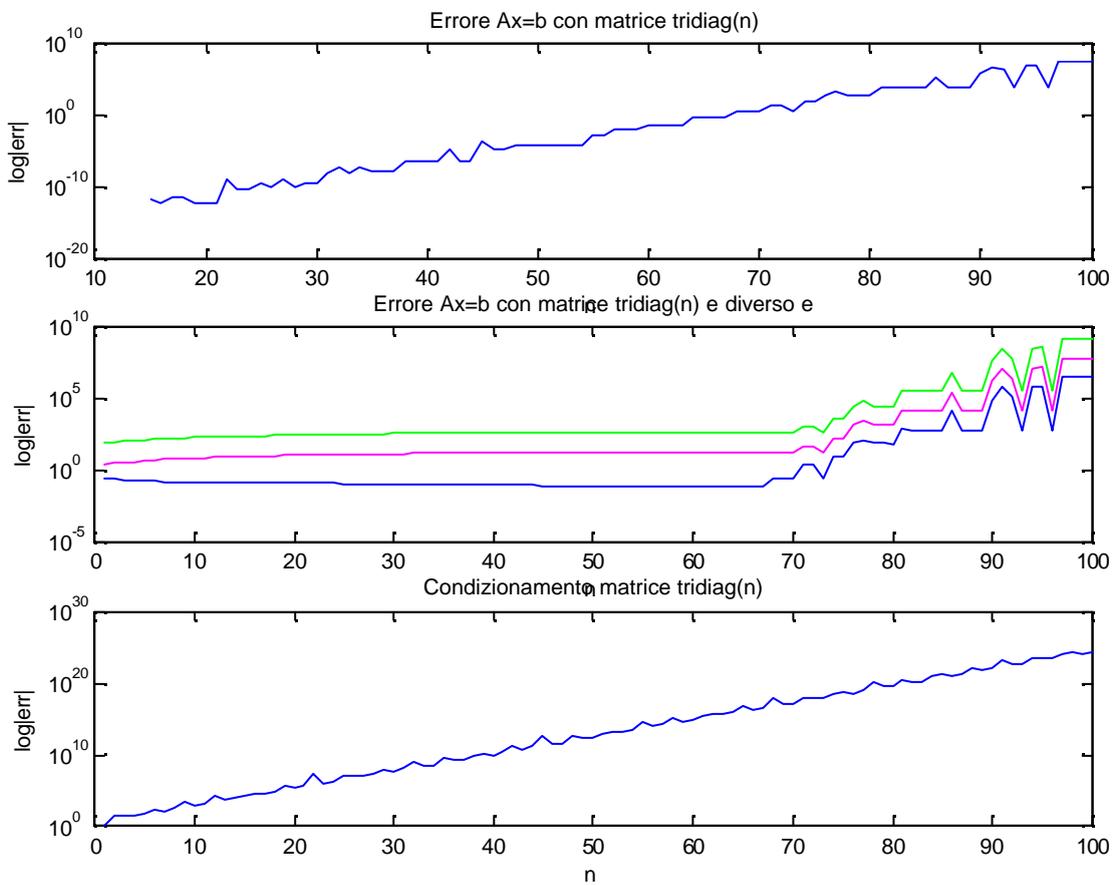
c=3;

A = b*diag(ones(i,1)) + c*diag(ones(i-1,1),1) + a*diag(ones(i-1,1),-1);;

Condizionamento della matrice



Risoluzione sistema lineare:



Errori relativi al calcolo di autovalori

In analogia con la risoluzione dei sistemi lineari, per trovare l'errore commesso dalla macchina partiamo da un problema di cui conosciamo la soluzione. Per far questo abbiamo generato due tipi di matrici:

- una matrice diagonale (simmetrica).
- una matrice triangolare (asimmetrica).

Entrambe con autovalori definiti da un vettore v da noi assegnato in 3 diverse varianti.

Abbiamo svolto i calcoli in parallelo in modo da poter poi confrontare i risultati; in tal modo abbiamo verificato che i problemi con matrici asimmetriche producono errori maggiori rispetto a quelli con matrici simmetriche.

In entrambi i casi abbiamo utilizzato il seguente algoritmo per il confronto fra risultato esatto (vettore v) e risultato fornito dalla macchina (vettore d):

- generiamo una matrice Q ortogonale, per la quale l'inversa equivale alla trasposta, in modo tale da non commettere errori nell'inversione.
- Tramite la relazione $A = Q^T * D * Q$ calcoliamo una matrice simile a D (che dovrebbe quindi avere gli stessi autovalori).
- Calcoliamo gli autovalori di A tramite la funzione "eig" e memorizziamo il risultato in un vettore d .
- Calcoliamo l'errore assoluto della nostra operazione tramite la norma della differenza tra i due vettori.
- Grafichiamo in parallelo i diversi risultati ottenuti al variare del vettore v e al crescere delle dimensioni delle nostre matrici.

Questo algoritmo è stato generato tramite il seguente m-file:

% Errore sul calcolo degli autovalori per matrici simmetriche e asimmetriche.

```
clear all
close all

for i= 1:100,
    n(i)=i;
    for j=1:i,
        v=1+1/(2*j+1);
        v1=1/(2*j+1);
        v2=1/(64*j+1);
    end
    D=diag(v);
    Das=diag(v)+diag(ones(i-1,1),1);
    D1=diag(v1);
    D2=diag(v2);
    Das1=diag(v1)+diag(ones(i-1,1),1);
    Das2=diag(v2)+diag(ones(i-1,1),1);
    Q=orth(rand(i));
    A=Q'*D*Q;
    Aas=Q'*Das*Q;
    A1=Q'*D1*Q;
    A2=Q'*D2*Q;
    Aas1=Q'*Das1*Q;
    Aas2=Q'*Das2*Q;
```

```

d=eig(A);
das=eig(Aas);
d1=eig(A1);
d2=eig(A2);
das1=eig(Aas1);
das2=eig(Aas2);
err(i) = norm (v'-d);
erras(i) = norm (v'-das);
err1(i) = norm (v1'-d1);
err2(i) = norm (v2'-d2);
erras1(i) = norm (v1'-das1);
erras2(i) = norm (v2'-das2);
con(i)= cond(A);
con1(i)= cond(A1);
con2(i)= cond(A2);
conas(i)= cond(Aas);
conas1(i)= cond(Aas1);
conas2(i)= cond(Aas2);
end

hold on
figure(1)
subplot(221)
semilogy(n,err,'b',n,err1,'g',n,err2,'r')
title('errore con matrice simmetrica')
xlabel('n')
ylabel('log(err)')

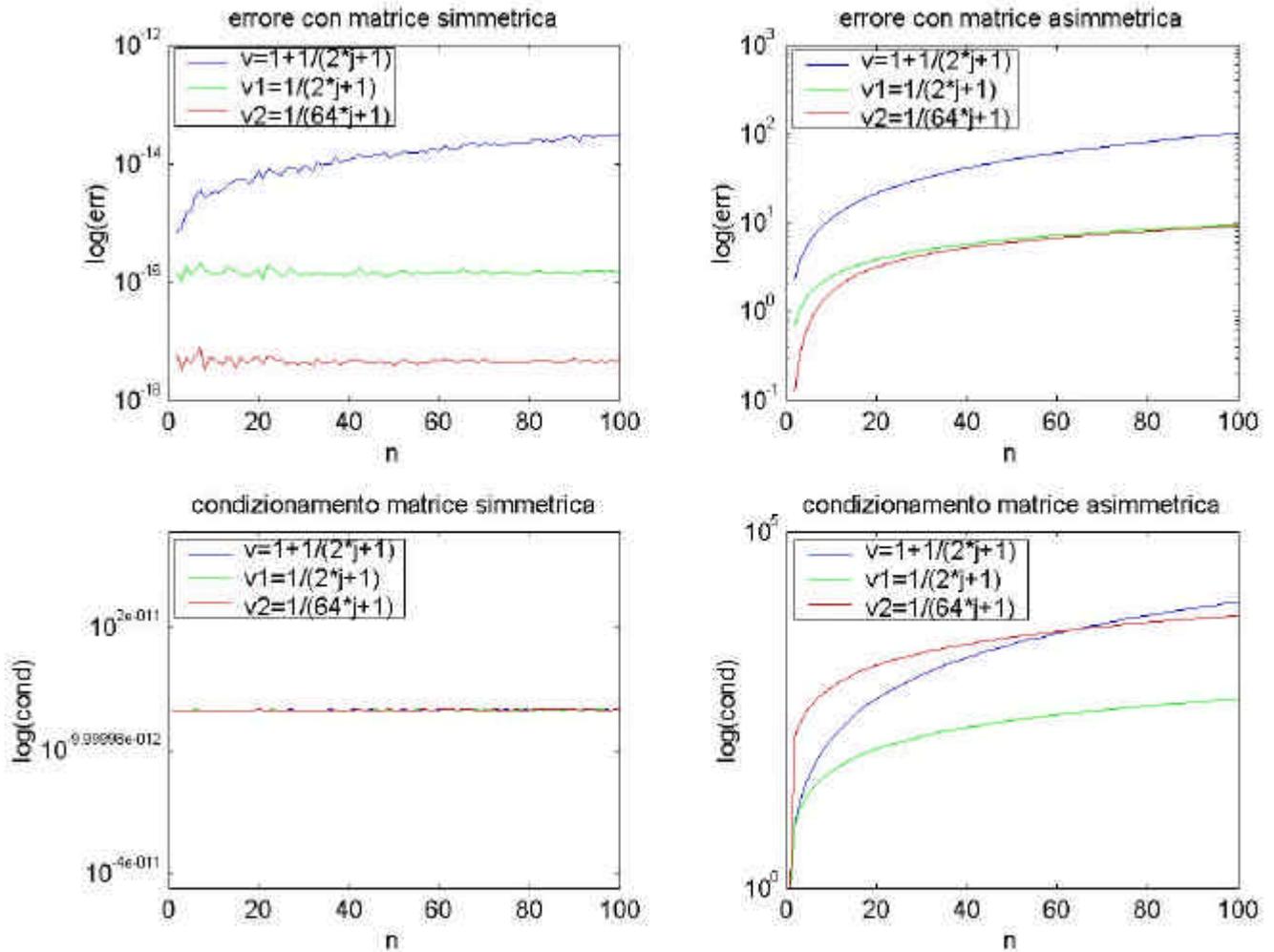
subplot(222)
semilogy(n,erras,'b',n,erras1,'g',n,erras2,'r')
title('errore con matrice asimmetrica')
xlabel('n')
ylabel('log(err)')

subplot(223)
semilogy(n,con,'b',n,con1,'g',n,con2,'r')
title('condizionamento matrice simmetrica')
xlabel('n')
ylabel('log(cond)')

subplot(224)
semilogy(n,conas,'b',n,conas1,'g',n,conas2,'r')
title('condizionamento matrice asimmetrica')
xlabel('n')
ylabel('log(cond)')

```

Con tale procedimento abbiamo ottenuto i seguenti risultati:



Dove si nota l'andamento dell'errore al crescere dell'ordine delle matrici n che in questo caso varia da 1 a 100.

Da notare che all'interno dei vettori v la lettera j è da intendersi come indice e non come numero immaginario.

E' palesemente visibile come i due errori (matrice simmetrica a sinistra e matrice asimmetrica a destra) si discostino l'uno dall'altro per diversi ordini di grandezza. Questo è comprensibile dai grafici più in basso rappresentanti il condizionamento delle matrici:

quelle simmetriche presentano un condizionamento unitario e costante con il crescere di n mentre quelle asimmetriche un condizionamento maggiore di 1 e crescente al crescere di n .

Ancora si può notare come il vettore v costituito da numeri prossimi a 1 produca errori assoluti maggiori rispetto agli altri due vettori costituiti da numeri sempre più prossimi allo zero.

Questa differenza tra i vettori non è distinguibile nel condizionamento della matrice simmetrica che rimane indipendentemente da essa costante.

Riferimenti

- *<http://www.comunidelchianti.it/~max/uni/geometria/geo/relazione/node1.html>*
- *Dispense Ricerca Operativa – prof. R.Tadei – Politecnico di Torino*
- *Dispense Ricerca Operativa – prof. P.Zuddas – Università di Cagliari*