ERRORE NEI SISTEMI LINEARI

LIMITE SUPERIORE DELL'ERRORE DEL MEG

Consideriamo il sistema lineare quadrato, non singolare,

$$Ax = b$$

Siano

- x la soluzione esatta;
- x_c la soluzione ottenuta con il MEG e una data aritmetica di macchina.

La domanda è: come possiamo sapere quanto vicina è x_c a x?

Definiamo l'**errore relativo**

$$\boxed{\frac{\|x - x_c\|}{\|x\|}}$$

Per $x_c = x$ l'errore è nullo. Ma l'errore non può essere calcolato. Devo stimarlo.

STIMA DELL’ERRORE USANDO IL RESIDUO

Il residuo associato a x_c è

$$b - Ax_c = r_c$$

E’ facile da calcolare. Inoltre, si può dimostrare che, mediamente, il MEG dà un residuo piccolo, ossia

$$\frac{\|r_c\|}{\|b\|} \ll 1 \quad \text{(ad esempio, } 10^{-9} \text{)}$$

Notare che il residuo associato a x è nullo. Quindi, sembra che a un piccolo residuo relativo debba anche corrispondere una soluzione x_c vicina a x ossia un piccolo errore relativo. Vediamo se effettivamente è così.

STIMA DELL'ERRORE USANDO IL RESIDUO

Riscriviamo $b - Ax_c = r_c$ come

$$Ax_c = b - r_c$$

Questa equazione dice che possiamo interpretare x_c come la soluzione in aritmetica esatta del sistema lineare $Ax = b$ in cui al temine noto b è stata aggiunta la piccola perturbazione $\delta b = -r_c$. Allora, sappiamo che

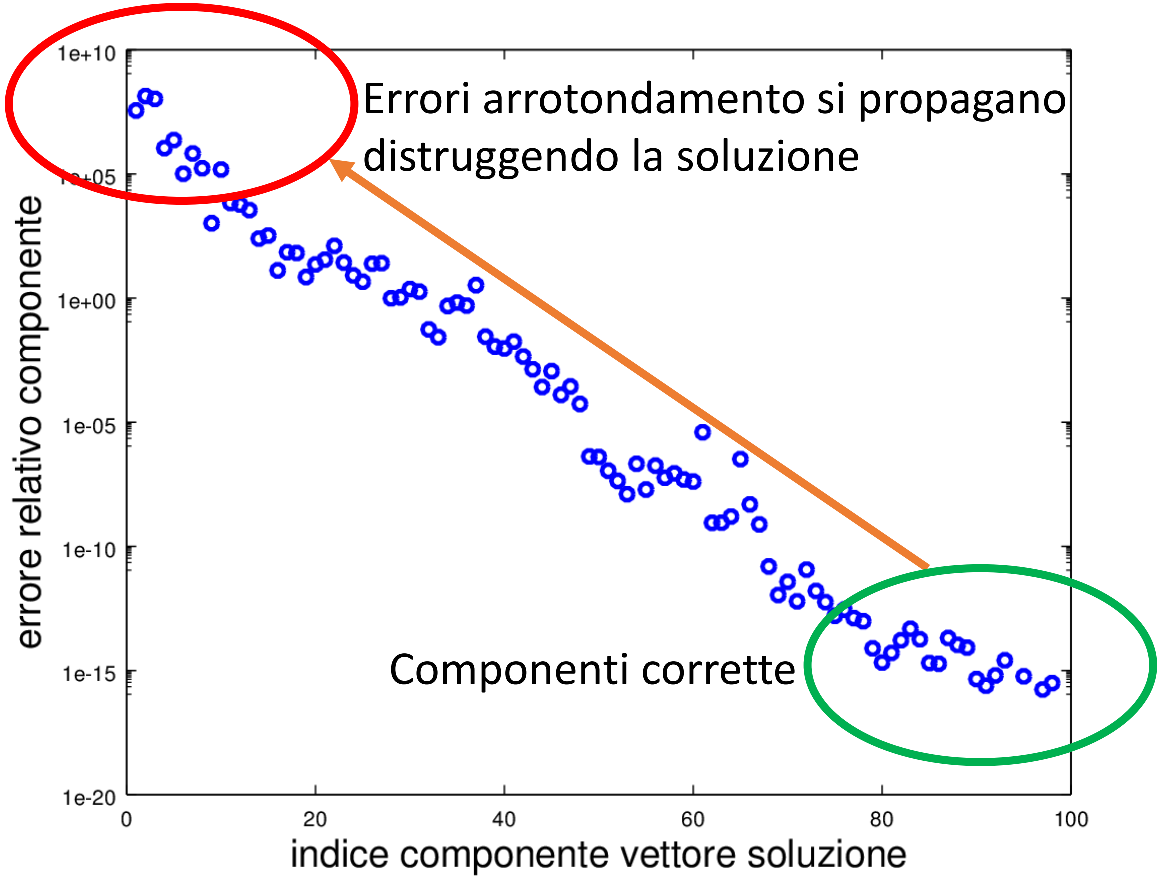
$$\boxed{\frac{\|x - x_c\|}{\|x\|}} \;\boxed{\leq}\; \mathrm{K(A)} \cdot \frac{\|\delta b\|}{\|b\|} = \boxed{\boxed{\mathrm{K(A)}} \cdot \boxed{\frac{\|r_c\|}{\|b\|}}}$$

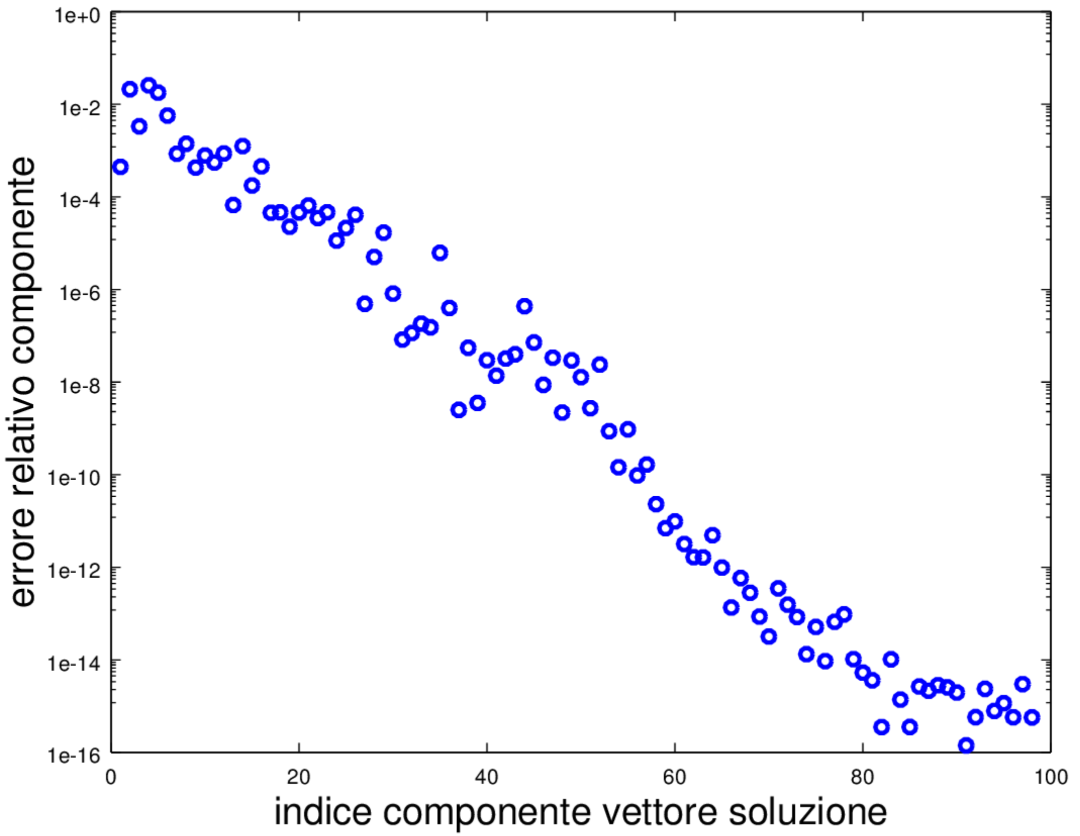
Limite superiore

Errore effettivo

0

SISTEMI TRIANGOLARI ALTI (n=100, random)





$K(U) = 10^{17}$ $\quad || b-Ux ||_2 = 10^{-9}$

$K(U) = 10^{17}$ $\quad || b-Ux ||_2 = 10^{-14}$

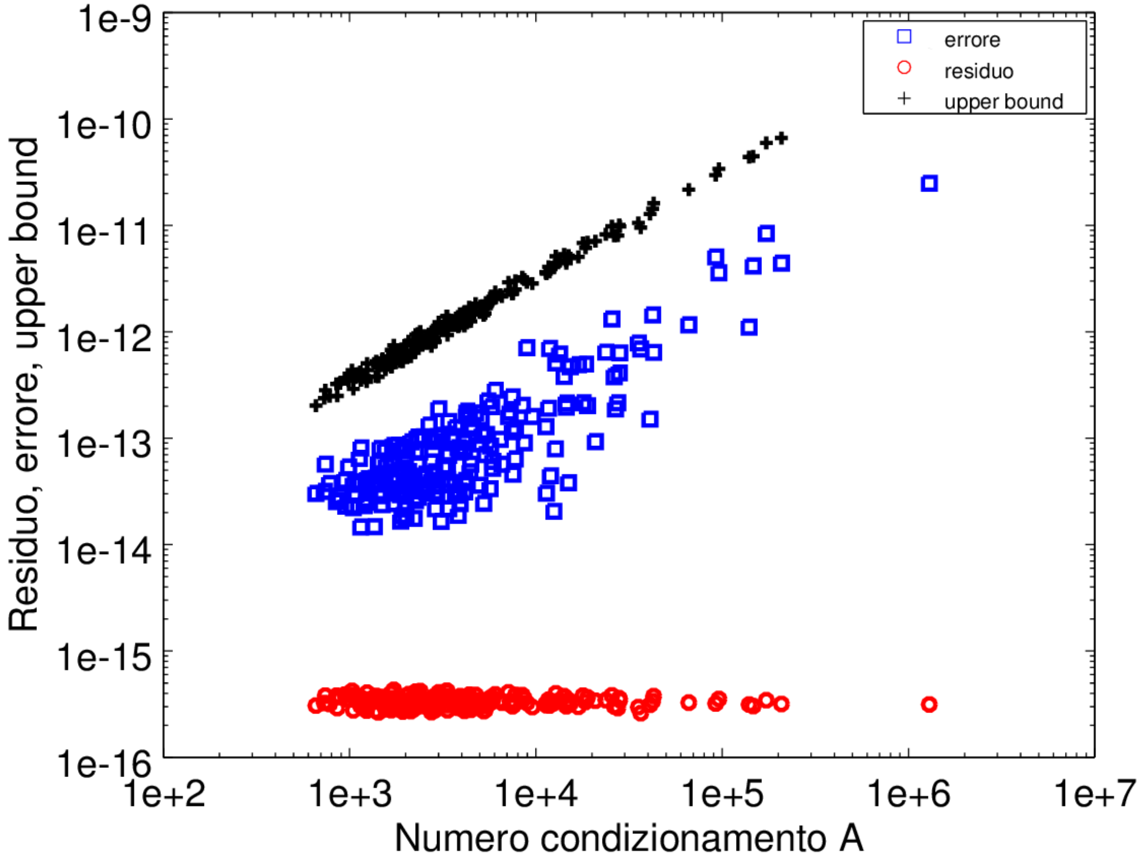
METODO DI SOSTITUZIONE ALL'INDIETRO

```
function x = triup(A,b)
%TRIUP metodo di sostituzione all'indietro
%    X = TRIUP(A,B) risolve il sistema lineare AX=B con
%    A matrice triangolare alta, B vettore colonna usando
%    il metodo di sostituzione all'indietro. La soluzione
%    e' data nel vettore colonna X.
%


n = size(A,1);      % dimensione del sistema lineare
x = zeros( n, 1 ); % vettore della soluzione e' inizializzato a tutti 0


% inizio a trovare le incognite partendo dal fondo
x(n) = b(n)/A(n,n);

for i=(n-1):-1:1
% calcolo x(i), i-esima componente del vettore soluzione
% N.B.: le componenti x(i+1:n) sono note perche' calcolate prima
    x(i) = ( b(i)-A(i,i+1:n)*x(i+1:n)  )/A(i,i);
end
```

METODO DI SOSTITUZIONE ALL'INDIETRO

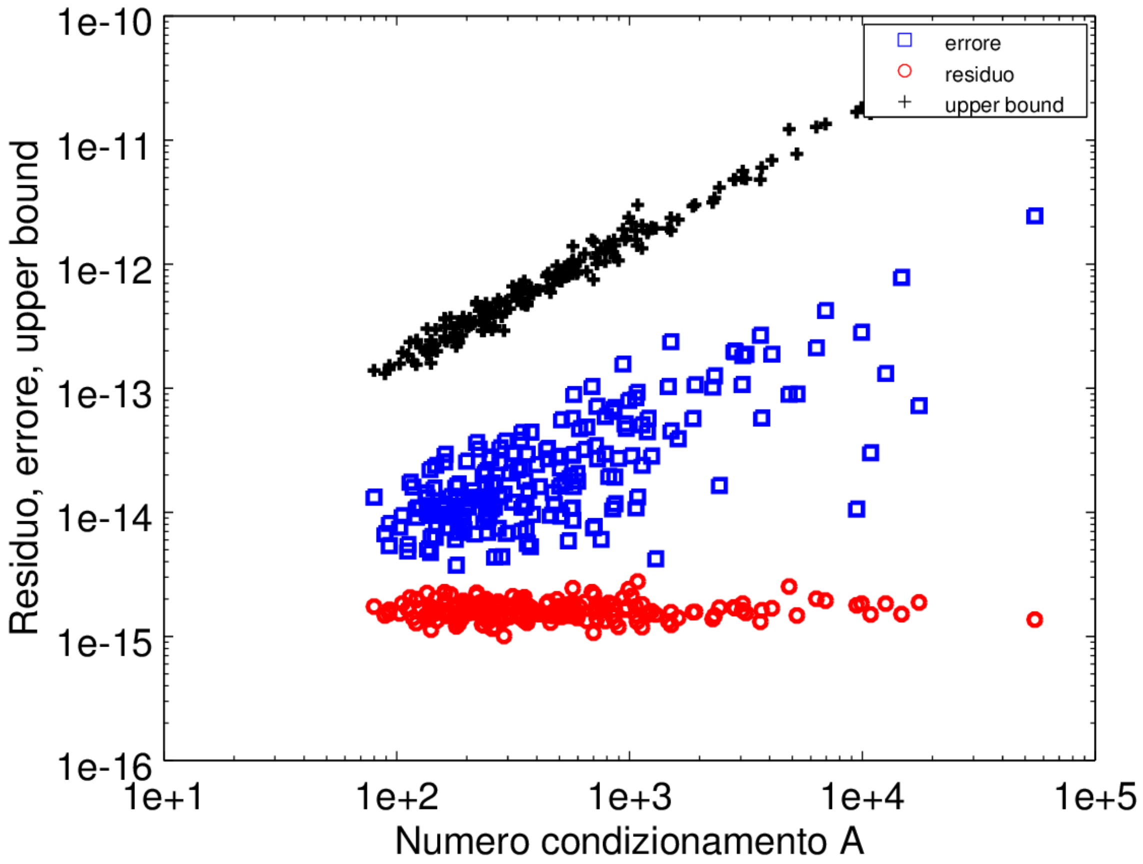
```
% script per provare la function triup

n = 100;        % dimensione del sistema
A = rand(n);    % matrice quadrata con elementi pseudocasuali in (0,1)
U = triu(A);    % prendo la parte alta di A

x = rand(n,1);% genero la soluzione...
b = U*x;        % ...ed il corrispondente termine noto

xc = triup(U,b); % risolvo il sistema lineare e calcolo...
en = norm(xc-x)/norm(x); %... la norma dell'errore relativo

disp( [ 'errore relativo = ', num2str(en)] )
```

LEGAME TRA ERRORE E RESIDUO (MEG con pivoting)



200 prove con sistemi lineari di dimensione n = 100 con elementi pseudocasuali con **distribuzione uniforme in (0,1)**.

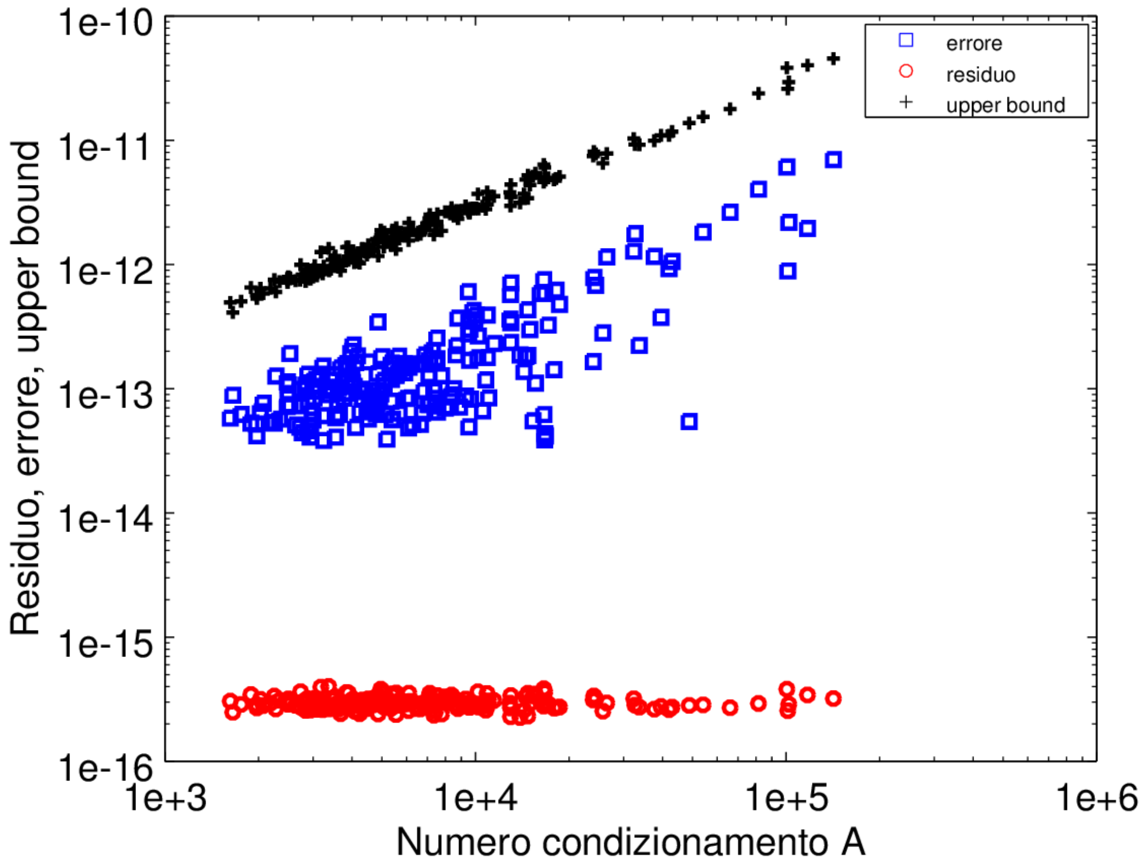
LEGAME TRA ERRORE E RESIDUO (MEG con pivoting)



200 prove con sistemi lineari di dimensione n = 100 con elementi pseudocasuali con **distribuzione normale con media 0 e varianza 1**.

Distribuzione normale
(con media=0, varianza = 1):
randn()

LEGAME TRA ERRORE E RESIDUO (MEG con pivoting)



200 prove con sistemi lineari di dimensione n = 100 con elementi pseudocasuali con **distribuzione normale con media 10 e varianza 9**.

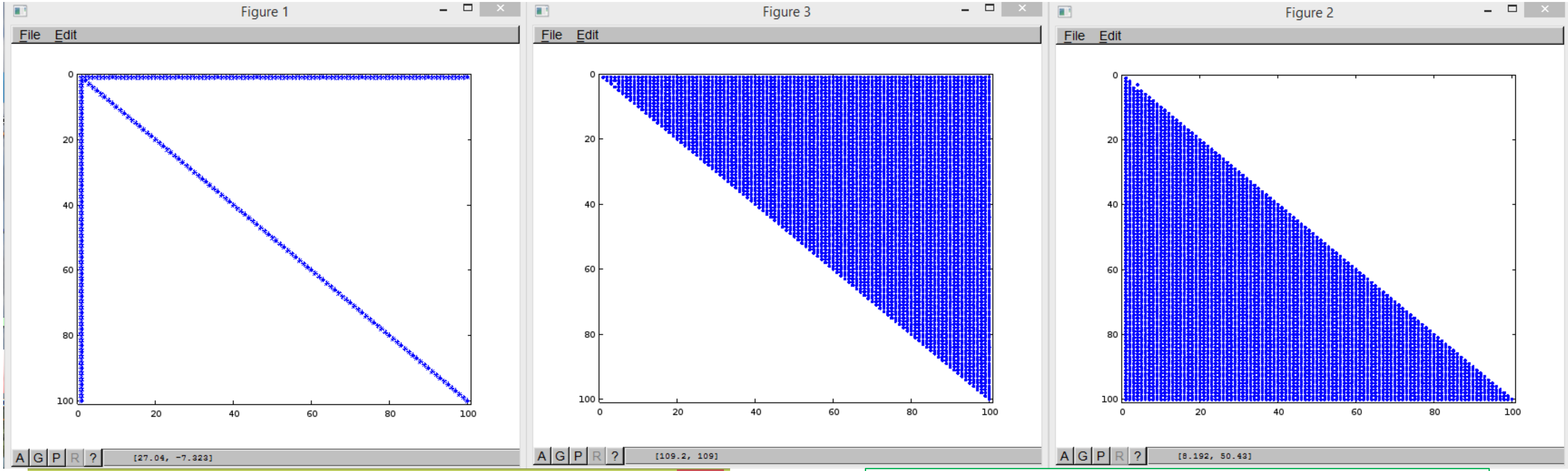
Distribuzione normale
(con media=m, varianza = v):
v*randn()+m

ESERCITAZIONE: ALGORITMO DI THOMAS

$$A = \begin{pmatrix} a_1 & b_1 & & & \\ c_2 & a_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & c_{n-1} & a_{n-1} & b_{n-1} \\ & & & c_n & a_n \end{pmatrix} \quad L = \begin{pmatrix} 1 & & & & \\ l_2 & 1 & & & \\ & \ddots & \ddots & & \\ & & l_{n-1} & 1 & \\ & & & l_n & 1 \end{pmatrix} \quad U = \begin{pmatrix} u_1 & b_1 & & & \\ & u_2 & b_2 & & \\ & & \ddots & \ddots & \\ & & & u_{n-1} & b_{n-1} \\ & & & & u_n \end{pmatrix}$$

$$\boxed{\begin{array}{l} u_1 = a_1 \\ \text{for i=2,...,n} \\ \qquad l_i = c_i / u_{i-1} \\ \qquad u_i = a_i - l_i b_{i-1} \\ \text{end} \end{array}}$$

FILL IN NELLA FATTORIZZAZIONE LU



Octave

File Edit Debug Window Help News

Current Directory: D:\roberto\corsi_2015\calcoloVR\laboratorio

Command Window

```
>> n = 100; % dimensione
>> A = sparse( zeros(n) ); % A in formato sparso
>> A(:,1) = 1;
>> A(1,:) = 1;
>> A = A+ diag( ones(n,1) );
>> spy(A)
>> figure(2), spy(L)
>> figure(3), spy(U)
>>
```

Variables in the current scope:

Attr	Name	Size	Bytes	Class
====	====	====	=====	=====
	A	100x100	3992	double
	L	100x100	80000	double
	U	100x100	80000	double
	n	1x1	8	double