Metodi Matematici per l'Ingegneria

Modulo MAT08

A.A. 2019-2020

Metodi Numerici

Esempi di tesina

Prima tesina

Dato il sistema:

$$\begin{cases} x + 2y - z = 6 \\ 2x + y + z = 3 \\ x + 3z = -2 \end{cases}$$

studiarne il condizionamento in norma 1.

Graficare il massimo errore relativo del vettore delle soluzioni in funzione di un errore pari a 1%, 5%, 10%, 20% sul vettore dei termini noti.

Seconda tesina

Si usi il metodo di Newton-Raphson per determinare con una precisione di 10^{-3} la più piccola radice dell'equazione:

$$e^{x} - 2\,x^{2} = 0$$

Soluzione Tesina 1

1. Considerazioni iniziali

Poche righe con:

Generalità sul metodo adottato, condizioni di applicabilità, etc.

2. Codice Matlab

```
clear
close all

A = [1 2 -1;
     2 1  1;
     1 0  3];

B = [6 3 -2]';

K = norm(A,1) * norm(inv(A),1)

X =A\B
```

Nota: inserire commenti!

```
B1 = B * 1.01;
X1 =A\B1

dB_on_B = norm(B1-B,1)/norm(B,1)

max_dX_on_X1 = K * dB_on_B

%%%%%%%%%%%%%%%%%%%

B2 = B * 1.05;
X2 =A\B2

dB_on_B = norm(B2-B,1)/norm(B,1)

max_dX_on_X2 = K * dB_on_B
```

```
B3 = B * 1.1;
X3 =A\B3

dB_on_B = norm(B3-B,1)/norm(B,1)

max_dX_on_X3 = K * dB_on_B

%%%%%%%%%%%%%%%%%%%

B4 = B * 1.2;
X4 =A\B4

dB_on_B = norm(B4-B,1)/norm(B,1)

max_dX_on_X4 = K * dB_on_B

figure
plot([max_dX_on_X1 max_dX_on_X2 max_dX_on_X3 max_dX_on_X4])
title('Grafico dell''errore'),
xlabel(' casi 1, 2, 3, e 4'), ylabel('massimo errore relativo su X')
```

3. Risultati ottenuti (con commenti)

```
>> prova1

K =

   9.999999999999998

X =

     1
     2
    -1

X1 =

   1.010000000000001
   2.020000000000000
  -1.010000000000000

dB_on_B =

   0.010000000000000

max_dX_on_X1 =

   0.100000000000001
```

```
X2 =

   1.050000000000000
   2.100000000000000
  -1.050000000000000

dB_on_B =

   0.050000000000000

max_dX_on_X2 =

   0.500000000000001
```

```
X3 =

   1.100000000000000
   2.200000000000000
  -1.100000000000000

dB_on_B =

   0.100000000000000

max_dX_on_X3 =

   1.000000000000001
```

```
X4 =

   1.200000000000000
   2.400000000000000
  -1.200000000000000


dB_on_B =

   0.200000000000000


max_dX_on_X4 =

   1.999999999999999

>>
```

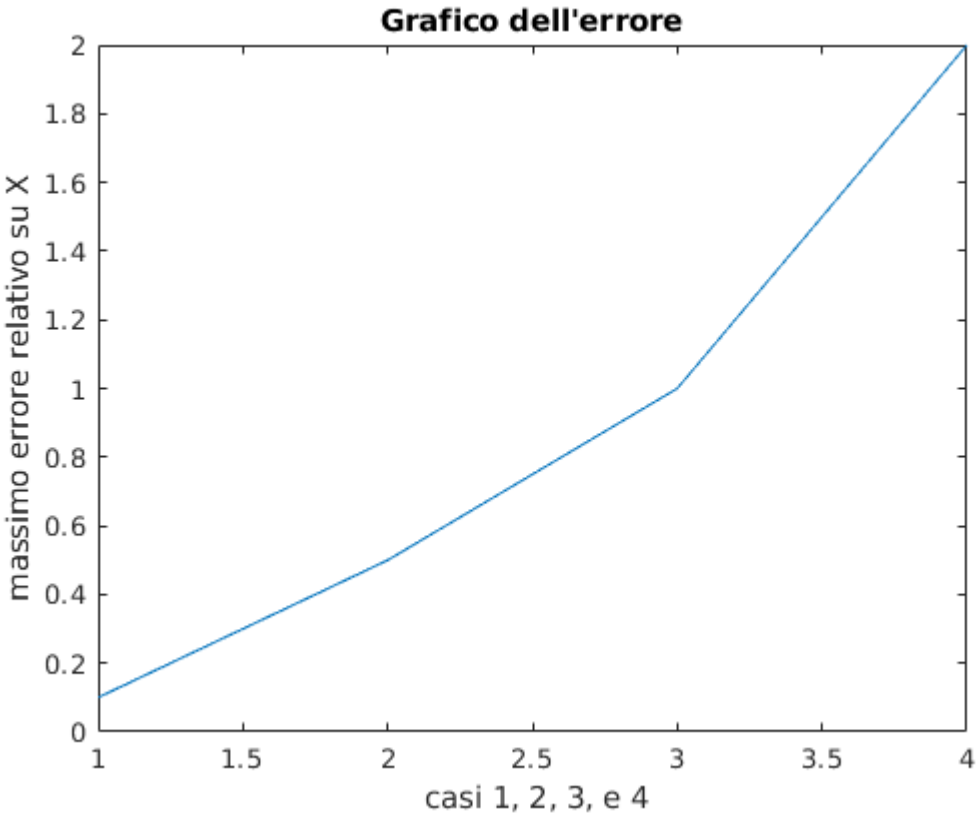


Grafico dell'errore
massimo errore relativo su X
casi 1, 2, 3, e 4
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
1
1.5
2
2.5
3
3.5
4

4. Conclusioni

Due righe con le conclusioni che si possono ricavare dai risultati sperimentali

Soluzione Tesina 2

1. Considerazioni iniziali

Poche righe con:

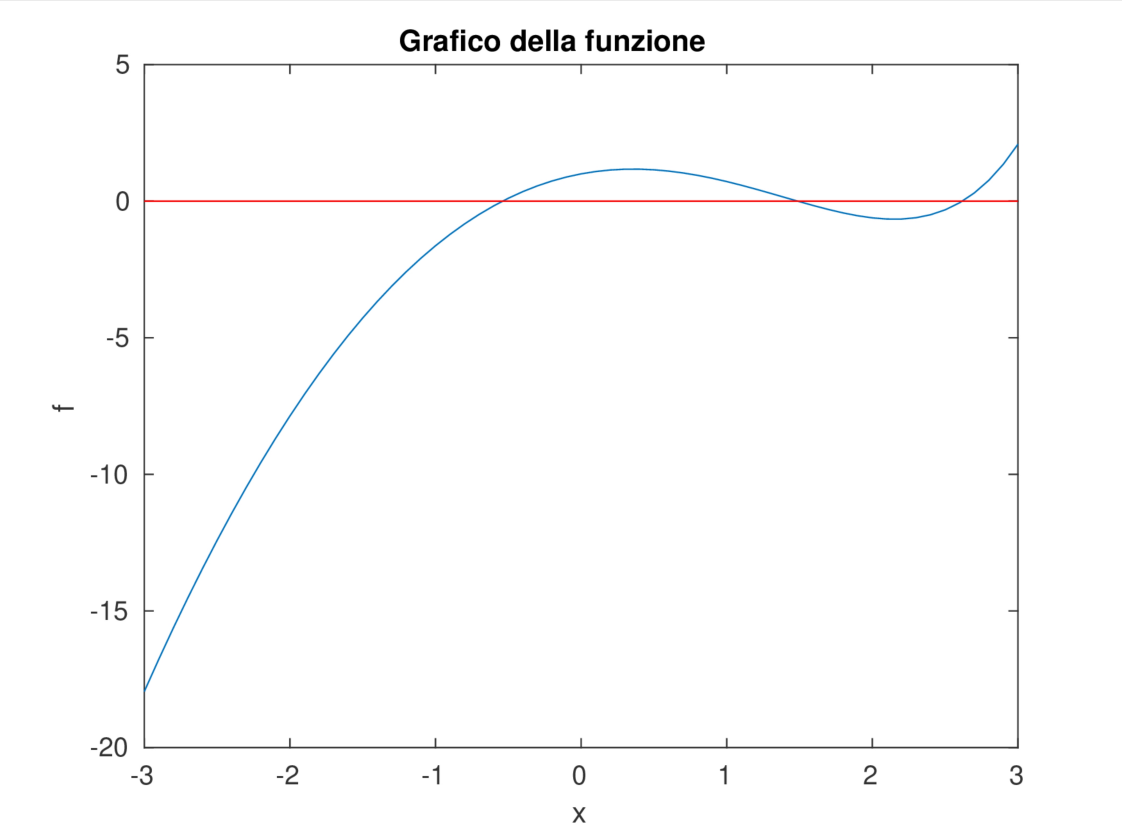
Generalità sul metodo adottato, condizioni di applicabilità, etc.

2. Codice Matlab

```
clear
close all

%grafico generale della funzione

x =[-3:.1:3];

f = exp(x)-2*x.^2;

figure
plot(x,f)
hold on
plot(x,zeros(1,length(x)),'r')
title('Grafico della funzione'), xlabel('x'), ylabel('f')
pause
```

```
f = input('introduci la funzione f = ')

a = input('introduci l''estremo inferiore dell''intervallo a = ')

b = input('introduci l''estremo superiore dell''intervallo b = ')

np = input('numero di punti in cui suddividere l''intervallo np = ')

if isempty(np)
  np=10;
end

x=linspace(a,b,np);

y=f(x);

pos = find(abs(diff(sign(y))));

intervallo(1,:) = x(pos);

intervallo(2,:) = x(pos+1);

fprintf('l''estremo inferiore e'' %6.5f\nl''estremo superiore e''%6.5f\n', intervallo)
```

3. Risultati ottenuti



```
>> prova2
introduci la funzione f = @(x)[exp(x)-2*x.^2]

f =

function_handle with value:

    @(x)[exp(x)-2*x.^2]

introduci l'estremo inferiore dell'intervallo a = -1

a =

    -1

introduci l'estremo superiore dell'intervallo b = 0

b =

    0

numero di punti in cui suddividere l'intervallo np =

np =

    []

l'estremo inferiore e'' -0.55556
l'estremo superiore e'-0.44444
```

```
% Nota: e^x−2x^2=0  e  df = exp(x)-4*x (calcolabile mediante calcolo simbolico)

[xn,n_iter,err] = newton_fun(@(x)[exp(x)-2*x.^2],@(x)[exp(x)-4*x],-.5,.5*10^(-3))

n_iter =

     1

xn =

  -0.540870672023623

err =

   0.040870672023623

n_iter =

     2

xn =

  -0.539835944080133

err =

%    0.001034727943491
```

```
n_iter =

     3

xn =

  -0.539835276903097

err =

     6.6717703528507 14e-07

xn =

  -0.539835276903097

n_iter =

     3

err =

     6.6717703528507 14e-07
```

4. Conclusioni

Due righe con le conclusioni che si possono ricavare dai risultati sperimentali

Es: bastano tre iterazioni per ottenere un errore di …

con un tempo di calcolo di (se si usa tic e toc)

etc.