PROVA PRATICA di CALCOLO NUMERICO

Prof. S. De Marchi

`Verona, 18 luglio 2006`

Il candidato dovrà scrivere su **ogni** foglio il cognome, nome, numero di matricola. I fogli saranno forniti da chi fa assistenza. **Consegnare fogli leggibili!**. Consegnare un floppy o inviare una email con ricevuta di lettura a **stefano.demarchi@univr.it** contenente tutti i files/scripts usati per produrre i risultati

NOTA: non allegate immagini in formato Matlab .fig ma .jpg o .eps.

1. Si consideri la funzione $f(x) = x^2 - 2 + \sin(\exp(x))$ sull'intervallo $[-2, 2]$.

   (a) si determini la radice positiva α usando il metodo di Newton.

   (b) si determini la radice negativa β usando il seguente metodo di iterazione, di Steffensen,

   $$x_{i+1} = x_i - \frac{f(x_i)^2}{f(x_i + f(x_i)) - f(x_i)} .$$

   Dire quante iterazioni vengono fatte con i due metodi. In tutti e due i casi usare $tol = 1.e-6$.

2. Sia `A=wilkinson(7)` la matrice di ordine 7 di Wilkinson. Determinare gli autovalori estremi (min e max) e i corrispondenti autovettori usando un opportuno metodo numerico.

3. Si considerino i valori di tabella relativi allo sforzo σ (misurato in MPa, ove $1MPa = 100\,N/cm^2$) applicato ad un tessuto provocandone una deformazione τ

σ	0	0.06	0.14	0.25	0.31	0.47	0.6	0.7
τ	0	0.08	0.14	0.2	0.23	0.25	0.28	0.29

   Stimare la deformazione corrispondente a $\sigma = 0.9\,MPa$ mediante un'approssimazione lineare ai minimi quadrati.

Tempo: **2,5 ore**.

SOLUZIONI

1. Osservo che le radici di $f(x)$ in [-2,2] sono due: $\alpha > 0$ e $\beta < 0$ visibili facendo un

   `ezplot('x.^2-2+sin(exp(x))',-2,2); grid.`

   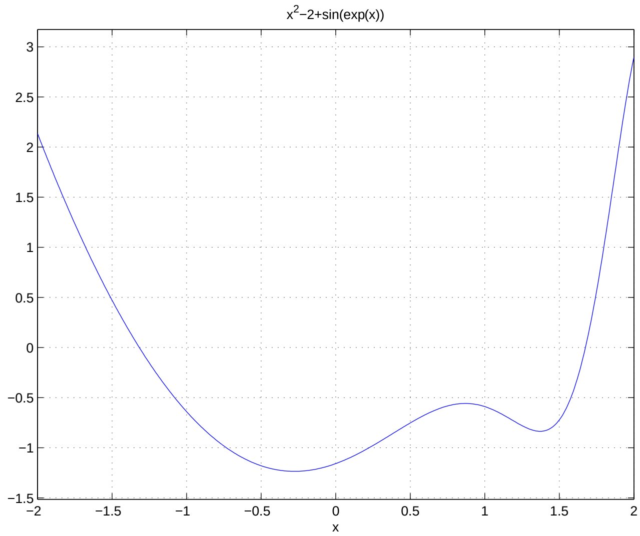

   Figure 1: $f(x)$ nell'intervallo richiesto

   Le determiniamo, come richiesto, nel seguente modo: la radice α con il metodo di Newton e la radice β con il metodo di Steffensen.

   Il metodo iterativo di Newton si implementa partendo da un valore iniziale x_0 e costruendo la successione

   $$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}\,,$$

   dove $f'(x) = 2x + \cos(\exp(x))\exp(x)$.

   Il metodo iterativo di Steffensen si implementa ancora una volta partendo da un valore iniziale x_0 e costruendo la successione

   $$x_{i+1} = x_i - \frac{f(x_i)^2}{f(x_i + f(x_i)) - f(x_i)}\,.$$

   In entrambi i casi ci si arresterà quando l'errore relativo risulta essere minore della tolleranza assegnata, $1.e-6$, ovvero quando sarà raggiunto un numero massimo (fissato) di iterazioni, $kmax$.

   Una possibile implementazione è la seguente.

```
clear
% Primo esercizio compito del 18/7/06
% Parte (a)
% ----------------------------------------------------
% Metodo di Newton per la ricerca di zeri di funzione
%------------------------------------------------------
x0=input('Valore iniziale per Newton '); itermax=100; tol=1.e-6;

[f0,f0d]=fund1(x0); % fund1: vedi sotto....
x1=x0-f0/f0d;
iter=1;

while abs(1-x0/x1)> tol & iter<=itermax,
    x0=x1;
    [f0,f0d]=fund1(x0);
    x1=x0-f0/f0d;
    iter=iter+1;
end
iter-1
disp(' Lo zero cercato con Newton e'' '); x0

% Parte (b)
% ----------------------------------------------------
% Metodo di Steffensen per la ricerca di zeri di funzione
%------------------------------------------------------
x0=input('Valore iniziale per Steffensen ');
x1=funSteffensen(x0);
iter=1;

while abs(1-x0/x1)> tol & iter<=itermax,
    x0=x1;
    x1=funSteffensen(x0);
    iter=iter+1;
end
iter-1
disp(' Lo zero cercato,con Steffensen, e'' '); x0


function [y,yd]=fund1(x)
y=x^2-2+sin(exp(x));
yd=2*x+cos(exp(x))*exp(x); %derivata di f
return
```

```
function f=funSteffensen(x)
% funzione iterativa di Steffensen che fa uso della
% funzione fund1
f1=fund1(x);
f2=fund1(x+f1);
f=x-f1^2/(f2-f1);
return

>> esame18luglio2006esI
Valore iniziale per Newton 2

ans =

     4

 Lo zero cercato con Newton e'

x0 =

    1.6757

Valore iniziale per Steffensen -2

ans =

     5

 Lo zero cercato,con Steffensen, e'

x0 =

   -1.3173

>>
```

2. La matrice `a=wilkinson(7)` è tridiagonale simmetrica. Se dovessimo determinare tutti i suoi autovalori, potremo usare il metodo di Jacobi o quello delle successioni di Sturm.

   Poiché ci viene chiesto di determinare solo quelli estremi, ovvero λ_M di modulo massimo e λ_m di modulo minimo, allora basterà usare il metodo delle potenze, per determinare λ_M

e il corrispondente autovettore e il metodo delle potenze inverse per determinare λ_m e il corrispondente autovettore. In entrambi i casi si è usata $tol = 1.e - 6$.

Entrambi i metodi sono stati visti durante le lezioni di laboratorio quindi diamo solo i risultati.

```
>> metPotenze
iterazioni fatte =

k =

    13

autovettore =

y1 =

    0.5400
    0.4116
    0.1848
    0.0984
    0.1848
    0.4116
    0.5400

autovalore di modulo massimo =

lambda1 =

    3.7616

Calcolo del residuo A x-lambda x =

ans =

  1.0e-003 *

    0.3539
   -0.2243
   -0.4346
   -0.3679
```

```
   -0.4346
   -0.2243
    0.3539

>> metPotenzeInverse
iterazioni fatte =

k =

    36

autovettore =

y1 =

   -0.1494
    0.4082
   -0.5577
    0.0000
    0.5577
   -0.4083
    0.1494

autovalore di modulo minimo =

ans =

    0.2679

Calcolo del residuo A x-lambda x =

ans =

  1.0e-004 *

   -0.2195
    0.4395
   -0.2216
   -0.4348
   -0.2213
    0.4393
   -0.2194
```

```
>>
```

3. Questo esercizio si ricollegava all'esercizio 2 del compito del 21 giugno 2006. In pratica si tratta di determinare il polinomio di primo grado, $p(\sigma) = a\sigma + b$ che approssima i dati le coppie $(\sigma_i, \tau_i), i = 1, ..., 8$ della tabella data.

   Quindi nel codice qui sotto ripotato, il grado richiesto è 1.

   Determinati i coefficienti a, b per rispondere alla richiesta basta calcolare $p(0.9)$. Pertanto $p(0.9) =$ sarà la deformazione dovuta allo sforzo $\sigma = 0.9MPa$.

```
clear

%-------------------------------------------
% Esercizio 3 del compito del 18 luglio 2006
%-------------------------------------------
%
m=input('grado del polinomio approssimante = ');


x=[0 0.06 0.14 0.25 0.31 0.47 0.6 0.7];

y=[0 0.08 0.14 0.2 0.23 0.250.28 0.29];

% Costruiamo il sistema delle equazioni normali
% per determinare il polinomio di grado m
k=0;
for i=1:m+1,
    for j=i:m+1,
        a(i,j)=sum(x.^(j+k-1));
        a(j,i)=a(i,j);
    end
   k=k+1;
   % termine noto
   if(i==1)
      b(1)=sum(y);
  else
      b(i)=(x.^(i-1))*y';
  end
end

%coefficienti del polinomio
```

```
c=inv(a)*b';

% plot dei punti e del polinomio approssimante
plot(x,y,'r*');
hold on
xx=min(x):0.01:max(x);
for i=1:length(xx),
    yy(i)=0;
for k=1:m+1,
    yy(i)=yy(i)+c(k)*xx(i)^(k-1);
end
end
plot(xx,yy,'b-.') hold off

disp('Valore corrispondente a \sigma=0.9 ');

tau=0; for i=0:m,
    tau=tau+c(i+1)*(0.9)^i;
end

tau

>> esame18luglio2006esIII
grado del polinomio approssimante = 1

Valore corrispondente a \sigma=0.9

tau =

    0.4021

>>
```

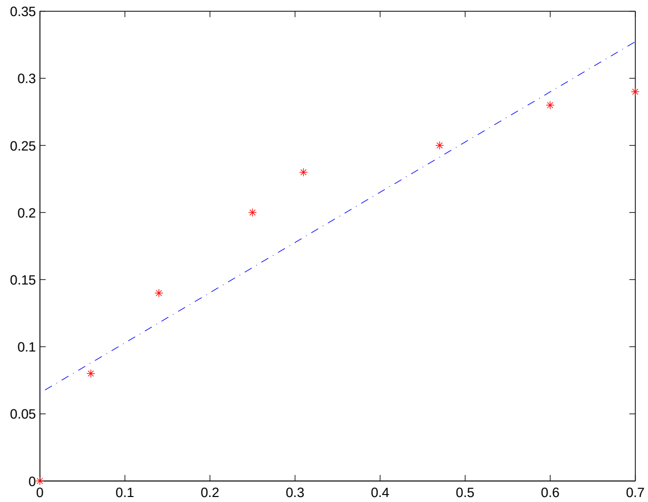

Figure 2: Punti e approssimante lineare ai minimi quadrati