PROVA PRATICA di CALCOLO NUMERICO

Prof. S. De Marchi

`Verona, 15 settembre 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 con tutti i files/scripts usati per produrre i risultati oppure inviare una email a `stefano.demarchi@univr.it`.

NOTA: allegate immagini in formato .jpg o .eps.

1. Si considerino le seguenti due funzioni

$$g_1(x) = \frac{rx}{1+kx}, \;\; g_2(x) = \frac{rx^2}{1+\left(\frac{x}{k}\right)^2} \; .$$

   Si prendano $r = 3, \; k = 1$.

   (a) determinare i punti fissi di $g_1(x)$ e $g_2(x)$;

   (b) dire quale dei punti fissi positivi può essere effettivamente calcolato con il metodo iterativo $x_{i+1} = g(x_i), \; i \geqslant 0$;

   (c) calcolare, a partire da $x_0 = 1$, il punto fisso più grande in modulo di $g_1(x)$ e $g_2(x)$, determinando anche il numero di iterazioni necessarie e l'errore.

   In entrambi i casi usare $tol = 1.e-6$ e test sull'errore relativo.

2. Si consideri la matrice

$$A = \left(\begin{array}{cccc} 3 & \frac{2}{3} & 2 & \frac{1}{3} \\ 0 & 2 & -1 & \frac{1}{2} \\ 0 & 0 & -\frac{5}{3} & -\frac{2}{3} \\ \frac{1}{2} & 0 & 1 & 2 \end{array} \right) \; . \qquad (1)$$

   e il vettore $\mathbf{b}$ costruito cosiccé la soluzione sia $x = ones(4,1)$. Partendo ora da un vettore iniziale $x^{(0)}$ si implementi il metodo di Jacobi accelerato

$$x^{(k+1)} = (I - \omega D^{-1}A)x^k + \omega D^{-1}\mathbf{b} \quad k \geqslant 0 \, ,$$

   con D la diagonale di A. Al variare di $\omega \in]0,2[$ si stimi il valore del parametro ottimale ω^* per la convergenza del metodo alla soluzione usando $tol = 1.e-4$ e test sulla norma 2 tra due soluzioni consecutive.

3. Si calcoli il numero minimo m di intervalli necessari per approssimare a meno di 10^{-4} l'integrale

$$\int_0^{\pi} \mathrm{e}^x \cos x dx$$

   usando la formula di Simpson composita. Quindi si verifichi la cosa computazionalmente.

Tempo: **3 ore**.

SOLUZIONI

1.

```
clear
%-----------------------------------------
% primo esercizio del 15 settembre 2006
%-----------------------------------------

%
% con k=1 e r=3
% g_1(x)=3x/(1+x); g_2(x)=3x^2/(1+x^2);
%
% (a) Punti fissi
%    risolvo g_i(x)=x, i=1,2
% si ottiene
% \xi_1=0, \xi_2=2 risolvendo g_1(x)=x
%
% \nu_1=0, \nu_2=(3+sqrt(5))/2 \sim 2.62
%          \nu_3=(3-sqrt(5))/2 \sim 0.38


% (b)
% per rispondere determino
% g_1^\prime(x)= 3/(1+x)^2
% g_2^\prime(x)= 6x/(1+x^2)^2
% e le valuto nei punti fissi. Se risultano essere in modulo <1 allora
% posso dire che il metodo converge
%
% g_1^\prime(\xi_1)=3 >1 non converge
% g_1^\prime(\xi_2)=3/5 < 1 converge
%
% g_2^\prime(\nu_1)=0 converge
% g_2^\prime(\nu_2)=0.25 < 1  converge
% g_2^\prime(\nu_3)=1.74 > 1 non converge


% (c)
% Basta implementare i due metodi iterativi per calcolare
% \xi_2 e \nu_2

x=input('Valore iniziale = ');
x1=3*x/(1+x);
e(1)=abs(x-x1); i=1;
```

```
while(abs(x-x1)>1.e-6* abs(x1) & i<100)
    x=x1;
    x1=3*x/(1+x);
    i=i+1;
    e(i)=abs(x-x1)/abs(x1);
end semilogy(1:length(e),e,'--'); legend('errore primo metodo');

figure(2) clear x,x1,e;

x=input('Valore iniziale = ');
x1=3*x^2/(1+x^2);
e(1)=abs(x-x1);
i=1;
while(abs(x-x1)>1.e-6*abs(x1) & i<100)
    x=x1;
    x1=3*x^2/(1+x^2);
    i=i+1;
    e(i)=abs(x-x1)/abs(x1);
end semilogy(1:length(e),e,'--');
legend('errore secondo metodo'); i
x

>> esame15settembre2006esI
Valore iniziale = 1

x1 =

    2.0000

Valore iniziale = 1

i =

    18


x =

    2.6180
```

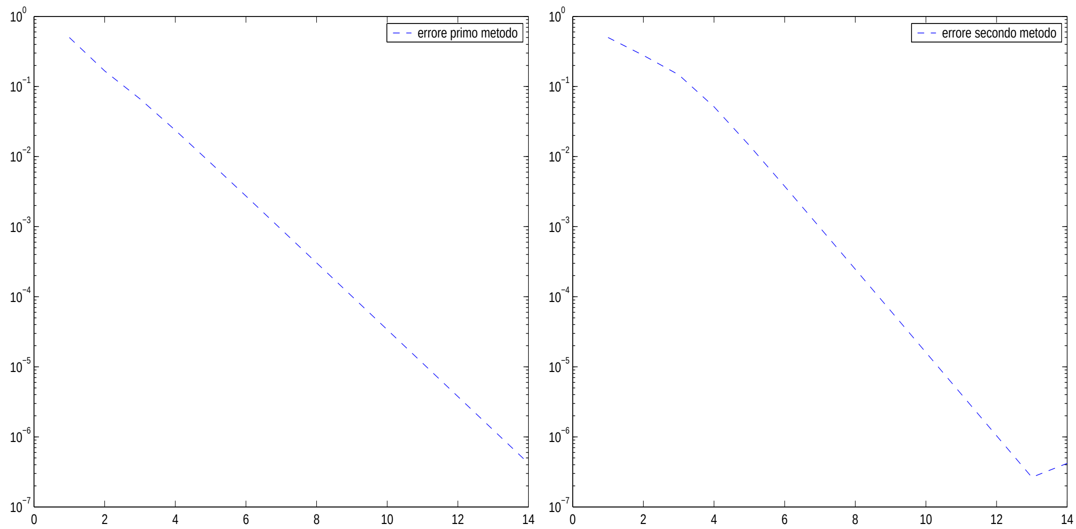


Figure 1: Grafico dell'errore relativo al metodo iterativo dell'Esercizio 1 applicato per calcolare le radici positive.

2.

```
clear
%-----------------------------------------
% secondo esercizio del 15 settembre 2006
%-----------------------------------------
%matrice
A=[3 2/3 2 1/3; 0 2 -1 1/2; 0 0 -5/3 -2/3; 1/2 0 1 2]; n=4;
b=A*ones(n,1); tol=1.e-4; kmax=100;

d=diag(diag(A));
w=linspace(0.1,2,50);
for j=1:length(w)
x0=zeros(n,1);
d1=inv(d);
P=eye(n)-w(j)*d1*A;
b1=w(j)*d1*b;
x1=P*x0+b1; k=1;
while norm(x1-x0,2) > tol & k < kmax,
    x0=x1;
    x1=P*x0+b1;
    k=k+1;
end
nn(j)=k;
```

```
end

plot(1:length(nn), nn,'-o')

[mm,indm]=min(nn);
disp('il minimo richiesto e'' ');
w(indm)


>> esame15settembre2006esII
il minimo richiesto e'

ans =

    0.8755
```

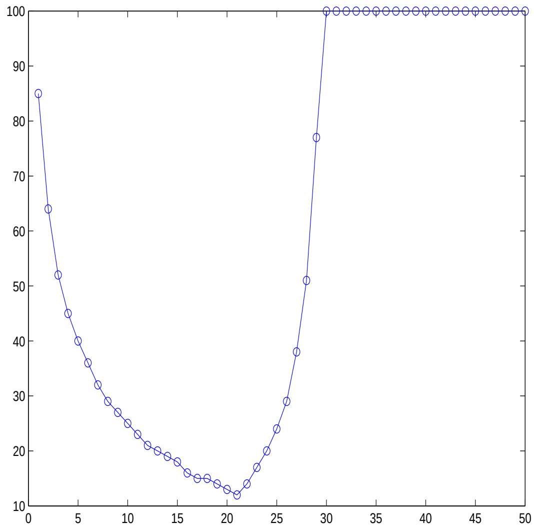


Figure 2: Grafico del numero di iterazioni al variare di ω. Le ascisse indicano l'indice corrispondente all'array di valori di ω che sono 50. Per valori di ω maggiori di 1 (ovvero indice maggiore di 20) si raggiungono sempre il numero max di iterazioni consentite (100).

3. ```
   clear
   %-----------------------------------------
   % terzo esercizio del 15 settembre 2006
   %-----------------------------------------

   % Per determinare il numero massimo m di intervalli
   ```
   ```

```
% dobbiamo usare la formula dell'errore
%
% detto h=(b-a)/m
%
% E_S=-h^4/16 (b-a)/180 f^(4)(\xi)
%
% da cui volendo che l'errore in modulo sia minore di 1.e-4
% posto M=maxf^(4) su [a,b] ed essendo h=(b-a)/(2*m) si
% ottiene la stima
% da cui m \ge ceil((M*pi^5/(1.e-6*180*16))^(1/4))

% f(x)=e^x cos(x)
% f^4(x)=-4e^xcos(x)

x=0:0.001:pi;
y4=-4*exp(x).*cos(x);
plot(x,y4,'-.');
grid;
M=max(abs(y4))
mm=ceil((M*pi^5/(1.e-6*180*16))^(1/4));


% Test
n=mm;
a=0; b=pi; h=(b-a)/n;
x=linspace(a,b,n+1); %punti equispaziati.
xx=a:0.01:b;
yy=funQ(xx);
realValue=quadl(@funQ,a,b,1.e-6);

fSc=funQ(x);

plot(xx,yy,'-.g',x,fSc,'ob'), hold on
     fSc(2:end-1)=2*fSc(2:end-1);
     ValSc=h*sum(fSc)/6;
     x=linspace(a+h/2,b-h/2,n);
     fSc=funQ(x);
     plot(x,fSc,'or')
     ValSc=ValSc+2*h/3*sum(fSc)
     title('Quadratura composita di Simpson e relativi nodi');
     disp('Errore assoluto')
     erroreT=abs(realValue-ValSc)
```

```
function y=funQ(x)
y=exp(x).*cos(x);
return


>> esame15settembre2006esIII

M =

   92.5079

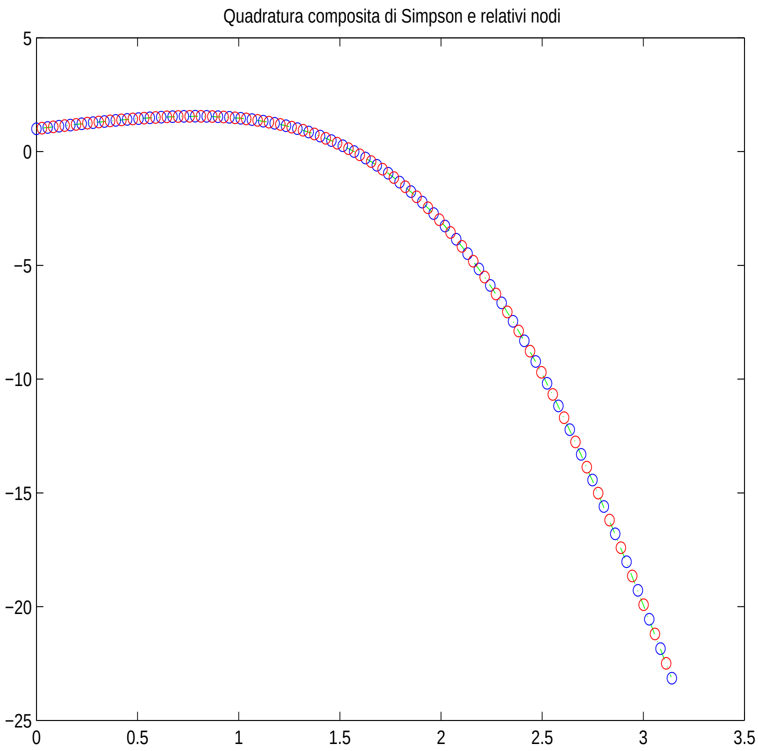

ValSc =

  -12.0703

Errore assoluto

erroreT =

  1.6602e-007
```


Quadratura composita di Simpson e relativi nodi
5
0
−5
−10
−15
−20
−25
0
0.5
1
1.5
2
2.5
3
3.5