Esame di Calcolo Numerico

Laurea in Statistica ed Informatica
Laurea Magistrale in Astronomia
Prof. S. De Marchi
`Padova, 16 dicembre 2010`

Il candidato dovrà scrivere su **ogni** foglio il cognome, nome, numero di matricola. Consegnare fogli leggibili!. Inviare poi tutti i files a `demarchi@math.unipd.it` **NOTA: non allegate immagini in formato Matlab .fig ma .jpg o .eps.**

1. Si consideri la funzione $f(x) = \frac{1}{x+2} - \frac{1}{x^2+1}$ di cui si vogliamo trovare gli zeri.
   (a) Individuare gli intervalli separatori delle due radici reali α_1 e α_2 di f, che denoteremo con I_{α_1} e I_{α_2}. Perchè $\alpha_1 + \alpha_2 = 1$?
   (b) Scelto I_{α_1}, si determini a priori quante iterazioni sono necessarie per determinare α_1 con il metodo di bisezione con `tol=1.e-8`.
   (c) Calcolare numericamente α_1 e α_2 ciascuna con un metodo iterativo convergente a meno di `tol=1.e-8`, indicando anche l'ordine di convergenza approssimato di ciascun metodo.

2. Assegnati i punti $x_0 = 0$, $x_1 = \frac{1}{2}$, $x_2 = 1$ e la funzione $f(x) = \frac{1}{1+x^2}$
   (a) Determinare il polinomio $p_2(x)$ in forma di Lagrange che interpola $f(x)$ nei punti assegnati e se ne plottino i rispettivi grafici
   (b) Dare una maggiorazione dell'errore d'interpolazione di $f(x)$ con $p_2(x)$
   (c) Approssimare $\int_0^1 f(x)dx$ con $\int_0^1 p_2(x)dx$ e calcolarne l'errore assoluto.
   (d) Quanti punti si dovrebbero considerare per avere un errore $\leq 10^{-4}$ con il metodo dei Simpson composito?

Tempo: **2 ore**.

SOLUZIONI

Primo esercizio

```
% --------------
% Esercizio 1
%---------------
close all
clear all
disp('Primo esercizio')
% a) Gli zeri si cercano per x>-2 (in x=-2, c'e' una singolarita').
% La funzione di cui si cercano gli zeri
% e' in effetti il polinomio f(x)=x^2-x-1 che ha
% due radici reali in [-1,2] che si trovano con la formula
% \alpha_{1,2}= (1\pm sqrt{5}) /2, ovvero \alpha_1 \approx -0.6 e
% \alpha_2 \approx 1.6. Come intervalli separatori possiamo
% prendere I_{\alpha_1}=[-1,0] e I_{\alpha_2}=[1,2].

  x=linspace(-1,2,1000);
  y=x.^2-x-1;
  plot(x,y); grid;
  pause(1)

% Ricordando che la somma delle radici
% di un trinomio (monico) e' il coefficiente di x (a meno del
% segno), questo dice che la somma delle radici e` 1.
%
% b) La disuguaglianza da risolvere e`
%    k>floor(log2(1.e8)-1)=25 => k=26

% c) Possiamo considerare i metodi
%    aventi funzione d'iterazione
%    g_1(x)=-sqrt(x+1) e g_2(x)=sqrt(x+1)

% verifichiamo che le funzioni d'iterazione hanno derivata
% in modulo minore di 1
figure(2)
x1=linspace(-0.7,0,100); x2=linspace(1,2,100);
y1=-1./(2*sqrt(x1+1)); y2=1./(2*sqrt(x2+1));
plot(x1,y1,'r', x2,y2,'b');
pause(1)
```

```
disp('OK');

[x1, x, niter, flag]=MetIterazioneFunz(inline('-sqrt(x+1)'), -0.7, 1.e-8, 100);
x1
ordine1=log(abs(x(end-2)-x1))/log(abs(x(end-3)-x1))

[x2, x, niter, flag]=MetIterazioneFunz(inline('sqrt(x+1)'), -0.7, 1.e-8, 100);
x2
ordine2=log(abs(x(end-2)-x2))/log(abs(x(end-3)-x2))

% Primo esercizio
% OK
%
% x1 =
%
%    -0.6180
%
%
% ordine1 =
%
%     0.9295
%
%
% x2 =
%
%     1.6180
%
%
% ordine2 =
%
%     1.0826

function [x0, x, niter, flag]=MetIterazioneFunz(g, x0, tol, kmax)
k=1; x(k)=x0; k=k+1; x(k)=g(x(k-1)); flag=1;
while abs(x(k)-x(k-1)) > tol*abs(x(k)) & k <= kmax
x(k-1)=x(k);
k=k+1;
x(k)=g(x(k-1));
end
% Se converge, x0 oppure x1 contengono il valore
```

```
% dello zero cercato. Altrimenti, si pone flag=0
% che indica la non convergenza
if (k > kmax)
flag=0;
end
niter=k;
x0=x(k);
return
```

Secondo esercizio

```
% --------------
% Esercizio 2
%---------------
close all
clear all
disp('Secondo esercizio')
a=0; b=1;
x=[a,0.5,b]; % nodi
y=[1, 4/5, 1/2] % valori della funzione nei nodi


% a) il polinomio d'interpolazione di grado 2 che interpola
%     f nei punti x, dato un insieme di punti target xx
xx=linspace(a,b,100);
f=1./(1+xx.^2);

% forma di Lagrange del polinomio d'interpolazione

p2= 2*(xx-x(2)).*(xx-x(3))*y(1)-4*xx.*(xx-x(3))*y(2)+2*xx.*(xx-x(2))*y(3);

plot(xx,p2,'r',xx,f,'b');
legend('polinomio', 'funzione'); % plot di confronto

% b) Essendo nodi equispaziati possiamo usare la formula (5.10) ovvero
%     |p2-f| \le max_{x \in [0,1]} |f^(3)(x)|h^3/(3*4)
%
%     con h=1/2 e f^(3)(x)=-48*x^3/(1+x^2)^4+24*x/(1+x^2)^3

h=1/2;
```

```
M3=max(abs(-48*xx.^3./((1+xx.^2).^4)+24*xx./((1+xx.^2).^3)));
errore_interp=M3*h^3/12  % ovvero circa 5.0e-2

% c) si tratta di approssimare l'integrale di f(x) su [0,1]
%    con il metodo di Simpson

int=h/3*(y(1)+4*y(2)+y(3));

% la primitiva di f(x) su [0,1] e' arctg(1)-arctg(0)=pi/4-0=pi/4

errore_quadrat=abs(pi/4-int) % errore circa 2.1e-3

%    d) basta ora usare la formula (6.39). Serve il calcolo della f^(4)(x)
%       per poi calcolarne il massimo
%       f^4(x)=384*x^4/(1+x^2)^5-288*x^2/(1+x^2)^4+24/(1+x^2)^3
fxx2=1+xx.^2;
M4=max(abs(384*xx.^4./fxx2.^5-288*xx.^2./fxx2.^4+24./fxx2.^3));
tol=1.e-4;
N=floor((M4/(tol*2880))^0.25)
% N>3. Prendendo un altro punto in pi si ottiene il risultato voluto.
%

N=2
h=(b-a)/N;
x=linspace(a,b,N+1); %punti equispaziati.
realValue=pi/4;
fSc=funQ(x);
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);
ValSc=ValSc+2*h/3*sum(fSc)
title('Quadratura composita di Simpson e relativi nodi');
disp('Errore assoluto')
erroreS=abs(realValue-ValSc)
% erroreS= 2.181634375375552e-007

% si noti come  erroreS sia molto minore di 10^-4, questo perche' la stima (6.39) e' in
% effetti una sovrastima. Avremmo potuto prendere N tra 1 e 2
% e sarebbe bastato (ma e' una cosa non fattibile)
```

```
% Infatti con per N=2 otteniamo gi un erroe di 6.0e-6.


function y=funQ(x)
y=1./(1+x.^2);
end
```