Esame di Calcolo Numerico

Laurea in Statistica ed Informatica
Laurea Magistrale in Astronomia
Prof. S. De Marchi
`Padova, 30 settembre 2011`

Il candidato dovrà scrivere su **ogni** foglio il cognome, nome, numero di matricola. **Non è ammesso l'uso di appunti!**

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. Data la matrice `A=diag(1:n)` di ordine $n = 5$,

   (a) Studiare il metodo iterativo dipendente dal parametro reale $\theta \in [0, 2/3]$:

   $$x^{(k+1)} = (I - \theta A)x^{(k)} + \theta\, b\,, \quad k \geqslant 0\,. \tag{1}$$

   Si chiede di verificare graficamente per quali valori di $\theta \in [0, 2/3]$ il metodo converge.

   (b) Sia θ^* tale che $\rho(\theta^*) = \min\{\rho(\theta),\ 0 \leqslant \theta \leqslant 2/3\}$. Siano $b \in \mathbb{R}^n$ cosicché `x=ones(n,1)`. Inoltre `x0=zeros(n,1)`. Risolvere quindi il sistema $Ax = b$ con il metodo iterativo (1) a meno di $tol = 1.e - 6$ con test sull'errore relativo.

   (c) Ripetere i calcoli con $n = 3, 10$. Cosa si nota e cosa si può osservare? I risultati cambiano se $\theta \in [0, 1]$?

2. Si consideri la funzione $f(x) = \dfrac{1}{x^2}$, $x \in [1, 3]$

   (a) Determinare il polinomio d'interpolazione in forma di Newton di grado 5 su punti di Chebyshev-Lobatto e calcolarne l'errore in norma infinito.

   (b) Quanti punti sono necessari per calcolare $\displaystyle\int_1^3 f(x)dx$ con la formula trapezoidale a meno di `tol=1.e-3`?

Tempo: **2 ore**.

SOLUZIONI

Alleghiamo i 2 M-files, che eseguiti, danno le soluzioni richieste.

Primo esercizio

```
clear all
close all
% Esercizio 1 esame del 30 settembre 2011

%parte(a)-------
n=input('Dammi dimensione sistema = ');

A=diag(1:n);
t=linspace(0,2/3,100);
for k=1:length(t),
    P=eye(n)-t(k)*A;
    r(k)=max(abs(eig(P)));
end

rt=max(find(r<1)); %indice in cui il raggio spettrale interseca la retta t=1
rrt=t(rt)
plot(t,r,'b',t,ones(1,length(t)),'r:',t(rt),r(rt),'ok')
legend('raggio spettrale','zona ci convergenza','intersezione')


%parte (b)-------
[tt,mr]=min(r);  %minimo di t in corrispondenza al

tm=t(mr)

pause

xs=ones(n,1);
P=eye(n)-tm*A;
b=1/tm*(xs-P*xs)  %calcolo del termine noto b
% o equivalentemente b=A*xs

%-----------------------------------------------------------------------
% calcolo del termine noto b direttamente dal sistema
% originale
% b=A*xs
```

```
% Naturalmente il risultato sara‘ lo stesso
%-------------------------------------------------------------------------

tol=1.e-6; nmax=100;
x0=zeros(n,1);
q=tm*b;
x1=P*x0+q;
k=1;
while (norm(x1-x0,inf)>tol*norm(x1,inf) & k <= nmax)
    x0=x1;
    x1=P*x0+q;
    k=k+1;
end
disp('Iterazioni e soluzione')
k-1
x1

%parte (c)-------
% Basta far eseguire ancora (a) leggendo n=3 e n=10.
```

Secondo esercizio

```
function esercizio2
% Esercizio 2 esame del 30 settembre 2011

%parte(a)---------
n=5;

disp('parte (a)')
a=1; b=3;
xc=cos([0:n]*pi/n);
x=(b-a)/2*xc+(b+a)/2
y=1./(x.^2);

d=DiffDivise(x,y);

targets=linspace(a,b,300);

ft=1./(targets.^2);
pt=Horner(d,x,targets); %Newton's interpolant

err=norm(ft-pt,inf)

pause

%parte (b)----------
disp('parte (b)')

tol=1.e-3;
%f''(x)=6/x^4 che devo massimizzare
ff=6*targets.^(-4);
[M,iM]=max(abs(ff))
nn=ceil(sqrt((b-a)^3*M/(12*tol)))

return


function [b]=DiffDivise(x,y)
%------------------------------------------------
% Algoritmo delle differenze divise
```

```
%-------------------------------------------------

n=length(x); b=y;
for i=2:n,
for j=2:i,
b(i)=(b(i)-b(j-1))/(x(i)-x(j-1));
end;
end;




function p=Horner(d,x,xt)
%-------------------------------------------------
% Questa funzione implementa lo schema di Horner
% per la valutazione del polinomio d'interpolazione
% in forma di Newton su un insieme di punti target
%-------------------------------------------------
% Inputs
% d: vettore delle differenze divise
% x: vettore dei punti d'interpolazione
% xt: vettore dei punti target
%
% Output
% p: il polinomio valutato in tutti i punti
% target
%----------------------------------------------

n=length(d);
for i=1:length(xt) p(i)=d(n);
for k=n-1:-1:1
p(i)=p(i)*(xt(i)-x(k))+d(k);
end
end
```