Esame di Calcolo Numerico

Laurea Magistrale in Astronomia
Prof. S. De Marchi
`Padova, 11 marzo 2011`

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. Data la matrice

$$A = \begin{pmatrix} \alpha & 2 & 3 & 1 \\ 2 & 13 & 1 & 8 \\ 3 & 1 & 16 & 0 \\ 1 & 8 & 0 & 14 \end{pmatrix}, \quad \alpha \in \mathbb{R}$$

   e il termine noto b cosicchè la soluzione $Ax = b$ sia $x = (1, 1, 2, 2)^T$.

   (a) Si determini il parametro ω^* (*omega ottimale*) per risolvere con il metodo SOR il sistema $Ax = b$, per $\alpha = 30$ (si usi tolleranza $\epsilon = 1.e-10$, $\mathbf{x}^{(0)} = \mathtt{ones}(4,1)$ e `nmax=500`). Si dica approssimativamente quante iterazioni sono necessarie per la soluzione di $Ax = b$ con la scelta di ω^*.

   (b) Approssimare l'autovalore di modulo massimo della matrice A con $\alpha = 30$ e $\alpha = 100$. Si dica perchè con $\alpha = 100$ si fanno meno iterazioni.

2. Si consideri la funzione $f(x) = x + \sin(x) + \dfrac{20}{1+x^2} - 2$ ristretta all'intervallo $I = [-2; 2]$

   (a) Determinare il polinomio d'interpolazione di grado 5 in forma di Newton sui nodi equispaziati. Calcolare anche l'errore d'interpolazione in norma infinito.

   (c) Ripetere i calcoli usando i punti di Chebyshev-Lobatto. Che cosa si osserva?

Tempo: **2 ore**.

SOLUZIONI

Primo esercizio

```
% Esercizio 1, esame del 11 marzo 2011
clear all
close all
tol=1.e-10;
a=30;
A=[a  2  3  1
2  13  1  8
3  1  16  0
1  8  0  14];
x=[1 1 2 2]';
b=A*x;
xin=ones(4,1);
nmax=1000;
omega=0.1:0.05:1.9;

%-------------
%parte (a)
%-------------

for i=1:length(omega)
 [xv,it,fl]=gaussril(A,b,xin,nmax,tol,omega(i));
 iter(i)=it;
end

disp('Omega ottimale ')
[min_iter,mm]=min(iter);
omega_star=omega(mm)
%omega_star=1.15

[x_star,it,fl]=gaussril(A,b,xin,nmax,tol,omega_star)
disp(' Le iterazioni corrispondenti ad omega*  sono = ');
min_iter
% Le iterazioni sono min_it=13 corrispondenti a quelle che si sono
% calcolate facendo eseguire gaussril con omega_star e contenute in
% it

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

```
%parte (b)
%--------------------

A(1,1)=a;
 [lam,x,iter]=MetPotenze(A,tol,nmax,ones(4,1))

 A(1,1)=100;
 [lam,x,iter]=MetPotenze(A,tol,nmax,ones(4,1))

 % con a=100 si fanno meno iterazioni, perch il rapporto
 % tra il pi grande e il secondo autovalore  piu' lontano da 1
 % dello stesso rapporto per a=30. Per a=30 servono
 % 28 iterazioni contro le 9 con a=100 .

function [xv,iter,flag]=gaussril(a,f,xin,nmax,toll,omega)
%
% [xv,iter,flag]=gaussril(a,f,xin,nmax,toll,omega)
%
% Metodo di Gauss Seidel rilassato per sistemi lineari.
% Per omega uguale a 1 si ottiene il metodo di Gauss Seidel
%
% Parametri di ingresso:
% a Matrice del sistema
% f Termine noto (vettore riga)
% xin Vettore iniziale (vettore riga)
% nmax Numero massimo di iterazioni
% toll Tolleranza sul test d'arresto (fra iterate)
% omega Parametro di rilassamento
%
% Parametri in uscita
% xv Vettore soluzione
% iter Iterazioni effettuate
% flag se flag=1 converge altrimenti flag=0
%------------------------------------------------------------
flag=1;
[n,m]=size(a);
d=diag(a);
dm1=ones(n,1)./d;
dm1=diag(dm1);
b=eye(size(a))-dm1*a;
```

```
g=dm1*f;
bu=triu(b);
bl=tril(b);
dif=[];
%
% Iterazioni
%
xv=xin;
xn=xv; i=0;
while i<nmax,
for j=1:n;

   xn(j)=(1-omega)*xv(j)+omega*(bu(j,:)*xv+bl(j,:)*xn+g(j));
end;
if abs(xn-xv)<toll,
iter=i;
i=nmax+1;
else
dif=[dif;norm(xn-xv)];
xv=xn;
i=i+1;
end,
end
if i==nmax,
disp('** Non converge nel numero di iterazioni fissato')
flag=0;
end

function [lam,x,iter]=MetPotenze(A,tol,kmax,x0)
%-----------------------------------------------
% Inputs
% A: matrice,
% tol: tolleranza;
% kmax: numero massimo d'iterazioni
% x0: vettore iniziale
%--
% Outputs
% lam : autovalore di modulo massimo
% x: autvettore corrispondente
% iter: iterazioni effettuate
```

```
%-----------------------------------------------
x0=x0/norm(x0); lam=x0'*(A*x0); err=tol*abs(lam)+1; iter=0;
while (err > tol*abs(lam) & abs(lam)~=0 & iter<=kmax)
x=A*x0;
x=x/norm(x);
lamnew=x'*(A*x);
err=abs(lamnew-lam);
lam=lamnew;
x0=x;
iter=iter+1;
end
return
```

Secondo esercizio

```
% Esercizio 2, esame del 11 marzo 2011
clear all
close all
n=6;
xx=linspace(-2,2,100);
yy=f(xx);

% parte (a)
x=linspace(-2,2,n);
y=f(x);
d=DiffDivise(x,y);

% polinomio d'interpolazione in forma di newton
for i=1:length(xx)
p(i)=d(n);
  for j=n-1:-1:1,
      p(i)=(xx(i)-x(j))*p(i)+d(j);
  end
end

disp('Errore in norma infinito')
err=norm(yy-p,inf)

% parte (b)
c=-cos([0:n-1]*pi/(n-1));
x1=2*c; %punti di Chebyshev-Lobatto in [-2,2]
y1=f(x1);
d1=DiffDivise(x1,y1);

% polinomio d'interpolazione in forma di newton
for i=1:length(xx)
p1(i)=d1(n);
  for j=n-1:-1:1,
      p1(i)=(xx(i)-x1(j))*p1(i)+d1(j);
  end
end

plot(xx,yy,'r',xx,p,'g',xx,p1,'b',x,y,'ms',x1,y1,'ok');
```

```
legend('funzione','poli. equisp','pol. Cheby.','punti equi.','punti Cheb.Lob')
disp('Errore in norma infinito')
err1=norm(yy-p1,inf)

% Si osserva che err1>err. Questo deriva dalla poca accuratezza con cui
% s'interpola nella parte centrale. Naturalmente aumentando n si assistera'
% ad una inversione di tendenza. Infatti per n=10, err1 e' circa meta' di
% err.


function [b]=DiffDivise(x,y)
%---------------------------------------------------
% Algoritmo delle differenze divise
%---------------------------------------------------
% Inputs
% x: vettore dei punti di interpolazione,
% y: vettore dei valori della funzione.
% Output
% b: vettore delle differenze divise b=[b_1,....b_n]
%----------------------------------------------------
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;
```

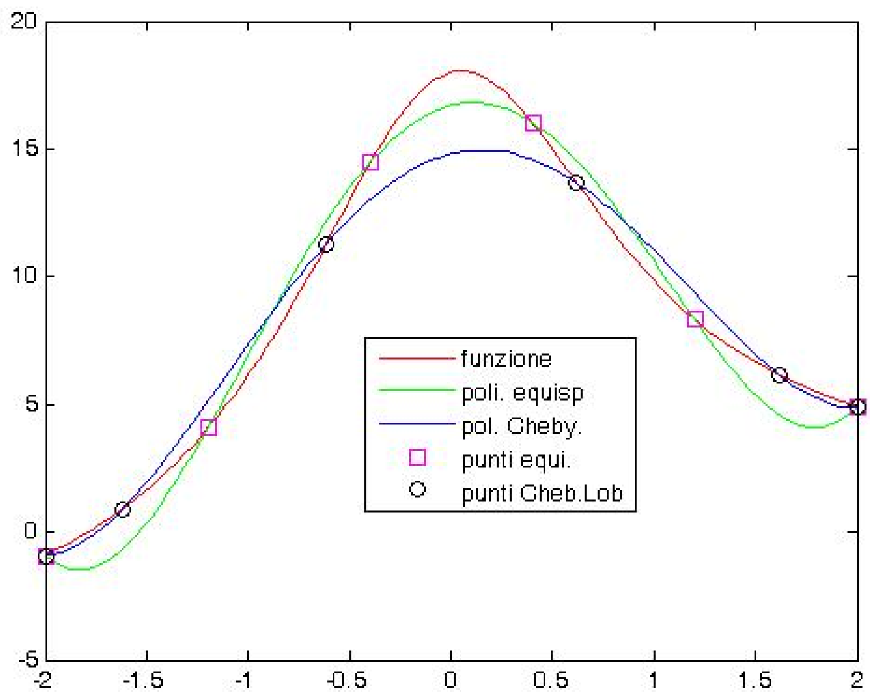

Figure 1: Confronto dei polinomi d'interpolazione in forma di Newton su nodi equispaziati e di Chebyshev-Lobatto