

PROVA PRATICA di CALCOLO NUMERICO

Prof. S. De Marchi

Verona, 23 marzo 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.

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

1. Si consideri la funzione $g(x) = x(2 - kx)$, $k > 0$.
 - (a) Quali i sono i punti fissi di $g(x)$?
 - (b) Si consideri ora il metodo di iterazione funzionale $x_{i+1} = g(x_i)$, $i \geq 0$. Si determini l'intervallo I_α di convergenza alla radice **positiva** α .
 - (c) Determinare l'ordine di convergenza del metodo iterativo.
 - (d) Si verifichino i risultati teorici nel caso $k = 3$, usando $tol = 1.e - 9$ e come test d'arresto l'errore assoluto.

2. Si consideri la matrice

$$A = \begin{pmatrix} \frac{1}{3} & \frac{2}{3} & 2 & 3 \\ 1 & 0 & -1 & 2 \\ 0 & 0 & -\frac{5}{3} & -\frac{2}{3} \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (1)$$

- (a) Perché il metodo delle potenze non funziona per il calcolo dell'autovalore di modulo massimo?
 - (b) Calcolare l'autovalore di modulo massimo e il corrispondente autovettore con il metodo delle potenze con shift.
3. Si consideri la funzione $f(x) = x + e^x + \frac{20}{1+x^2} - 5$ ristretta all'intervallo $[-2, 2]$.
 - (a) Determinare il polinomio d'interpolazione di grado 5 in forma di Newton sui nodi equispaziati $x_k = -2 + kh$, $k = 0, \dots, 5$.
 - (b) Calcolare l'errore d'interpolazione in un generico punto $x \in (-2, 2)$.
 - (c) Ripetere i calcoli usando i punti di Chebyshev.

Tempo: **3 ore**.

SOLUZIONI

1. (a) I punti fissi di $g(x)$ si ottengono risolvendo $g(x) = x$ e sono $x_1 = 0$ e poiché $k > 0$, $x_2 = 1/k$.
- (b) Detta $\alpha = x_2 = 1/k$ la radice positiva, l'intervallo di convergenza del metodo $x_{i+1} = g(x_i)$ si ottiene verificando la disuguaglianza $|g'(1/k)| < 1$. Risolvendola si ottiene $\frac{1}{2k} < x < \frac{3}{2k}$.
- (c) Circa l'ordine di convergenza del metodo iterativo si deve verificare chi è il seguente limite

$$\lim_{i \rightarrow \infty} \frac{|x_i(2 - kx_i) - \alpha|}{|x_i - \alpha|^p}$$

risulta finito se e solo se $p = 2$. Il metodo ha quindi ordine di convergenza 2. Oppure basta osservare che $g'(1/k) = 0$ che ci garantisce che il metodo, come nel caso di Newton, ha ordine di convergenza almeno quadratica pur di partire da un valore iniziale interno ad I_α .

- (d) Sia ora $k = 1$ che implica $\alpha = 1/3$ e quindi l'intervallo $I_\alpha = (1/2, 1/6)$. Il codice usato per la soluzione numerica è

```
clear

%-----
% Esercizio 1 del compito del 23 marzo 2006
%-----

x=input('Valore iniziale = '); k=3;

x1=2*x-k*x^2;
e(1)=abs(x-x1);
i=1;
while(abs(x-x1)>1.e-9 & i<100)
    x=x1;
    x1=2*x-k*x^2;
    i=i+1;
    e(i)=abs(x-x1);
end semilogy(1:i,e,'--');
legend('errore');
i
x

% Esecuzione
```

```
>> met_it
Valore iniziale = .4

i =

    5

x =

    0.3333

>>
```

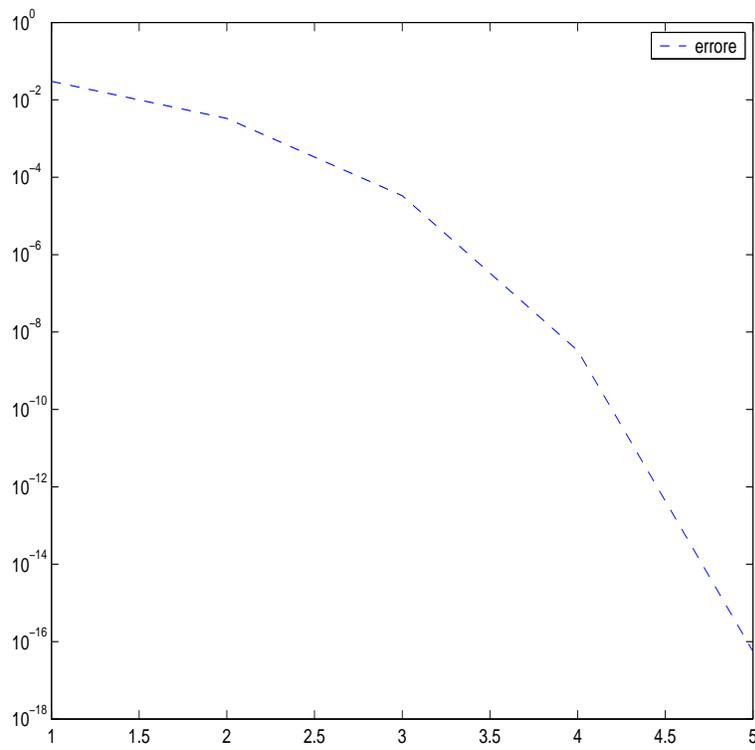


Figure 1: Grafico dell'errore relativo al metodo iterativo dell'Esercizio 1 applicato alla radice positiva $\alpha = 1/3$ a partire da $x_0 = .4$

2. • Il metodo delle potenze non funziona perchè ci sono 2 autovalori di modulo massimo, ovvero $\lambda_1 = 1$ e $\lambda_2 = -1$.
- Usando quindi il metodo delle potenze con shift, con il codice qui sotto riportato,

```

clear
%-----
% Esercizio 2: compito del 23/03/06
%-----
%
% clear
%-----
% Questo il metodo delle potenze inverse con shift per il
% calcolo dell'autovalore di A pi vicino ad un numero mu dato
%-----
tol=1.e-10;
eta=input('Dammi eta = ');

mat=[1/3 2/3 2 3; 1 0 -1 2; 0 0 -5/3 -2/3; 0 0 1 0];

%-----
n=max(size(mat));

a=inv(mat-eta*eye(n));
kmax=2000;
x0=ones(n,1);
y0=x0/norm(x0);
x1=a*y0;
lambda0=y0'*x1;
y0=x1/norm(x1);
x1=a*y0;
lambda1=y0'*x1;
k=0;
while( k <= kmax &abs(lambda1-lambda0)>tol*abs(lambda1) &
abs(lambda1)>0 )
    lambda0=lambda1;
    y0=x1/norm(x1);
    x1=a*y0;
    lambda1=y0'*x1;
    k=k+1;
end
disp('iterazioni fatte = '); k

```

```

disp('autovettore = '); y0

disp ('autovalore pi vicino a eta = '); 1/lambda1+eta
disp('Calcolo del residuo A x-lambda x = '); a*y0-lambda1*y0

% Esecuzione

Dammi eta = 1.2 iterazioni fatte =

k =

    11

autovettore =

y0 =

    0.7071
    0.7071
   -0.0000
    0.0000

autovalore pi vicino a eta =

ans =

    1.0000

Calcolo del residuo A x-lambda x =

ans =

    1.0e-010 *

   -0.1140
    0.1140
   -0.0603
    0.0970

```

Come si nota, più vicino è il punto di partenza da 1 o -1 e più veloce sarà il metodo

delle potenze inverse con shift.

3. Questo esercizio si poteva risolvere completamente con il codice Matlab per l'interpolazione in forma di Newton (che usa le differenze divise) sia su nodi equispaziati che di Chebyshev.

```

%-----
% Interpolazione di funzioni: forma di Newton
%-----
clear a=input('estremo sx = '); b=input('estremo dx = ');
nmax=input('grado massimo polinomio di interpolazione = ');
tipoNodi=input('Nodi equispaziati (E) o di Chebyshev (C) ?','s');

xx=a:0.01:b; % target points
yy=fun(xx); % i valori della funzione sui targets

xerr=input(' Punto in cui valutare l''errore = '); err=[]; for
n=2:nmax,
    switch tipoNodi
    case {'E'}
        x=a:(b-a)/n:b;
        titlestring=['Polinomio d''interpolazione di grado ', num2str(n),' , nodi equispaziati'];
    case {'C'}
        x=-(b-a)/2*cos((2*[1:n+1]-1)*pi/(2*(n+1)))+(b+a)/2;
        titlestring=['Polinomio d''interpolazione di grado ', num2str(n),' , nodi di Chebyshev'];
    otherwise
        error('tipo nodi errato')
    end

    y=fun(x);
    d=diffDivise(x,y);
    p=horner(x,d,xx);
    plot(xx,yy,'-r',xx,p,'--b',x,zeros(1,length(x)),'ok')
    legend('funzione','Pol. interpolante','nodi interp.');
```

```

title(titlestring);

% -----
% Calcolo dell'errore
% inserisco xerr e il valore f(xerr) nel posto corrispondente
% nota: non serve ordinare il vettore perch le differenze divise
%       sono invarianti per permutazioni dei punti.
%-----

x=[x, xerr];
y=[y, fun(xerr)];

```

```

    fn1=diffDivise(x,y);
    err(n-1)=1;
    for i=1:n+1,
        err(n-1)=err(n-1)*(xerr-x(i));
    end
    err(n-1)=err(n-1)*fn1(end);
end

stringErrore=['Errori d'interpolazione in ',num2str(xerr)];
disp(stringErrore); err

```

```

%-----FUNZIONE da interpolare

```

```

function [y,yp]=fun(x) global p

% Funzione del 23/3/06
y=x+exp(x)+20./(1+x.^2)-5;
yp=1+exp(x)-40*x./((1+x.^2).^2);

```

```

return

```

```

%----- FUNZIONE delle Differenze Divise

```

```

function d=diffDivise(x,y)
%-----
% Differenze divise di y nei punti x
%-----
n=length(y); d=y; for i=1:n,
    if i~=1,
        for j=2:i,
            d(i)=(d(i)-d(j-1))/(x(i)-x(j-1));
        end
    end
end return

```

```

%----- Horner

```

```

function p=horner(x,c,xx)
%-----
% Valuta sui punti xx il polinomio d'interpolazione

```

```

% nella forma di Newton costruito sui nodi x e di
% coefficienti c
% Inputs:
% x: vettore ascisse punti d'interpolazione
% c: coefficienti del polinomio nella forma di Newton
% xx: vettore in cui si valuta
%
% Output
% p: vettore dei valori del polinomio d'interpolazione
%-----
n=length(c); for j=1:length(xx) p(j)=c(n);
  for i=n-1:-1:1,
    p(j)=c(i)+(xx(j)-x(i))*p(j);
  end
end

return

% Esecuzione su nodi equispaziati

>> InterpolazioneNewton
estremo sx = -2 estremo dx = 2 grado massimo polinomio di
interpolazione = 5 Nodi equispaziati (E) o di Chebyshev (C) ?E
Punto in cui valutare l'errore = .2
Errori d'interpolazione in 0.2

err =

    -0.7781    4.3450   -0.2851    0.9030

>>

```

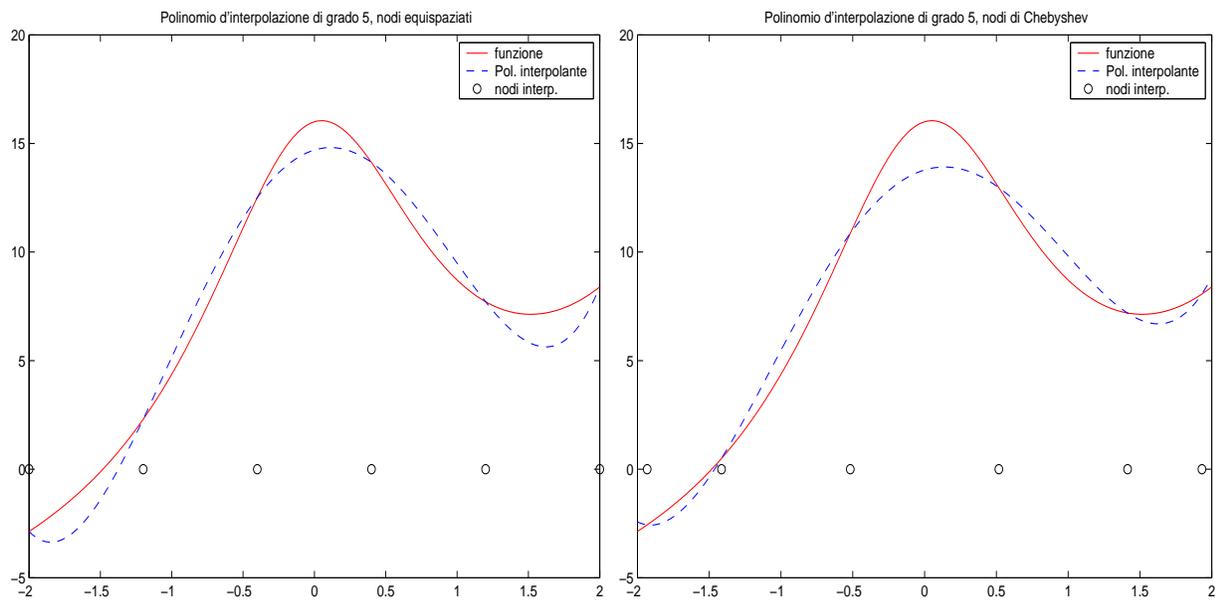


Figure 2: Funzione ed interpolante di grado 5: SX su nodi equispaziati; DX su nodi di Chebyshev