

# PROVA PRATICA di CALCOLO NUMERICO

Prof. S. De Marchi

Verona, 19 dicembre 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!**. Per la consegna: inviare una email con tutto (diary, testo degli m-files e figure) a [stefano.demarchi@univr.it](mailto:stefano.demarchi@univr.it).

**NOTA BENE: allegare immagini in formato .jpg o .eps**

1. Si consideri il metodo d'iterazione funzionale

$$x_{i+1} = x_i + e^{1-x_i} - 1.$$

Provare, dapprima teoricamente e quindi numericamente usando  $tol = 1.e - 9$ , che questo procedimento converge all'unico punto fisso della funzione d'iterazione. Calcolarne anche l'ordine di convergenza.

2. Si considerino le matrici

$$A_1 = \begin{pmatrix} -7 & -9 & 9 \\ 11 & 13 & -9 \\ -16 & -16 & 20 \end{pmatrix} \quad A_2 = \begin{pmatrix} -4 & -5 & 4 \\ 14 & 15 & -5 \\ -1 & -1 & 11 \end{pmatrix}. \quad (1)$$

entrambe con autovalori reali e distinti.

- (a) Calcolare tutti gli autovalori di  $A_1$  e  $A_2$ , mediante il metodo delle inverse con shift (usare  $tol = 1.e - 6$ ).
  - (b) Si dica perchè il metodo delle potenze per il calcolo dell'autovalore di modulo massimo ci impiega di più nel caso della matrice  $A_2$ ?
3. Si consideri la funzione  $f(x) = \frac{20}{1+x^2} - 5e^x$  ristretta all'intervallo  $[0, 1]$ . Determinare l'unico polinomio (d'interpolazione) di secondo grado,  $p_2(x) = a_0 + a_1x + a_2x^2$  tale che

$$p_2(0) = f(0), \quad p_2(1) = f(1), \quad \int_0^1 p_2(x)dx = \int_0^1 f(x)dx.$$

Per il calcolo dell'integrale della funzione usare il metodo di Simpson composito con errore minore di  $1.e - 6$ .

Fare il grafico della funzione e del polinomio.

Tempo: **3 ore**.

## SOLUZIONI

1. Se indichiamo con  $g(x) = x - 1 + e^{1-x}$  la funzione d'iterazione. Anzitutto risolvendo  $g(x) = x$  si ottiene che l' **unico** punto fisso è  $x = 1$ .

Circa la regione di convergenza, basta risolvere la disuguaglianza

$$|g'(x)| = |1 - e^{1-x}| < 1$$

trovando che essa è verificata per  $x > 1 - \log 2 \approx 0.3$

Per l'ordine di convergenza, basta verificare che  $g'(1) = 0$  mentre  $g''(x) = e^{1-x}$  in  $x = 1$  risulta uguale a 1, quindi diversa da zero. Pertanto il metodo è del secondo ordine.

Il codice corrispondente è

```
clear

%-----
% Esercizio 1 del compito del 19 dicembre 2006
%-----

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

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

>> esame19dicembre2006esI
Valore iniziale = 2

i =

    6

x =
```

1.0000

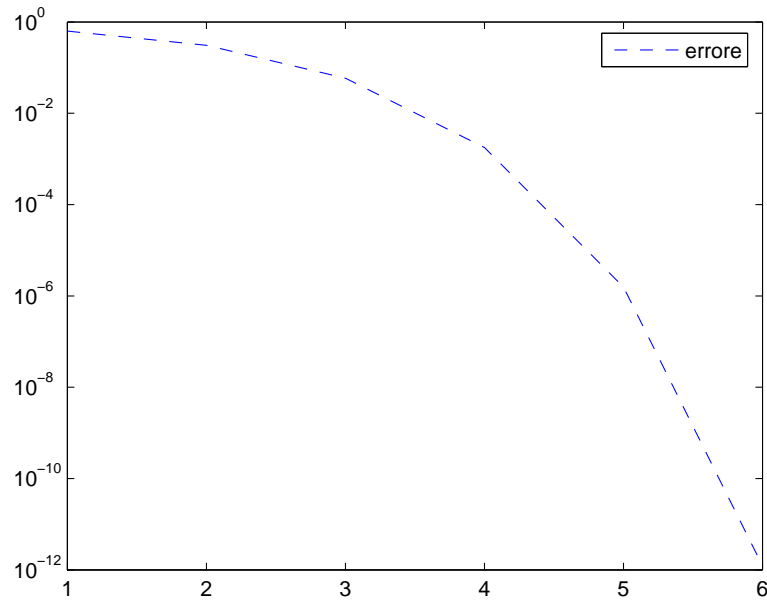


Figure 1: Esercizio1: dall'andamento dell'errore si individua la convergenza quadratica

2. Applicando tre volte su ciascuna delle matrici il metodo delle potenze inverse con shift (vedasi qui sotto i files matlab) si ottengo tutti gli autovalori. Per  $A_1$ , gli autovalori che si determinano sono: 20, 4, 2. Per  $A_2$  sono, 11.6180, 9.3820, 1. Quindi  $\lambda_M^{(1)} = 20$  e  $\lambda_M^{(2)} = 11.618$ .

Per rispondere al secondo quesito basta calcolare i rapporti tra l'autovalore successivo  $\lambda_2^{(i)}$  e il più grande in modulo. Dalla teoria sappiamo che più grande è il rapporto e più lenta sarà la convergenza. Per  $A_1$ ,  $\lambda_2(1)/\lambda_1(1) = 4/20 = 0.2$ . Per  $A_2$ ,  $\lambda_2(2)/\lambda_1(2) = 9.382/11.618 \approx 0.81$ . Per questo il metodo delle potenze converge più lentamente all'autovalore più grande in modulo di  $A_2$

```
clear;
```

```
%-----  
% Esercizio 2 del compito del 19 dicembre 2006  
%-----
```

```
a1=[-7 -9 9; 11 13 -9; -16 -16 20];
```

```
a2=[-4 -5 4; 14 15 -5; -1 -1 11];
```

```
MPshift(a1,1,1.e-6,100)
MPshift(a1,10,1.e-6,100)
MPshift(a1,20,1.e-6,100)
```

```
MPshift(a2,1,1.e-6,100)
MPshift(a2,10,1.e-6,100)
MPshift(a2,20,1.e-6,100)
```

```
function [lamb, ylamb]=MPshift(mat,eta,tol, kmax)
```

```
%-----
% Metodo delle Potenze con shif
% Inputs:
%     mat= matrice
%     eta= stima dell'autovalore
%     tol= tolleranza
%     kmax = max iterazioni
%
% Outputs:
%     lamb=autovalore vicino a eta
%     ylamb= autovettore corrispondente
%
%-----
```

```
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
lamb=1/lambda1+eta;
ylamb = y0;
```

```
>> esame19dicembre2006esII
```

```
ans =
```

```
2.0000
```

```
ans =
```

```
4.0000
```

```
Warning: Matrix is close to singular or badly scaled.
```

```
Results may be inaccurate. RCOND = 1.644775e-017.
```

```
> In MPshift at 17
```

```
In esame19dicembre2006esII at 12
```

```
ans =
```

```
20
```

```
Warning: Matrix is close to singular or badly scaled.
```

```
Results may be inaccurate. RCOND = 2.220446e-017.
```

```
> In MPshift at 17
```

```
In esame19dicembre2006esII at 14
```

```
ans =
```

```
1.0000
```

```
ans =
```

```
9.3820
```

```
ans =
```

```
11.6180
```

3. Si tratta di risolvere il sistema lineare con matrice non singolare

$$A = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & \frac{1}{2} & \frac{1}{3} \end{pmatrix}$$

e termine noto

$$b = \begin{pmatrix} 15 \\ 5 * (2 - \exp(1)) \\ \int_0^1 f(t) dt \end{pmatrix}$$

Il calcolo dell'integrale con il metodo di Simpson composito, da come risultato 7.1166 (vedere sotto). Ma osserviamo che l'integrale definito è calcolabile facilmente analiticamente ottenendo

$$20 \arctan(1) - 5(e - 1) = 5\pi - 5e + 5 \approx 7.1166.$$

Risolviendo il sistema si trovano i valori dei coefficienti del polinomio. Il sistema lo possiamo risolvere tranquillamente con Gauss usando la funzione matlab } e otteniamo

$$a_0 = 15, a_1 = -10.118; a_2 = -8.474.$$

In Figura 2 si vedono i grafici della funzione e del polinomio.

```
clear
%-----
% Esercizio 3 del compito 19 dicembre 2006
%-----
% Simpson composito
%-----
format long
n=input('Numero di punti della suddivisione ');
a=0;
b=1;
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);
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)
disp('Errore assoluto')
erroreT=abs(realValue-ValSc)

```

%-----

```

function y=funQ(x)
y=20./(1+x.^2)-5*exp(x);
return

```

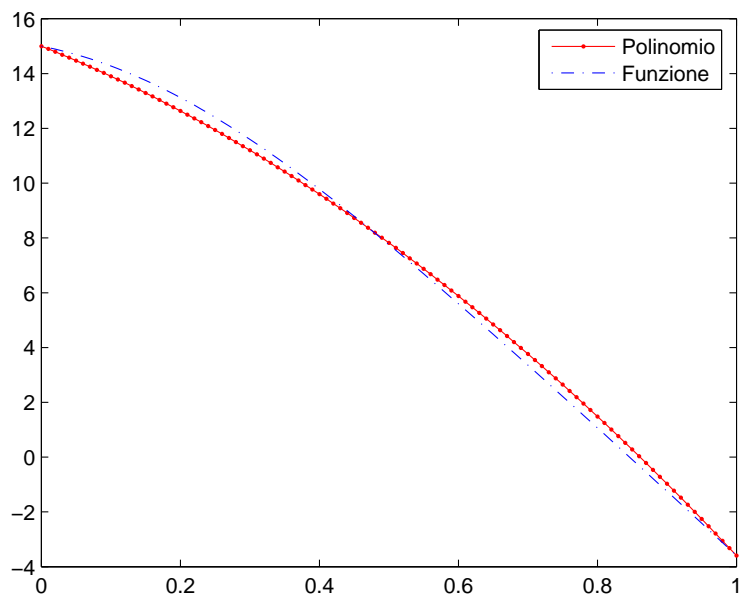


Figure 2: Funzione e polinomio d'interpolazione