PROVA PRATICA DI CALCOLO NUMERICO
per MATEMATICA APPLICATA E INFORMATICA MULTIMEDIALE
Prof. Stefano De Marchi
`Verona, 22 giugno 2007`

Il candidato dovrà scrivere su **ogni** foglio il cognome, nome, numero di matricola. **Consegnare fogli leggibili!**. Inviare una email a `stefano.demarchi@univr.it` contenente tutti i files e le figure prodotte. **NOTA: allegate immagini in formato .jpg o .eps**

1. Dato il polinomio
$$p_n(x) = \sum_{i=0}^{n} a_i x^i, \quad a_0 a_n \neq 0 \ ,$$
per determinare la radice ξ_1 di modulo massimo si può usare il *metodo di Bernoulli*. Tale metodo consiste nell'applicare il *metodo delle potenze* alla matrice (di Frobenius)
$$F = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ 0 & \dots & & 0 & 1 \\ -\frac{a_0}{a_n} & -\frac{a_1}{a_n} & \dots & -\frac{a_{n-2}}{a_n} & -\frac{a_{n-1}}{a_n} \end{pmatrix}$$
   (a) Calcolare ξ_1 per il polinomio
   $$p_6(x) = 13x^6 - 364x^5 + 2912x^4 - 9984x^3 + 16640x^2 - 13312x + 4096 \ ,$$
   usando tolleranza $tol = 1.e-6$.

   (b) Calcolare quindi (per deflazione) ξ_2 la seconda radice più grande in modulo. Per calcolare ξ_2, si suggerisce dapprima di costruire la matrice P di *Householder* tale
   $$PFP^T = \begin{pmatrix} \xi_1 & \mathbf{a}^T \\ \mathbf{0} & F_1 \end{pmatrix}$$
   cosicchè $Px_1 = e_1$ con x_1 autovettore associato a ξ_1 (calcolato al passo 1) e $e_1 = (1, 0, ..., 0)^T$. La matrice P si può costruire mediante il seguente codice Matlab:

```
% Costruisco la matrice di Householder P t.c. P*x1=(1,0,...,0)

x12=norm(x1,2); beta=1/(x12*(x12+abs(x1(1)) ));
v(1)=sign(x1(1))*(abs(x1(1))+x12); v(2:n)=x1(2:n);
P=eye(n)-beta*v'*v;   % n = dimensione matrice
```

Pertanto, per calcolare ξ_2 si applicherà il metodo delle potenze alla matrice F_1.

Confrontare i risultati con la funzione `roots(c)`, con `c` vettore dei coefficienti del polinomio $p_6(x)$, aiutandosi anche con un grafico.

2. Un corpo in caduta libera all'equatore, subisce una deviazine dalla verticale dovuta all'accelerazione di Coriolis. Supponendo che al tempo $t = 0$ il corpo sia fermo (cioe' $x(0) = 0, v(0) = 0$ e $a(0) = 0$) e che la sua accelerazione di Coriolis sia nota solo in alcuni istanti di tempo, si determini lo spostamento dalla verticale dovuto a tale accelerazione dopo $t = 100$ *sec.*.

   In tabella elenchiamo ad istanti distinti del tempo t, i corrispondenti valori dell'accelerazione a:

t	10	15	30	40	50	70	100
a	.0144	.0216	.0432	.0576	.072	.1008	.1439

   Suggeriamo di calcolare mediante integrazione dell'accelerazione, la velocità v negli istanti di tempo indicati in tabella, usando la *formula di quadratura dei trapezi composita*. Integrando nuovamente con i trapezi compositi negli stessi istanti di tempo, si calcolerà la deviazione $x(t)$.

   Nota bene: gli intervalli di tempo non sono equispaziati.

   Facendo poi uso della funzione Matlab `interp1` (interpolante lineare) determinare la deviazione dovuta all'accelerazione di Coriolis a partire dai vettori `t` (tempo) e `v` (velocità). Tale deviazione cambia se invece dell'interpolazione lineare si usa quella polinomiale cubica o splines (sempre servendosi di `interp1`) ?

   Quale sarebbe la distanza percorsa dal corpo dopo $t = 100$ *sec* (in assenza di attrito)? Qui ci deve ricordare la 2^a legge della dinamica!

◊◊

Tempo: **3 ore**.

SOLUZIONI

Es. 1.

```
%-------------------------------------
% Compito di CN del 22/6/07 : Es 1
%-------------------------------------
clear; nmax=1000; tol=1.e-6;

c=[4096 -13312 16640 -9984 2912 -364 13];

c1=c(1:length(c)-1)/c(length(c)); %normalizzo
n=length(c1);

% Costruisco la matrice di Frobenius

F(n,:)=-c1; F=F+diag(ones(1,n-1),1);

% Applico il metodo delle potenze per il calcolo
% dell'autovalore di modulo max della matrice F
t0=ones(n,1); m0=1000; t1=F*t0; m1=max(abs(t1)); t1=t1/m1;

k=1;

while( abs(m1-m0) >tol & k <= nmax  )
   t0=t1;
   m0=m1;
   t1=F*t0;
   m1=max(abs(t1));
   t1=t1/m1;
   k=k+1;
end; rmax1=m1

t1

% Calcolo del secondo autovalore-radice per deflazione
% Costruisco la matrice di Householder P t.c. P*t1=(1,0,...,0)
t12=norm(t1,2); beta=1/(t12*(t12+abs(t1(1)) ));
v(1)=sign(t1(1))*(abs(t1(1))+t12); v(2:n)=t1(2:n);
P=eye(n)-beta*v'*v;
```

```
F1=P*F*P';

%Per determinare il successivo autovalore devo riapplicare il metodo delle
% potenze alla sottomatrice di F1 di dimensione n-1 x n-1

F2=F1(2:n,2:n);

t0=ones(n-1,1);
m0=1000;
t1=F2*t0;
m1=max(abs(t1));
t1=t1/m1; k=1;

while( abs(m1-m0) >tol & k <= nmax  )
   t0=t1;
   m0=m1;
   t1=F2*t0;
   m1=max(abs(t1));
   t1=t1/m1;
   k=k+1;
end; t2=t1;

rmax2=m1

t2


disp('Controllo usando la funzione roots')

roots(c(end:-1:1))
%----------------- RISULTATI

rmax1 =

   17.4605


t1 =

   -0.0000
```

```
   -0.0000
   -0.0002
   -0.0033
   -0.0573
   -1.0000


rmax2 =

    4.6303

t2 =

   -0.0137
   -0.0625
   -0.2740
   -1.0000
    0.0612

Controllo

ans =

   17.4605
    4.6303
    2.2741
    1.4764
    1.1438
    1.0147
```

Es. 2.

```
%----------------------------------------
% Esame 22 giugno 2007, esercizio 2
%----------------------------------------
clear;
t=[0 10 15 30  40  50  70  100];

a=[0 .0144 .0216 .0432 .0576 .072 .1008 .1439];
%.................................
% Calcolo dapprima la velocita‘ usando
% la formula dei trapezi composita
%................................
n=length(t);

v(1)=0;
for i=2:n;
  v(i)=v(i-1)+(t(i)-t(i-1))/2*(a(i)+a(i-1));
end;

%....................................
% Calcolo ora la deviazione, x(t), usando
% la formula dei trapezi composita
%...................................
x(1)=0;
for i=2:n;
  x(i)=x(i-1)+(t(i)-t(i-1))/2*(v(i)+v(i-1));
end;

disp(' Dopo 100 sec. la deviazione e‘ '); x(n)


sss=input('Interpolante lineare(l), cubica(c), spline(s)) ? ','s');

xi=min(t):max(t);
%xi=t;
if(sss=='l')
  disp('Interpolante lineare');
  yi=interp1(t,v,xi);
elseif(sss=='c')
  disp('Interpolante cubica');
  yi=interp1(t,v,xi,'cubic');
```

```
elseif(sss=='s')
  disp('Interpolante spline cubica');
  yi=interp1(t,v,xi,'spline');
else
  disp('Error'); break;
end;

%
% Calcolo ora la deviazione, su yi(t), usando
% la formula dei trapezi composita
%
m=length(yi);
xx(1)=0;
for i=2:m;
  xx(i)=xx(i-1)+(xi(i)-xi(i-1))/2*(yi(i)+yi(i-1));
end;

disp(' Dopo 100 sec. la deviazione e` '); xx(m)

% Per il calcolo dello spazio percorso in
% assenza di attrito, basta ricordare, ad esempio
% che il corpo e` soggetto alla sola forza peso: m *g
% Per la seconda legge della dinamica:
%  mg=ma= m d^2x(t)/dt^2 ===> (integrando 2 volte) x(t)=1/2 g t^2
%
disp(' Lo spazio percorso dal corpo dopo 100 sec. e` :');
h=0.5*9.81*100^2

% NOTA: analogo risultato si ottiene con il principio di conservazione
%       dell'energia (ma bisogna risolvere un equazione diff. non lineare
%       (anche se semplice).




%------------------------- Esecuzione

comp32
 Dopo 100 sec. la deviazione e`
```

```
ans =

  244.9575


Quale interpolante lineare(l), cubica(c), spline(s)) ? l
Interpolante lineare
 Dopo 100 sec. la deviazione e‘

ans =

  244.9575

 Lo spazio percorso dal corpo dopo 100 sec. e‘ :

h =

       49050

>> comp32
 Dopo 100 sec. la deviazione e‘

ans =

  244.9575


Quale interpolante lineare(l), cubica(c), spline(s)) ? c
Interpolante cubica
 Dopo 100 sec. la deviazione e‘

ans =

  240.0538

 Lo spazio percorso dal corpo dopo 100 sec. e‘ :

h =

       49050
```

```
>> comp32
 Dopo 100 sec. la deviazione e‘

ans =

  244.9575


Quale interpolante lineare(l), cubica(c), spline(s)) ? s
Interpolante spline cubica
 Dopo 100 sec. la deviazione e‘

ans =

  239.9972

 Lo spazio percorso dal corpo dopo 100 sec. e‘ :

h =

       49050
```