

Cognome: _____ Nome: _____ Matricola: _____

Laboratorio di Calcolo Numerico - Corso di Laurea in Matematica Appello d'esame del 19/06/2012

ESERCIZIO 1 [10 punti]

Si considerino le seguenti coppie di valori

$$x = [1200.5, 1201.5, 1202.5, 1203, 1204, 1205]; \quad y = [3, 1.5, 1.5, 1, 1, 0];$$

1. Si generi il polinomio globale interpolante di Lagrange usando il comando `polyfit`. Quale approccio matematico viene usato seguendo questo procedimento?
2. Si generi il polinomio globale interpolante di Lagrange usando la function `lagr` (può essere scaricata dalla pagina web dei laboratori). Quale approccio matematico viene usato seguendo questo procedimento?
3. Aiutandosi con una rappresentazione grafica dei dati e dei polinomi trovati ai due punti precedenti si dia una spiegazione dei risultati ottenuti.

- approccio in `polyfit`
- approccio in `lagr`
- spiegazione risultati:

ESERCIZIO 2 [10 punti]

Si consideri il sistema lineare

$$Ax = b,$$

con

$$A = \begin{bmatrix} 1/2 & 0 & 0 & -1 \\ 0 & 1 & -1/3 & 0 \\ 0 & -1/4 & 1 & 0 \\ -1 & 0 & 0 & 4/5 \end{bmatrix},$$

e termine noto b scelto in modo tale che la soluzione esatta sia data dal vettore $x = (1, 1, 1, 1)^T$.

1. Per la risoluzione di tale sistema lineare, si vuole usare dapprima il seguente metodo iterativo, indicato come metodo (*)

$$\begin{cases} x^{(0)} & \text{dato,} \\ x^{(k+1)} = Bx^{(k)} + b, & k = 0, \dots \end{cases}$$

con $B = (I - A)$. Si verifichi sperimentalmente con Matlab (usando la condizione necessaria e sufficiente) se il metodo proposto converge alla soluzione esatta

2. Si verifichi sperimentalmente con Matlab (usando la condizione necessaria e sufficiente) che i metodi di Jacobi e Gauss-Seidel *non* convergono alla soluzione esatta
3. Si proponga una semplice permutazione delle righe della matrice A tale per cui i i metodi di Jacobi e Gauss-Seidel convergono alla soluzione esatta. Quante iterazioni sono necessarie per ciascuno dei due metodi utilizzando i dati `x0=zeros(4,1)`, `toll=1e-4`?

- il metodo (*) converge alla soluzione esatta? Perché?
- motivo per cui i metodi di Jacobi e Gauss Seidel non convergono
- matrice A permutata:

nr. iterazioni Jacobi= nr. iterazioni Gauss-Seidel=

ESERCIZIO 3 [10 punti]

Si consideri il calcolo dell'integrale

$$I = \int_{-1}^3 |x(x-1)(x-2)| dx$$

1. Si calcoli tramite il toolbox simbolico il valore esatto I
2. Quanti intervalli n_{PM} sono necessari per approssimare l'integrale con un errore (differenza in modulo tra valore esatto e valore approssimato) inferiore a 10^{-5} usando la formula del punto medio?
3. Sapendo che la formula di Simpson integra esattamente polinomi di grado 3, proporre una strategia (ed utilizzarla) per calcolare numericamente ed in maniera esatta l'integrale I con la formula di Simpson composta su n_S intervalli equispaziati opportunamente scelti (Suggerimento: si tracci dapprima un grafico della funzione integranda e si consideri la regolarità di tale funzione!)

- valore esatto I
- numero intervalli $n_{PM} =$
- numero intervalli $n_S =$

Appello 19/06/2012 - Soluzione

Esercizio 1.

1. Costruiamo l'interpolante con il comando `polyfit`

```
>> x=[1200.5 1201.5 1202.5 1203 1204 1205];
>> y=[3 1.5 1.5 1 1 0];
>> plot(x,y,'ro'), hold on
>> p=polyfit(x,y,5);
```

```
Warning: Polynomial is badly conditioned. Add points with distinct X
values, reduce the degree of the polynomial, or try centering
and scaling as described in HELP POLYFIT.
```

```
> In polyfit at 80
>> xx=linspace(1200.5,1205);
>> plot(xx,polyval(p,xx),'g')
```

2. Usiamo ora la function `lagr`

```
>> for i=1:numel(xx), L(i)=lagr(x,y,xx(i)); end
>> plot(xx,L,'m')
```

3. Osserviamo che il polinomio interpolante ottenuto con il comando `polyfit` -che risolve il sistema lineare associato alla matrice di Vandermonde costruita dai dati da interpolare- dà origine ad un polinomio il cui comportamento è nettamente peggiore rispetto a quello generato con il comando `lagr`, che si basa sulla costruzione della base di Lagrange. La spiegazione sta nel fatto che, sebbene teoricamente i due metodi calcolino lo stesso polinomio (!), la matrice di Vandermonde è in questo caso particolarmente malcondizionata (come indica anche il warning fornito da Matlab) e ciò dà luogo ad un risultato affetto da importanti errori di approssimazione.

Esercizio 2.

1. Il metodo (*) non converge, infatti il raggio spettrale della matrice di iterazione risulta essere maggiore di 1:

```
>> A=[1/2 0 0 -1; 0 1 -1/3 0; 0 -1/4 1 0; -1 0 0 4/5];
>> b=A*ones(4,1);
>> B=eye(4)-A;
>> rhoB=max(abs(eig(B)))
```

```
rhoB =
```

```
1.3612
```

2. anche i metodi di Jacobi e Gauss-Seidel non convergono per la stessa ragione

```
>> D=diag(diag(A));
>> E=-tril(A,-1);
>> F=-triu(A,1);
```

```
>> BJ=inv(D)*(E+F);
>> rhoBJ=max(abs(eig(BJ)))
```

```
rhoBJ =
```

```
1.5811
```

```
>> BGS=inv(D-E)*(F);
>> rhoBGS=max(abs(eig(BGS)))
```

```
rhoBGS =
```

```
2.5000
```

3. una semplice permutazione che permette di ottenere una matrice a dominanza diagonale stretta per righe e quindi soddisfare condizione sufficiente per convergenza di Jacobi e Gauss-Seidel è data da

```
>> P=eye(4);
>> P(1,1)=0; P(1,4)=1;
>> P(4,1)=1; P(4,4)=0;
>> Ap=P*A
```

```
Ap =
```

```
-1.0000    0    0    0.8000
    0    1.0000 -0.3333    0
    0   -0.2500    1.0000    0
    0.5000    0    0   -1.0000
```

```
>> bp=P*b;
```

```
>> [x,niter]=gseidel(Ap,bp,zeros(4,1),200,1e-4);
```

```
>> niter =
```

```
11
```

```
>> [x,niter]=jacobi(Ap,bp,zeros(4,1),200,1e-4);
```

```
>> niter =
```

```
8
```

Esercizio 3.

1. Calcoliamo il valore esatto dell'integrale con il toolbox simbolico

```
>> syms x
>> I=int(abs(x.*(x-1).*(x-2)),-1,3)
I =
5
```

2. Calcoliamo il numero di intervalli necessari a punto medio per avere un errore minore di 10^{-5}

```
>> clear all, clear syms
>> a=-1; b=3;
>> f=inline('abs(x.*(x-1).*(x-2))','x');
>> toll=1e-5; err=1+toll;
>> m=1;
>> while(err>toll),
           h=(b-a)/m;
           xm=a+h/2:h:b-h/2;
           Ipm=h*sum(f(xm));
           err=abs(I-Ipm);
           m=m+1;
end
>> m

m =
897
```

3. Tracciamo il grafico della funzione: osserviamo che essa presenta, come è facile immaginare, tre punti angolosi, rispettivamente in $x = 0, 1, 2$. Pertanto è conveniente considerare 4 intervalli di integrazione ciascuno di ampiezza $h = 1$.

```
>> I = simpcomp(f,-1,3,4);
I=
5
```