Esame di Calcolo Numerico

Nome :

Cognome :

Matricola :

Gli esercizi sono di due tipologie:

- (T) indica un esercizio riguardante gli argomenti teorici del corso: tale esercizio va svolto sul foglio che verrà poi consegnato a mano.
- (M) indica un esercizio da svolgere in ambiente MatLab: il codice verrà sviluppato a computer, la consegna verrà effettuata tramite il portale `esami.lamping.unife.it`. **Tutti gli M–files devono contenere nelle prime due righe (commentate) il proprio numero di matricola e subito sopra il proprio cognome (pena l'esclusione dall'appello).**

Gli esercizi denotati con (M+T) riguarderanno argomenti sia teorici che pratici: la parte teorica dovrà essere svolta sul foglio, mentre la parte di programmazione MatLab verrà consegnata tramite il portale.

Istruzioni per il download dei codici forniti

1. collegarsi al sito `esami.lamping.unife.it`
2. accedere con le proprie credenziali di ateneo
3. inserire il codice (ad esempio, 7) dell'esame che verrà fornito in sede di esame
4. nella finestra ALLEGATI cliccare sui file per eseguire il download

Istruzioni per la consegna

1. collegarsi al sito `esami.lamping.unife.it`
2. accedere con le proprie credenziali di ateneo
3. inserire il codice (ad esempio, 7) dell'esame che verrà fornito in sede di esame
4. cliccare sulla voce CONSEGNA
5. caricare i file

Per una maggiore sicurezza, provvedere all'upload dei propri codici ogni venti minuti.

L'esame ha una durata di 4 ore.

Esercizio 1

1. (T) (3 punti) Valutare l'errore inerente e algoritmico nel calcolo dell'espressione (fare il grafo):

$$f(x) = (x+1)^2 - (x-1)^2$$

   Se invece di calcolare $f(x)$ con la formula $(x+1)^2 - (x-1)^2$, si svolge l'epressione, cambia l'errore inerente? Cambia l'errore algoritmico? Quale formula per il calcolo di $f(x)$ risulta più stabile?

2. (M) (2 punti) Scrivere un M-script file che accetta in input il grado n di un polinomio (controllare che il valore introdotto sia un intero positivo), i coefficienti del polinomio, un numero reale a e stampa dopo averli calcolati il valore del polinomio e della derivata prima in a, usando la function `ruffini_horner` in allegato.
   Esempio di input e output nella finestra di comando:

```
n=3
coeff di x^3=3
coeff di x^2=2
coeff di x^1=1
coeff di x^0=9
a=1
Valore del polinomio in 1 =15
Valore della derivata del polinomio in 1 =14
```

Esercizio 2

Si consideri il sistema $Ax = b$, ove A è la matrice quadrata tridiagonale A di ordine 30:

$$A = \begin{pmatrix} 11 & -\sqrt{1} & & & & & & & \\ -1+\frac{1}{2} & 13 & -\sqrt{3} & & & & & & \\ & -1+\frac{1}{3} & 15 & -\sqrt{5} & & & & & \\ & & -1+\frac{1}{4} & 17 & -\sqrt{7} & & & & \\ & & & \ddots & \ddots & \ddots & & & \\ & & & & -1+\frac{1}{28} & 65 & -\sqrt{55} & & \\ & & & & & -1+\frac{1}{29} & 67 & -\sqrt{57} & \\ & & & & & & -1+\frac{1}{30} & 69 & \end{pmatrix}$$

e b un vettore di 30 elementi tutti uguali a 1.

1. (T) (2 punti) Si enunci la condizione necessaria e sufficiente per l'esistenza e l'unicit'a della fattorizzazione LR di Gauss di una generica matrice $A \in \mathbb{R}^{n\times n}$. Si specifichino poi le condizioni solamente sufficienti. Per la matrice definita sopra, vale una di queste condizioni sufficienti e perchè?

2. (M) (3 punti) Si realizzi un M-script file che costruisce la matrice A e il termine noto $b \in \mathbb{R}^{30}$. Si risolva il sistema lineare $Ax = b$ usando la function di Matlab `lu` per calcolare la fattorizzazione LR senza pivoting e le M-function `sollower` e `solupper` in allegato per le risoluzioni dei sistemi triangolari. Calcolare e stampare il residuo normalizzato (in norma infinito). Fornire una stima del numero di condizione del sistema, per verificare se il residuo normalizzato risulta essere una buona stima dell'errore relativo. La soluzione risulta accettabile?

Esercizio 3

1. (T) (3 punti) Risolvere con il metodo di eliminazione di Gauss con pivoting parziale il seguente sistema lineare:

$$\begin{aligned} x_1 + 2x_2 - x_3 &= 2 \\ x_1 - 4x_2 + x_3 &= -2 \\ 5x_1 - 2x_2 &= 3 \end{aligned}$$

   Indicare gli elementi delle matrici L,R e P della fattorizzazione $PA = LR$ usata per risolvere il sistema. Calcolare il determinante di A, usando la fattorizzazione.

2. (M) (1 punto) Realizzare un M-script file che controlli i risultati ottenuti nell'esercizio precdente, utilizzando la M–function `gauss2` per il calcolo di $PA = LR$ e le M-functions `sollower.m` e `solupper.m` per la risoluzione del sistema (vedere codici in allegato). Stampare la soluzione e il residuo normalizzato (in norma infinito).

3. (M) (3 punti) Calcolare la matrice simmetrica $B = A^T A$; verificare se la matrice è definita positiva, eseguendo la fattorizzazione di Cholesky mediante la function Matlab `chol`. Stampare il risultato. Nel caso B sia definita positiva, risolvere il sistema $Bx = c$, con $c = (1,\ 1,\ 1)^T$, usando il fattore di Choleski calcolato e l'operatore di Matlab \ per la risoluzione dei sistemi triangolari. Stampare il risultato.

4. (T) (1 punto) Il risultato ottenuto (B è definita positiva) è prevedibile teoricamente? Spiegare.

Esercizio 4

Si consideri la seguente matrice di ordine 30:

$$A = \begin{pmatrix} 2 & \frac{1}{2} & & & & & & \\ 1 & 4 & \frac{1}{3} & & & & & \\ & 2 & 6 & \frac{1}{4} & & & & \\ & & 3 & 8 & \frac{1}{5} & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & 27 & 56 & \frac{1}{29} & \\ & & & & & 28 & 58 & \frac{1}{30} \\ & & & & & & 29 & 60 \end{pmatrix}$$

1. (T) (3 punti) Dato il sistema $Ax = b$, enunciare con quali metodi iterativi è possibile ottenere la soluzione del sistema, essendo verificate le condizioni di convergenza (motivare). Quale metodo risulta asintoticamente più veloce? Motivare.

2. (M) (3 punti) Realizzare un M-script file Matlab che costruisce mediante istruzioni di memorizzazione sparsa la matrice A; determinare il vettore $b \in \mathbb{R}^{30}$ tale che la soluzione di $Ax = b$ sia un vettore costituito da tutti 1. Utilizzando la function `sor` fornita in Appendice, applicare il metodo SOR per la risoluzione del sistema $Ax = b$, utilizzando una tolleranza `tol = 1e-6` , un numero massimo di iterazioni `maxit = 1000` e vettore iniziale nullo. Si utilizzino diversi valori del parametro di rilassamento (per esempio valori $0.8, 1, 1.1, 1.3, 1.6, 1.9$), stampando per ogni valore il numero di iterazioni impiegate dal metodo per risolvere il sistema. Per quale valore del parametro di rilassamento si ha il numero di iterazioni minore?

Esercizio 5

Si consideri la seguente funzione

$$f(x) = e^{-2x}$$

definita sull'intervallo $[0, 2]$.

1. (M) (3 punti) Mediante la function `pol_lagrange` in allegato si costruisca nella stessa finestra grafica il grafico del polinomio di interpolazione di grado 4 relativo a nodi equispaziati nell'intervallo $[0, 2]$ e il grafico della funzione esatta (per quest'ultimo grafico si considerino almeno 100 punti nell'intervallo $[0, 2]$). In una diversa finestra grafica, si costruisca il grafico del valore assoluto dell'errore e si stampi il suo valore massimo.

2. (T) (3 punti) Fissato un errore massimo pari a 10^{-3}, si determini il grado del polinomio di interpolazione basato su nodi di Chebyshev che approssima la funzione $f(x)$ nell'intervallo $[0, 2]$. Determinare i nodi. (Suggerimento: si osservi che $f^{n+1}(x) = (-1)^{n+1}2^{n+1}e^{-2x}$.)

Esercizio 6

E'stata condotta una prova sperimentale per determinare la distanza y percorsa da un certo corpo dal punto di rilascio in funzione del tempo x. I dati raccolti sono i seguenti:

x	0	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6
y	0	0.0401	0.172	0.330	0.580	0.841	1.10	1.42	1.62

1. (M) (3 punti) Si realizzi un M-script che, dopo aver accettato i dati, ne costruisce l'approssimazione nel senso dei minimi quadrati di grado 2 usando la function di Matlab `polyfit`. Stampare i coefficienti della parabola ottenuta. Realizzare nella stessa finestra grafica la visualizzazione dei dati (con cerchietti) e del grafico della parabola (usando 100 punti nell'intervallo $[0, 1.6]$). Si calcoli la somma degli scarti quadratici (somma dei quadrati dei residui).

2. (T) (3 punti) Spiegare cosa si intende per approssimazione nel senso dei minimi quadrati? Si evidenzino le differenze principali tra un'approssimazione nel senso dei minimi quadrati e l'interpolazione polinomiale.

Esercizio 7

Si consideri l'equazione

$$x = \sin(x + 2)$$

definita nell'intervallo $[0.2, 0.9]$.

1. (T) (2 punti) Verificare che in tale intervallo il metodo è globalmente convergente. Che tipo di velocità di convergenza si può prevedere teoricamente?
2. (M) (2 punti) Realizzare un M-script file che calcola la soluzione con tolleranza pari a 10^{-4} partendo dal valore 0.2. Stampare il numero di iterazioni e la soluzione ottenuta.

1 Allegati

```
function [ r, q] = ruffini_horner( p, a );
% schema di ruffini horner
%  INPUT
%   p: vettore dei coefficienti del polinomio (da quello di grado piu'
%       alto)
%   a: valore in cui calcolare il polinomio
%  OUTPUT
%   r: valore del polinomio in a
%   q: vettore dei coefficienti del polinomio quoziente p(x)/(x-a)
%
q(1)=p(1);
n=length(p)-1;  % grado del polinomio
for i=2:n+1
    q(i)=q(i-1)*a+p(i);
end;
r=q(n+1);
q=q(1:n);
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function [L,R,P]=gauss2(A);
% fattorizzazione di Gauss con pivoting parziale - II versione
% P e' un vettore!!
%
[m,n]=(size(A));
temp=zeros(1,n);
P=1:m;
tol=eps*norm(A,inf);
r=min([m-1,n]);
for k=1:r
   [amax,ind]=max(abs(A(k:m,k)));
   ind=ind+k-1;
   if k~= ind
      aux=P(k);
      P(k)=P(ind);
      P(ind)=aux;
      temp=A(ind,:);
      A(ind,:)=A(k,:);
      A(k,:)=temp;
   end;
   if abs(A(k,k))>tol
      A(k+1:m,k)=A(k+1:m,k)/A(k,k);
      %  operazione di base: aggiornamento mediante diadi
```

```
      A(k+1:m,k+1:n)=A(k+1:m,k+1:n)-A(k+1:m,k)*A(k,k+1:n);
   end;
end;
R=zeros(m,n);L=eye(m);
R=triu(A);
L=L+tril(A(1:m,1:m),-1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function x = solupper(r,b);
% orientato per righe
n = length(b);
x = b;
x(n) = x(n)/r(n,n);
for i = n-1:-1:1
% SDOT
    x(i) = x(i)-r(i,i+1:n)*x(i+1:n);
    x(i) = x(i)/r(i,i);
end;

%%%%%%%%%%%%%%%%%%%%%%%%

function x = sollower(l,b);
% orientato per righe
n = length(l);
x = b;
x(1) = x(1)/l(1,1);
for i = 2:n
%SDOT
    x(i) = x(i)-l(i,1:i-1)*x(1:i-1);
    x(i) =x(i)/l(i,i);
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [x,k]=sor(A,b,x, maxit,tol,omega);
 % metodo di sor - Gauss Seidel estrapolato
 n = max(size(A));

 for k=1:maxit
     xtemp=x;
     for i=1:n
         x(i)= (-A(i,[1:i-1, i+1:n])*x([1:i-1, i+1:n])+b(i))/A(i,i);
         x(i)=(1-omega)*xtemp(i)+omega*x(i);
     end;

     if norm(xtemp-x,inf)<tol*norm(x,inf)

         break;
     end;
```

```
 end;
 if k == maxit
     fprintf('convergenza non raggiunta\n');
 end;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [p,coeff]=pol_lagrange(x,y,punti);
% x: vettore dei nodi o punti di osservazione
% y: vettore delle osservazioni
% punti: punti in cui calcolare il polinomio di Lagrange
%
n1=length(y); coeff=zeros(size(x)); p=zeros(size(punti));
for i=1:n1
    coeff(i)=y(i)/prod(x(i)-x([1:i-1,i+1:n1]));
end;
for k=1:length(punti)
    ij=find(punti(k)==x);
    if isempty(ij)
        % calcolo del polinomio di Lagrange
        temp=prod(punti(k)-x); %
        p(k)=temp*sum(coeff./(punti(k)-x));
    else
        p(k)=y(ij(1));
    end;
end;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


function [x,it]=iterazione(gname,x0,tol,maxit);
x=x0;
for it=1:maxit
   x0=x;
   x=feval(gname,x);
   fprintf('it=%g  x=%g x0=%g \n',it,x,x0);
   if abs(x-x0)<=tol+eps*abs(x0)
      break
   end;
end;
if it>=maxit
    fprintf('Raggiunto max limite di iterazioni \n');
end;

%%%%%%%%%%%%%%%%%%%%%%%%%
```