DIPARTIMENTO DI
ELETTROTECNICA
ED ELETTRONICA

CORSO DI LAUREA IN INGEGNERIA ELETTRONICA

ESAME DI CALCOLO NUMERICO

PROGRAMMI IN MATLAB

ANNO ACCADEMICO 1996 / 1997

Prof. G. Piazza

Dott. T. Politi

Eseguito da Marinelli Vito matr. 507566U

Questa relazione potrete trovarla in formato HTML al sito : http://digilander.iol.it/marinel/vito/calcolo/calcolo.htm

Sempre nel sito : http://digilander.iol.it/marinel potrete trovare altro materiale didattico.

INDICE GENERALE

1. Algebra.m

```
% RISOLUZIONE DI SISTEMI LINEARI E FATTORIZZAZIONI DI MATRICI

help algebra;
dom=0;
while dom==0
  clear;
  dom=0;
  disp('Scegli il problema da risolvere : ');
  err=0;
  while err==0
      disp('1. Calcolo di fattorizzazioni di matrici');
      disp('2. Risoluzione di sistemi lineari');
      s=input('Introduci il numero corrispondente : ');
      if s~=1 & s~=2
        disp('Il valore introdotto é errato.Ricomincia');
      else
        err=1;
      end
  end
  if s==1
    A=input('Introduci la matrice da fattorizzare : ');
  else
    A=input('Introduci la matrice dei coefficienti : ');
  end
  err=0;
  while err==0
      [m,n]=size(A);
      if m<n
        disp('Il numero di righe deve essere maggiore o uguale a quello delle colonne');
        A=input('Introducila nuovamente : ');
      else
        err=1;
      end
  end                                % verifica del tipo di matrice :
  if n~=m
    quadrata=0;
  else                               % se é quadrata;
    quadrata=1;
  end
  rango=rank(A);
  if rango<n
    singolare=1;
  else                               % se é singolare;
    singolare=0;
  end
  if singolare==0 & quadrata==1
    i=1;
    tridiagonale=1;
    while i<=n
      j=1;
      while j<=n
       if j~=i & j~=i-1 & j~=i+1            % se é triangolare;
         if A(i,j)~=0
           tridiagonale=0;
           j=n;
           i=n;
         end
       end
       j=j+1;
```

```
      end
      i=i+1;
   end
   i=1;
   jac=1;
   while i<=n
        if A(i,i)==0                    % se ha gli elementi della diagonale non nulli;
          jac=0;
          i=n;
        end
        i=i+1;
   end
   i=1;
   simmetrica=1;
   while i<=n
      j=1;
      while j~=i & j<=n
          if A(i,j)~=A(j,i)
            simmetrica=0;                  % se é simmetrica;
            j=n;
            i=n;
          end
          j=j+1;
      end
      i=i+1;
   end
   i=1;
   if simmetrica==1
     defpos=1;
     while i<n
         if det(A(1:i,1:i))<=0              % se é simmetrica definita positiva;
           defpos=0;
           i=n;
         end
         i=i+1;
     end
   else
     defpos=0;
   end
else
   simmetrica=0;
   defpos=0;
   jac=0;
   tridiagonale=0;
end
if s==1
   disp('Scegli il tipo di fattorizzazione : ');
   if singolare==1 | quadrata==0
     disp('1. Fattorizzazione LU (non applicabile per questo tipo di matrice)');
   else
     disp('1. Fattorizzazione LU');
   end
   disp('2. Fattorizzazione QR');
   if defpos==0 | quadrata==0
     disp('3. Fattorizzazione di Cholesky (non applicabile per questo tipo di matrice)');
   else
     disp('3. Fattorizzazione di Cholesky ');
   end
   fatt=input('Introduci il numero corrispondente : ');
   err=0;
```

```
while err==0
    if (fatt~=1 & fatt~=2 & fatt~=3)|(fatt==1 & (singolare==1 | quadrata==0))..
      |(fatt==3 & (defpos==0 | quadrata==0))
      fatt=input('Il valore introdotto é errato.Introducilo di nuovo : ');
    else
      err=1;
    end
end                                  % Fattorizzazioni di matrici
if fatt==1
  disp('Scegli il metodo di calcolo della fattorizzazione LU : ');
  disp('1. Utilizzando il metodo di eliminazione di Gauss');
  disp('2. Utilizzando la dimostrazione del teorema di esistenza ed unicità della fattorizzazione');
  disp('3. Tecnica compatta');
  disp('4. Metodo di Doolittle');
  if simmetrica==0
    disp('5. Metodo per matrici simmetriche (non applicabile in questo caso)');
  else
    disp('5. Metodo per matrici simmetriche');
  end
  if tridiagonale==0
    disp('6. Metodo per matrici tridiagonali (non applicabile in questo caso)');
  else
    disp('6. Metodo per matrici tridiagonali');
  end
  metlu=input('Introduci il numero corrispondente : ');
  err=0;
  while err==0
      if (metlu~=1 & metlu~=2 & metlu~=3 & metlu~=4 & metlu~=5 & metlu~=6)|..
        (metlu==5 & simmetrica==0)|(metlu==6 & tridiagonale==0)
        metlu=input('Il valore introdotto é errato.Introducilo di nuovo : ');
      else
        err=1;
      end
  end
  if metlu==1
    [L,U,risp]=lugauss(A,n);                % Metodo di eliminazione di Gauss
  end
  if metlu==2
    [L,U,risp]=ludimost(A,n);               % Utilizzando la dimostrazione
  end
  if metlu==3
    [L,U,risp]=lucompat(A,n);               % Tecnica compatta
  end
  if metlu==4
    [L,U,risp]=ludoolit(A,n);               % Metodo di Doolittle
  end
  if metlu==5
    fc=0;
    [L,U,C,risp,r]=lusimm(A,n,fc);          % Metodo per matrici simmetriche
    if r==1
      disp('La matrice di Cholesky di A é :');
      C
    end
  end
  if metlu==6
    [L,U,risp]=lutrid(A,n);                 % Metodo per matrici tridiagonali
  end
  if risp==0
    disp('la fattorizzazione LU della matrice : ');
    disp(A);
```

```
      disp('é la seguente : ');
      L
      U
      dom=1;
    else
      disp('La fattorizzazione LU per questa matrice non si può ottenere');
      err=0;
      while err==0
          disp('Scegli tra : ');
          disp('1. Calcolare la fattorizzazione QR');
          disp('2. Uscire dal programma');
          s2=input('Introduci il numero corrispondente : ');
          if s2~=1 & s2~=2
            disp('Il valore introdotto é errato.Ricomincia');
          else
            err=1;
          end
      end
      if s2==1
        fatt=2;
      else
        dom=1;
      end
    end
  end
  if fatt==2
    [Q,R,risp]=fattqr(A,m,n);
    if risp==0
      disp('la fattorizzazione QR della matrice : ');
      disp(A);                                % Fattorizzazione QR
      disp('é la seguente : ');
      Q
      R
    else
      disp('La fattorizzazione QR non si può ottenere');
      disp('Ricomincia e utilizza un altro metodo.');
    end
    dom=1;
  end
  if fatt==3
    disp('Scegli il metodo di calcolo della fattorizzazione di Cholesky : ');
    err=0;
    while err==0
        disp('1. Tecnica compatta');            % Fattorizzazione di Cholesky
        disp('2. Metodo di Doolittle');
        s2=input('Introduci il numero corrispondente : ');
        if s2~=1 & s2~=2
          disp('Il valore introdotto é errato.Ricomincia');
        else
          err=1;
        end
    end
    if s2==1
      L=chocomp(A,n);
    else
      L=chodool(A,n);
    end
    disp('La matrice di Cholesky della matrice : ');
    disp(A);
    disp('é la seguente : ');
```

```
      L
      dom=1;
    end
  end
  if s==2 & singolare==0                    % Risoluzione di sistemi lineari
    dom=1;
    b=input('Introduci il vettore colonna  : ');
    err=0;
    while err==0
        [o,h]=size(b);
        if o~=m | h~=1
          disp('Il vettore introdotto ha dimensioni errate.Ricomincia');
          b=input('Introduci nuovamente il vettore colonna  : ');
        else
          err=1;
        end
    end
    if quadrata==1
    if b==zeros(m,1)
      disp('L''unica soluzione é il vettore nullo');
    else
      disp('Scegli il metodo di risoluzione : ');
      err2=0;
      risp2=0;
      while err2==0
        disp('Utilizzare :');
        if risp2==0
          disp('1. - Il metodo di eliminazione Gauss');
        else
          disp('1. - Il metodo di eliminazione Gauss (Non applicabile in questo caso)');
        end
        disp('2. - La strategia di pivoting parziale nel metodo di eliminazione di Gauss');
        disp('3. - La strategia di pivoting totale nel metodo di eliminazione di Gauss');
        disp('4. - La fattorizzazione delle matrici');
        if jac==0
disp('5. - Metodo di Jacobi (Non applicabile in questo caso.Fare un opportuno scambio di righe)');
        else
disp('5. - Metodo di Jacobi ');
        end
        if defpos==1
          disp('6. - Metodo di più ripida discesa');
        else
          disp('6. - Metodo di più ripida discesa (non applicabile per questo tipo di matrici)');
        end
        sist=input('Introduci il numero corrispondente : ');
        err=0;
        while err==0
          if (sist~=1 & sist~=2 & sist~=3 & sist~=4 & sist~=5 & sist~=6)|(sist==1 & risp2==1)|...
            (sist==5 & jac==0)|(sist==6 & defpos==0)
            sist=input('Il valore introdotto é errato.Introducilo di nuovo : ');
          else
            err=1;
          end
        end
        if sist==1                          % Metodo di eliminazione Gauss
          [x,risp]=gauss(A,b,n);
          if risp==0
            disp('Il vettore soluzione del sistema lineare é : ');
            x
            err2=1;
```

```
    else
      disp('Metodo non applicabile ');
      disp('Utilizzare le strategie di pivoting oppure la fattorizzazione QR ');
      risp2=1;
    end
  end
  if sist==2                      % Strategia di pivoting parziale
    x=lupivpar(A,b,n);
    disp('Il vettore soluzione del sistema lineare é : ');
    x
    err2=1;
  end
  if sist==3                      % Strategia di pivoting totale
    x=lupivtot(A,b,n);
    disp('Il vettore soluzione del sistema lineare é : ');
    x
    err2=1;
  end
  if sist==4                      % La fattorizzazione delle matrici
    disp('Scegli il tipo di fattorizzazione : ');
    if risp2==1
      disp('1.Fattorizzazione LU (non applicabile in questo caso)');
    else
      disp('1. Fattorizzazione LU');
    end
    disp('2. Fattorizzazione QR');
    if defpos==0
      disp('3. Fattorizzazione di Cholesky (non applicabile in questo caso)');
    else
      disp('3. Fattorizzazione di Cholesky ');
    end
    sistf=input('Introduci il numero corrispondente : ');
    err=0;
    while err==0
      if (sistf~=1 & sistf~=2 & sistf~=3 & sistf~=4) | (sistf==1 & risp2==1) |...
        (sistf==3 & defpos==0)
        sistf=input('Il valore introdotto é errato.Introducilo di nuovo : ');
      else
        err=1;
      end
    end
    if sistf==1                   % Utilizzando la fattorizzazione LU
      if tridiagonale==1             % più conveniente
        [L,U,risp]=lutrid(A,n);
      else
        if simmetrica==1
          fc=1;
          [L,U,C,risp,r]=lusimm(A,n,fc);
        else
          [L,U,risp]=ludoolit(A,n);
        end
      end
      if risp==0 & tridiagonale==1
        z(1)=b(1);
        for i=2:n
          z(i)=b(i)-L(i,i-1)*z(i-1);
        end
        x(n)=z(n)/U(n,n);
        for i=n-1:-1:1
          x(i)=(z(i)-U(i,i+1)*x(i+1))/U(i,i);
```

```
        end
      end
      if risp==0 & tridiagonale==0
        x=lurisol(L,U,b,n);
      end
      if risp==0
        disp('Il vettore soluzione del sistema lineare é : ');
        x
        err2=1;
      else
        disp('Metodo non applicabile ');
        disp('Utilizzare le strategie di pivoting oppure la fattorizzazione QR ');
        risp2=1;
      end
    end
    if sistf==2                         % Utilizzando la fattorizzazione QR
      [x,a,risp]=minquad(A,b,m,n);
      disp('Il vettore soluzione del sistema lineare é : ');
      x
      err2=1;
    end
    if sistf==3                         % Utilizzando la fattorizzazione di Cholesky
      L=chodool(A,n);
      disp('Il vettore soluzione del sistema lineare é : ');
      x=lurischo(L,b,n)
      err2=1;
    end
  end
  if sist==5
    [x,risp]=jacobi(A,b,n);             % Metodo di Jacobi
    if risp==0
      disp('Il vettore soluzione del sistema lineare é : ');
      x=x';
      x
      err2=1;
    else
      disp('Metodo non applicabile perché non convergente');
      disp('Utilizzare altri metodi');
    end
  end
  if sist==6                            % Metodo di più ripida discesa
    e=input('Introduci l''errore massimo che si può commettere (al massimo 0.1) : ');
    err=0;
    while err==0
        if e>0.1 | e<=0
          disp(' Il valore dell''errore massimo introdotto non é corretto');
          e=input('Introducilo nuovamente : ');
        else
          err=1;
        end
    end
    disp('Il vettore soluzione del sistema lineare é : ');
    x=meripdis(A,b,n,e)
    err2=1;
  end
 end
end
else
 B=A;
 B(1:m,n+1)=b;
```

```
        if rank(B)==n+1
          disp('Il sistema non ammette soluzioni');
          disp('Verrà calcolata quindi la soluzione di minima norma ');
          disp('utilizzando il problema dei minimi quadrati ');
          [x,a,risp]=minquad(A,b,m,n);
          disp('la soluzione ai minimi quadrati é :');
          x
          disp('dove il vettore residuo ha norma 2 pari a : ');
          a
        else
          [x,a,risp]=minquad(A,b,m,n);
          disp('la soluzione é :');
          x
        end
      end
    end
    if s==2 & singolare==1                          % caso di matrice singolare
 disp('Il sistema non si può risolvere perché la matrice dei coefficienti non ha rango massimo');
      disp('Infatti ha rango :');
      rango
      dom=1;
    end
    if dom==1
      fine=input('Se vuoi ricominciare nuovamente premi il tasto "1" :');
      if fine==1
        dom=0;
      end
    end
  end
```

```
% CALCOLO DEGLI AUTOVALORI DI UNA MATRICE
help autoval;
inizio=0;
while inizio==0
clear;
inizio=0;
A=input('Introduci la matrice (di ordine almeno 2x2) : ');
dom=0;
while dom==0
    [n,m]=size(A);
    if m~=n | n==1
      disp('La matrice deve essere quadrata e di ordine almeno 2x2');
      A=input('Introduci nuovamente la matrice:');
    else
      dom=1;
    end
end
if rank(A)==0
   disp('Hai introdotto una matrice nulla per cui i suoi autovalori sono tutti nulli');
else
singolare=1;
if rank(A)<n
  singolare=0;
end
i=1;                                         % verifica del tipo di matrice:
simmetrica=1;
while i<=n
    j=1;                                     % se é simmetrica
    while j~=i & j<=n
        if A(i,j)~=A(j,i)
          simmetrica=0;
          tridiagonale=0;
          j=n;
          i=n;
        end
        j=j+1;
    end
    i=i+1;
end
if simmetrica==1                                  % se é tridiagonale simmetrica
  i=3;
  tridiagonale=1;
  while i<=n
      j=1;
      while j<=i-2
          if A(i,j)~=0
            tridiagonale=0;
            j=i;
            i=n;
          end
          j=j+1;
      end
  i=i+1;
  end
end
disp('Scegliere tra i seguenti metodi di calcolo degli autovalori: ');
dom=0;
while dom==0
    disp('1. Metodo delle potenze (per il calcolo dell''autovalore di modulo minimo)');
```

```
    if singolare==0
      disp('                    (non applicabile per questo tipo di matrice) ');
    end
    disp('2. Metodo delle potenze (per il calcolo dell''autovalore di modulo massimo)');
    disp('3. Metodo delle potenze inverse (per il calcolo di un particolare autovalore)');
    disp('4. Metodo di calcolo di tutti gli autovalori reali di matrici triangolari simmetriche');
    if tridiagonale==0
      disp('                    (non applicabile per questo tipo di matrice) ');
    end
    if simmetrica==1
      disp('5. Metodo di calcolo dell''autovalore minimo per matrici simmetriche');
      disp('6. Metodo di calcolo dell''autovalore massimo per matrici simmetriche');
    else
disp('5. Metodo di calcolo dell''autovalore di modulo minimo per matrici simmetriche ');
      disp('                    (non applicabile per questo tipo di matrice)');
disp('6. Metodo di calcolo dell''autovalore massimo per matrici simmetriche');
      disp('                    (non applicabile per questo tipo di matrice)');
    end
    disp('7. Metodo QR (per il calcolo di tutti gli autovalori)');
    if simmetrica==0
      disp(' ATTENZIONE : i primi 3 metodi convergono sempre SOLO SE la matrice ha solo autovalori reali ');
    end
    s=input('Introduci il numero corrispondente : ');
    err=0;
    while err==0
      if (s~=1 & s~=2 & s~=3 & s~=4 & s~=5 & s~=6 & s~=7 ) | (s==1 & singolare==0) |...
        (s==4 & tridiagonale==0) | ((s==5 | s==6) & simmetrica==0)
        s=input('Il valore introdotto é errato.Introducilo di nuovo : ');
      else
        err=1;
      end
  end

% METODO DELLE POTENZE PER IL CALCOLO DELL'AUTOVALORE DI MODULO MINIMO oppure
% CALCOLO DELL' AUTOVALORE DI MODULO MINIMO DI UNA MATRICE SIMMETRICA
  if s==1 | s==5
    if simmetrica==0
      [l,risp]=autmin(A,n);
    else
      [l,risp]=autmin(A,n);
    end
    if risp==1
      disp('Metodo non applicabile');
      disp('Utilizzare altri metodi se possibile');
    else
      dom=1;
      disp('L''autovalore di modulo minimo della matrice : ');
      disp(A);
      disp('é il seguente : ');
      l
    end
  end

% METODO DELLE POTENZE PER IL CALCOLO DELL'AUTOVALORE DI MODULO MASSIMO oppure
% CALCOLO DELL' AUTOVALORE DI MODULO MINIMO DI UNA MATRICE SIMMETRICA
  if s==2 | s==6
    if simmetrica==0
      l=autmax(A,n);
    else
      l=autmaxsi(A,n);
```

```
    end
    dom=1;
    disp('L''autovalore di modulo massimo della matrice : ');
    disp(A);
    disp('é il seguente : ');
    l
  end

% METODO DELLE POTENZE INVERSE
  if s==3
    [H,risp]=aupotinv(A,n,simmetrica);
    if risp==1
      disp('Metodo non applicabile');
      disp('Utilizzare altri metodi se possibile oppure fissare un autovalore approssimato diverso :');
      disp('Scegli nuovamente : ');
    else
      dom=1;
      disp('L''autovalore della matrice : ');
      disp(A);
      disp('é il seguente : ');
      H
    end
  end

% METODO DI CALCOLO DI TUTTI GLI AUTOVALORI REALI DELLE MATRICI TRIDIAGONALI SIMMETRICHE
  if s==4
    e=input('Introduci l''errore massimo che si può commettere (massimo 0.1) : ');
    err=0;
    while err==0
        if e>0.1 | e<=0
          disp(' Il valore dell''errore massimo introdotto non é corretto');
          e=input('Introducilo nuovamente : ');
        else
          err=1;
        end
    end
    disp('Tutti gli autovalori della matrice  :');
    disp(A);
    disp('sono i seguenti : ');
    l=auttrsim(A,n,e);
    l=l'
    dom=1;
  end

% METODO QR
  if s==7
    [l,risp]=autmetqr(A,n);
    dom=1;
    if risp==0
      disp('Tutti gli autovalori della matrice  :');
      disp(A);
      disp('sono i seguenti : ');
      l=l'
    else
      disp('Metodo non applicabile. Provare con gli altri se é possibile');
    end
  end
end
end
inizio=input('Se vuoi ricominciare nuovamente premi il tasto "0" :');
end
```

```
% INTERPOLAZIONE DI FUNZIONI

help interpol;
z=1;
risp=0;
dom=0;
while dom==0
  clear;
  dom=0;
  disp('Scegli il tipo di interpolazione : ');
  disp('1. Interpolazione di Lagrange');
  disp('2. Interpolazione di Newton');
  disp('3. Utilizzando lo schema di Neville');
  disp('4. Interpolazione di Hermite');
  disp('5. Interpolazione con spline lineari');
  disp('6. Interpolazione con spline cubiche naturali');
  disp('7. Interpolazione con spline cubiche complete');
  disp('8. Approssimazione polinomiale ai minimi quadrati');
  s=input('Introduci il numero corrispondente : ');
  err=0;
  while err==0
      if s~=1 & s~=2 & s~=3 & s~=4 & s~=5 & s~=6 & s~=7 & s~=8
        s=input('Il valore introdotto é errato.Introducilo di nuovo : ');
      else
        err=1;
      end
  end
  disp('Introduci : ');
  err=0;
  while err==0
      x=input('- il vettore colonna dei nodi interpolanti : ');
      [n,h]=size(x);
      if h~=1
        disp('Non é stato introdotto un vettore colonna.Ricomincia');
        disp('Introduci : ');
      else
        if n<=1
          disp('I nodi devono essere almeno 2.Introduci nuovamente : ');
        else
          err=1;
        end
      end
  end
  non=0;
  if s==6 | s==7
    if n<3
      disp('Per applicare questo metodo i nodi devono essere almeno 3');
      disp('Fissare più nodi o adoperare un altro metodo :');
      non=1;
      risp=1;
    end
  end
  if non==0;
   err=0;
   while err==0
        y=input('- il vettore colonna dei corrispondenti valori della funzione : ');
        [c,h]=size(y);
        if n~=c | h~=1
          disp('ATTENZIONE : deve avere le stesse dimensioni dell''altro vettore.RICOMINCIA');
              disp('Introduci : ');
```

```
    else
      err=1;
    end
  end
  if s==4
    err=0;
    while err==0
      z=input('- il vettore colonna con corrispondenti valori delle derivate prime : ');
      [t,h]=size(z);
      if t~=n | h~=1
        disp('ATTENZIONE : deve avere le stesse dimensioni dei primi 2 vettori.RICOMINCIA');
        disp('Introduci : ');
      else
        err=1;
      end
    end
  end
  q=1;                              % ordinamento dei nodi
  while q==1
    q=0;
    for i=1:n-1
      if x(i)>x(i+1)
        u=x(i);
        x(i)=x(i+1);
        x(i+1)=u;
        u=y(i);
        y(i)=y(i+1);
        y(i+1)=u;
        if s==4
          u=z(i);
          z(i)=z(i+1);
          z(i+1)=u;
        end
        q=1;
      end
    end
  end
  graf=input('Se conosci la funzione f corrispondente ai punti introdotti premere "1" : ');
  if graf==1
    err=0;
    while err==0
        disp('Introduci : ');
        F=input('- Il nome della function contenente la funzione : ','s');
        if exist(F)==0
          disp('La funzione introdotta non esiste,introduci : ');
        else
          err=1;
        end
    end
  end
  err=0;
  while err==0
 xi=input('- il vettore colonna con i punti nei quali calcolare la funzione interpolante : ');
      [c,h]=size(xi);
      if h~=1
        disp('Non é stato introdotto un vettore colonna.Ricomincia');
        disp('Introduci : ');
      else
        if min(xi)<min(x) | max(xi)>max(x);
          disp('Non conviene eseguire estrapolazioni.Ricomincia');
```

```
        disp('Introduci : ');
      else
        err=1;
      end
    end
end
risp=0;
if s==1                        % Interpolazione di Lagrange
  L=intlagr(x,y,xi,n,c);
  dom=1;
end
if s==2                        % Interpolazione di Newton
  L=intnewt(x,y,xi,n,c);
  dom=1;
  L=L';
end
if s==3                        % Utilizzando lo schema di Neville
  L=intnevil(x,y,xi,n,c);
  dom=1;
  L=L';
end
if s==4                        % Interpolazione di Hermite
  L=intherm(x,y,z,xi,n,c);
  asc=x(1):.07:x(n);
  asc=asc';
  [m2,h]=size(asc);
  ord=intherm(x,y,z,asc,n,m2);
  plot(asc,ord,'w',xi,L,'+r',x,y,'og');
  title('Interpolazione di Hermite');
  dom=1;
  ord=0;
end
if s==5                        % Interpolazione con spline lineari
  L=spllin(x,y,xi,n,c);
  dom=1;
end
if s==6                        % Interpolazione con spline cubiche naturali
  [L,risp]=splcubna(x,y,xi,n,c);
  if risp==1
    disp('Non é possibile fare questo tipo di interpolazione');
    disp('adoperane un altro : ');
  else
    dom=1;
  end
end
if s==7                        % Interpolazione con spline cubiche complete
  k0=input('- il valore della derivata prima nel nodo più piccolo fissato : ');
  k1=input('- il valore della derivata prima nel nodo più grande fissato : ');
  [L,risp]=splcubco(x,y,xi,n,c,k0,k1);
  if risp==1
    disp('Non é possibile fare questo tipo di interpolazione');
    disp('adoperane un altro : ');
  else
    dom=1;
  end
end
if s==8                        % Approssimazione polinomiale ai minimi quadrati
  err=0;
  while err==0
   m=input('- Il grado del polinomio interpolante (minore del numero di nodi introdotto) : ');
```

```
        if m>n | round(m)~=m
 disp('Il grado del polinomio interpolante é maggiore del numero di nodi introdotto.Ricomincia');
           disp('Introduci :');
        else
           err=1;
        end
    end
    [L,risp]=intminqu(x,y,xi,n,m,c);
    if risp==1
          disp('Non é possibile fare questo tipo di interpolazione');
          disp('adoperane un altro : ');
       else
          dom=1;
          L=L';
       end
    end
  end
  if risp==0
     if graf==1
        hold on;
        asc=x(1):0.1:x(n);
        [h,k]=size(asc);
        for i=1:k
           ord(i)=feval(F,asc(i));
        end
        plot(asc,ord,'b');
        gtext('blu=funzione f');
     end
     xlabel('asse x');
     ylabel('asse y');
     gtext('+=valori calcolati');
     gtext('o=nodi interpolanti');
     gtext('nero=funzione interpolante');
     hold off;
  end
  if dom==1
     disp('Il vettore soluzione y corrispondente ai punti x immessi é : ');
     disp('x=             y=');
     disp([xi L]);
     dom=input('Se vuoi ricominciare premi "0" : ');
  end
 end
end
```

```
% CALCOLO DELLO ZERO DI UNA FUNZIONE

help zero;
disp('Scegliere tra i seguenti metodi di calcolo dello zero di una funzione: ');
q=0;
while q==0
   clear;
   q=0;
   err=0;
   while err==0
         disp('1. Metodo di bisezione');
         disp('2. Metodo di Newton-Rapson');
         disp('3. Metodo delle secanti a due punti');
disp('4. Metodo di Newton a doppio passo per il calcolo di tutti gli zeri reali di un generico polinomio');
         s=input('Introduci il numero corrispondente : ');
         if s~=1 & s~=2 & s~=3 & s~=4
            disp('Scelta errata');
            disp('Ricomincia');
         else
            err=1;
         end
   end
   if s~=4
      disp('INTRODUCI:');
      err=0;
      while err==0
            F=input('- Il nome della function contenente la funzione : ','s');
            if exist(F)==0
               disp('La funzione non esiste, introduci :');
            else
               err=1;
            end
      end
      e=input('- L''errore massimo che si può commettere (massimo 0.1) : ');
      err=0;
      while err==0
         if e>0.1 | e<=0
            disp(' Il valore dell''errore massimo introdotto non é corretto');
            e=input('Introducilo nuovamente : ');
         else
            err=1;
         end
      end
   end
   if s==1

% METODO DI BISEZIONE

      a=input('- L''estremo inferiore dell''intervallo : ');
      b=input('- L''estremo superiore dell''intervallo tale che f(a)*f(b)<0 : ');
      if a>=b
         disp('L''estremo inferiore deve essere strettamente minore di quello superiore');
         disp('Ricomincia dall''inizio .');
         disp('Scegli nuovamente il metodo di calcolo : ');
      else
       w1=feval(F,a);
       w2=feval(F,b);
       if w1*w2>0
         disp('Il metodo non è applicabile .');
             disp('Infatti agli estremi la funzione assume lo stesso segno.');
```

```
        disp('Scegli un altro metodo :');
      else if w1==0
          disp('Lo zero è : ');
          c=a;
          g=0;
        else if w2==0
            disp('Lo zero è : ');
            c=b;
            g=0;
          else
            k=(log((b-a)/e))/log(2);
            bk=b;
            ak=a;
            k=ceil(k);
            i=1;
            while i<=k
                c=(ak+bk)/2;
                g=feval(F,c);
                if g==0
                  disp('Lo zero é : ');
                  i=k;
                else
                  w=feval(F,ak);
                  if g*w>0
                    ak=c;
                  else bk=c;
                  end
                end
                i=i+1;
              end
          end
          while g>e
           c=(ak+bk)/2;
           g=feval(F,c);
           if g==0
             disp('Lo zero é : ');
             c
           else
             w=feval(F,ak);
             if g*w>0
               ak=c;
             else bk=c;
             end
           end
          end
        end
        if g~=0
          disp('Lo zero é approssivamente: ');
        end
        disp(c);
        q=1;
      end
     end
   end
   if s==2

% METODO DI NEWTON-RAPSON
     err=0;
     while err==0
        G=input('- Il nome della function contenente la derivata prima della funzione : ','s');
```

```
        if exist(G)==0
          disp('La funzione non esiste, introduci :');
        else
          err=1;
        end
    end
    x0=input('- L''approssimazione iniziale dello zero da calcolare : ');
    i=1;
    g=feval(F,x0);
    if g==0
      disp('Lo zero é : ');
      x1=x0;
      i=2;
      w=1;
    else
      w=feval(G,x0);
      if w==0
        disp('Metodo non applicabile');
        disp('Infatti si annulla la derivata prima della funzione in esame.');
        disp('Scegli un altro metodo o cambia l''approssimazione iniziale :');
        i=2;
      else
        x1=x0-(g/w);
      end
    end
    while i==1 & (abs(x1-x0)>=e | abs(g)>=e)
        x0=x1;
        g=feval(F,x0);
        if g==0
          disp('Lo zero é : ');
          i=2;
        else
          w=feval(G,x0);
          if w==0
            disp('Metodo non applicabile');
            disp('Infatti si annulla la derivata prima della funzione in esame.');
            disp('Scegli un altro metodo o cambia l''approssimazione iniziale :');
            i=2;
          else
            x1=x0-(g/w);
          end
        end
    end
    if w~=0
      if g~=0
        disp('Lo zero é approssivamente: ');
      end
      q=1;
      x0
    end
  end
  if s==3

% METODO DELLE SECANTI A DUE PUNTI

    dom=0;
    while dom==0
        x0=input('- La prima approssimazione iniziale dello zero da calcolare : ');
        x1=input('- La seconda approssimazione iniziale dello zero da calcolare : ');
        if x0==x1
```

```
          disp('I due punti devono essere differenti.Ricomincia');
          disp('Introduci nuovamente : ');
        else
          b=feval(F,x0);
          c=feval(F,x1);
          if b==0
            disp('Lo zero é :');
            b
            q=1;
          else
            i=1;
            while i==1  & (abs(x1-x0)>e | abs(c)>e)
                if c==b
                  disp('Metodo non applicabile');
                  disp('Scegline un altro oppure cambia i dati iniziali:');
                  i=2;
                else
                  q=1;
                  if c==0
                    disp('Lo zero é :');
                    i=2;
                    x2=c;
                  else
                    x2=x1-c*((x1-x0)/(c-b));
                    b=c;
                    x0=x1;
                    x1=x2;
                    c=feval(F,x1);
                  end
                end
            end
            if q==1
              if c~=0
                disp('Lo zero é approssivamente: ');
              end
              disp(x2);
            end
          end
          dom=1;
        end
    end
  end
  if s==4

% METODO DI NEWTON A DOPPIO PASSO

    disp('Scegli tra :');
    err=0;
    while err==0
        disp('1. - Introdurre a priori il numero di zeri reali del polinomio interessato');
        disp('2. - Applicare altri metodi ');
        pr=input('Introduci il numero corrispondente : ');
        if pr~=1 & pr~=2
          disp('Scelta errata');
          disp('Ricomincia');
        else
          err=1;
        end
    end
    if pr==1
```

```
disp('Introduci :');
disp('- il vettore dei coefficienti del polinomio a partire dal termine');
vett=input('  noto fino ad arrivare a quello di grado massimo : ');
err=0;
err1=0;
while err==0
    [o,h]=size(vett);
    if o~=1 & h~=1
      vett=input('  Non é stato introdotto un vettore.Inseriscilo : ');
      err1=1;
    else
      if o==1 & h==1
        vett=input('  Il vettore deve avere almeno 2 elementi.Inseriscili : ');
        err1=1;
      else
        [o,h]=size(vett);
        if o==1
          vett=vett';
          o=h;
        end
        j=o;
        while j>0
          if vett(j)==0                      % controllo dell'effettivo
            vett=vett(1:j-1);                  % grado del polinomio
          else
            j=0;
          end
            j=j-1;
        end
        [o,h]=size(vett);
        if o==0
          vett=input('  E''stato introdotto un vettore nullo.Inseriscilo nuovamente : ');
          err1=1;
        else
          err=1;
        end
      end
    end
end
if err1==1
  disp('Introduci :');
end
disp('- il numero di zeri reali del polinomio (nel caso di polinomi di secondo grado');
num=input('  verranno calcolati tutti gli zeri compreso quelli complessi) : ');
err=0;
err1=0;
while err==0
    if num<0 | round(num)~=num
      num=input('  Valore errato.Inserisci quello giusto : ');
      err1=1;
    else
      if num>o-1
        disp('  il numero di zeri non può superare il grado del polinomio');
        num=input('  Inserisci quello giusto : ');
        err1=1;
      else
        err=1;
      end
    end
end
```

```
        if err1==1
          disp('Introduci :');
        end
        g=input('- l''errore massimo di approssimazione (al massimo 0.1): ');
        err=0;
        while err==0
            if g>0.1 | g<=0
              g=input('  Valore errato.Inserisci quello giusto : ');
            else
              err=1;
            end
        end
        if o==1
          disp('Il polinomio introdotto ha grado zero');
        else
          if num==0
            disp('Il polinomio non ha zeri reali');
          else
            [b,risp,dom]=zerpol(vett,num,g);
            if risp==0
              disp('Gli zeri del polinomio sono : ');
              disp(b);
              if dom==1
                disp('OSS.:in questo caso é stato possibile calcolare anche gli zeri complessi');
              end
            else
              disp('Metodo non applicabile in questo caso');
            end
          end
        end
        q=1;
      end
    end
    if q==1
      fine=input('Se vuoi ricominciare nuovamente premi il tasto "1" :');
      if fine==1
        q=0;
        disp('Scegli di nuovo tra :');
      end
    end
end
```

```
% FORMULE DI QUADRATURA

help formquad;
inizio=0;
while inizio==0
clear;
inizio=0;
disp('INTRODUCI:');
err=0;
while err==0
     F=input('- Il nome della function contenente la funzione : ','s');
     if exist(F)==0
        disp('La funzione introdotta non esiste,introduci : ');
     else
        err=1;
     end
end
a=input('- L''estremo inferiore di integrazione : ');
b=input('- L''estremo superiore di integrazione : ');
if a==b
   disp(' L''integrale definito é nullo ');
else
    err=0;
    while err==0
     n=input('- Il numero di suddivisioni dell''intervallo di integrazione :');
     if n<=0 | round(n)~=n
       disp('Il numero di suddivisioni deve essere un intero naturale non nullo.Introduci');
     else
       err=1;
     end
    end
    h=(b-a)/n;
    x(1)=a;                                   % determinazione del vettore
    for i=2:n                                 % sul quale applicare la
        x(i)=x(i-1)+h;                          % formula di quadratura
    end
    x(n+1)=b;
    for i=1:n+1
        y(i)=feval(F,x(i));
    end
    e=0;
    while e==0
      disp('Scegliere tra le seguenti formule di quadratura : ');
      disp('1. Trapezi composta');
      disp('2. Simpson composta');
      s=input('Introduci il numero corrispondente : ');
      q=0;
      if s~=1 & s~=2
        disp('Scelta errata.Ricomincia :');
      else
        risp=1;
        while q==0 & risp==1
          if s==1 & risp==1                          % Trapezi composta
             c=0;
             for i=2:n
                 c=c+y(i);
             end
             T=(h/2)*(y(1)+y(n+1))+h*c
             q=1;
           elseif s==2 & risp==1                         % Simpson composta
```

```
            if rem(n,2)~=0
               disp('Il numero di intervalli deve essere pari');
               disp('Allora premere :');
               disp('1. Per applicare la formula dei trapezi composta ');
               disp('2. Per uscire dal programma ');
               risp=input('Introduci il numero corrispondente : ');
               s=1;
               if risp~=1
                  risp=2;
                  q=1;
               end
            else
               c=0;
               for i=1:n/2-1
                   c=c+y(2*i+1);
               end
               d=0;
               for i=1:n/2
                   d=d+y(2*i);
               end
               T=(h/3)*(y(1)+y(n+1)+2*c+4*d)
               q=1;
            end
         end
        end
        e=1;
      end
    end
end
inizio=input('Se vuoi ricominciare nuovamente premi il tasto "0" :');
end
```

```
% METODI MULTISTEP PER EQUAZIONI DIFFERENZIALI

help metmulst;
inizio=0;
while inizio==0
clear;
inizio=0;
disp('INTRODUCI:');
err=0;
while err==0
    F=input('- Il nome della function contenente la funzione f(t,y): ','s');
    if exist(F)==0
      disp('La funzione non esiste, introduci :');
    else
      err=1;
    end
end
err=0;
while err==0
  a=input('- L''estremo inferiore dell''intervallo di definizione : ');
  b=input('- L''estremo superiore dell''intervallo di definizione : ');
  if a>=b
     disp('I due estremi dell''intervallo sono errati.Introduci :');
  else
     err=1;
  end
end
y(1)=input('- La condizione iniziale corrispondente all''estremo inferiore dell''ntervallo : ');
err=0;
while err==0
     n=input('- Il numero di suddivisioni dell''intervallo di definizione :');
     if n<=0 | round(n)~=n
       disp('Il numero di suddivisioni deve essere un intero naturale non nullo.Introduci');
     else
       err=1;
     end
end
disp('Scegliere tra : ');
err=0;
while err==0
     disp('1. Metodo di Eulero implicito');
     disp('2. Metodo di Eulero esplicito');
     disp('3. Metodo dei trapezi');
     s=input('Introduci il numero corrispondente : ');
     if s~=1 & s~=2 & s~=3
        disp('Scelta errata.Ricomincia :');
     else
        err=1;
     end
end
h=(b-a)/n;
t(1)=a;
for i=2:n+1
    t(i)=t(i-1)+h;
end
t(n+1)=b;
if s==1 | s==3
   err=0;
   while err==0
     G=input('Introduci il nome della function contenente laderivata prima della funzione rispetto a y : ','s');
```

```
    if exist(G)==0
      disp('La funzione non esiste, introduci :');
    else
      err=1;
    end
  end
  e=input('Introduci l''errore massimo che si può commettere (massimo 0.1) : ');
  err=0;
  while err==0
    if e>0.1 | e<=0
      disp(' Il valore dell''errore massimo introdotto non é corretto');
      e=input('Introducilo nuovamente : ');
    else
      err=1;
    end
  end
end
if s==1
  [y,risp]=eulimpl(F,G,t,y,n,h,e);           % METODO DI EULERO IMPLICITO
end
if s==2
  risp=0;
  y=eulespl(F,t,y,n,h);                      % METODO DI EULERO ESPLICITO
end
if s==3
  [y,risp]=mettrap(F,G,t,y,n,h,e);           % METODO DEI TRAPEZI
end
if risp==0
  disp('In corrispondenza dei seguenti punti t ');
  disp('La funzione soluzione dell''equazione differenziale assume i valori y che sono ');
  t=t';
  y=y';
  disp('t=            y=');
  disp([t y]);
else
  disp('Metodo non applicabile.Utilizza gli altri metodi');
end
inizio=input('Se vuoi ricominciare nuovamente premi il tasto "0" :');
end
```

FUNZIONI ADOPERATE

1. Funzione.m

```
function f=funzione(x);
f=sin(x)+x^2-3*x;
```

2. Derivata.m

```
function f=derivata(x);
f=cos(x)+2*x-3;
```

3. Fun.m

```
function f=fun(t,y);
f=-2*t*y+t*exp(-t*t);
```

4. Fund.m

```
function f=fund(t,y);
f=-2*t;
```

5. Funz.m

```
function f=funz(x);
f=10*sin(x)+0.25*(x^2-3*x);
```

1. Autpotinv.m

```
% METODO DELLE POTENZE INVERSE PER IL CALCOLO DEGLI AUTOVALORI

function [H,risp]=aupotinv(A,n,simmetrica);
% H          é l'autovalore da calcolare
% risp       controlla se il metodo é applicabile
% A          é la matrice in cui calcoliamo l'autovalore
% n          é la dimensione della matrice A
% simmetrica  tiene conto del fatto che A può essere simmetrica

lc=input('Inserisci il valore approssimato dell''autovalore da calcolare : ');
B=A-lc*eye(n);
if rank ( B )<n
  risp=1;
else
  if simmetrica==1
    [l,risp]=autminsi(B,n);
  else
    [l,risp]=autmin(B,n);
  end
  if risp==0
    H=lc+l;
  else
    H=0;
  end
end
```

2. Autmax.m

```
% QUESTO ALGORITMO DETERMINA L'AUTOVALORE DI MODULO PIU' GRANDE DI UNA MATRICE

function l=autmaxmo(A,n);
% l         é l'autovalore da calcolare
% A         é la matrice in cui calcoliamo l'autovalore
% n         é la dimensione della matrice A

x=diag(eye(n));
Kmax=input('Introduci il numero massimo di iterate : ');
err=0;
while err==0
    if Kmax<=0 | round(Kmax)~=Kmax
      disp(' ');
      disp(' Il valore introdotto non é corretto');
      Kmax=input('Introducilo nuovamente : ');
    else
      err=1;
    end
end
for k=0:Kmax
   y=A*x;
   j=0;
   i=1;
   while j==0
       if abs(x(i))==1
         j=i;
       end
       i=i+1;
   end
   l=y(j)/x(j);
   x=y/max(abs(y));
end
```

3. Autmaxsi.m

```
% QUESTO ALGORITMO DETERMINA L' AUTOVALORE DI MODULO PIU' GRANDE DI UNA MATRICE
% SIMMETRICA

function l=autsimm(A,n);
% l        é l'autovalore da calcolare
% A        é la matrice in cui calcoliamo l'autovalore
% n        é la dimensione della matrice A

x=diag(eye(n));
Kmax=input('Introduci il numero massimo di iterate : ');
err=0;
while err==0
     if Kmax<=0 | round(Kmax)~=Kmax
        disp(' ');
        disp(' Il valore introdotto non é corretto');
        Kmax=input('Introducilo nuovamente : ');
     else
        err=1;
     end
end
for k=0:Kmax
   y=A*x;
   l=(x'*y)/(norm(x))^2;
   x=y/norm(y);
end
```

4. Autmetqr.m

```
% CALCOLO DEGLI AUTOVALORI DI UNA MATRICE CON IL METODO QR

function l=autmetqr(A,n);
% l        é il vettore con tutti gli autovalori da calcolare
% A        é la matrice in cui calcoliamo l'autovalore
% n        é la dimensione della matrice A

kmax=input('Introduci il numero massimo di iterate : ');
err=0;
while err==0
     if kmax<=0 | round(kmax)~=kmax
        disp(' ');
        disp(' Il valore introdotto non é corretto');
        kmax=input('Introducilo nuovamente : ');
     else
        err=1;
     end
end
i=1;
while i<=kmax
   [Q,R,risp]=fattqr(A,n,n);                 % Calcolo della fattorizzazione QR
   if risp==0                                % ad ogni passo
      A=R*Q;
      i=i+1;
   else
      l=0;
      i=kmax+1;
   end
end
if risp==0
i=2;
```

```
while i<=n
   if abs(A(i,i-1))>=eps
      b=A(i-1,i-1)+A(i,i);
      c=A(i-1,i-1)*A(i,i)-A(i-1,i)*A(i,i-1);
      Delta=sqrt(b*b-4*c);
      l(i-1)=(b-Delta)/2;                    % estrapolazione dalla matrice
      l(i)=(b+Delta)/2;                      % ottenuta all'ultima iterata
      i=i+2;
   else
      l(i-1)=A(i-1,i-1);
      if i==n
         l(i)=A(i,i);
      end
      i=i+1;
   end
end
if i==n+1
   l(n)=A(n,n);
end
end
```

5. Autmin.m

```
% QUESTO ALGORITMO DETERMINA L'AUTOVALORE DI MODULO PIU' PICCOLO DI UNA MATRICE

function [l,risp]=autminmo(A,n);
% l        é l'autovalore da calcolare
% risp     controlla se il metodo é applicabile
% A        é la matrice in cui calcoliamo l'autovalore
% n        é la dimensione della matrice A

risp=0;
[L,U,risp]=lucompat(A,n);                    % Calcolo della fattorizzazione LU
if risp==0
  x=diag(eye(n));
  Kmax=input('Introduci il numero massimo di iterate : ');
  err=0;
  while err==0
      if Kmax<=0 | round(Kmax)~=Kmax
        disp(' ');
        disp(' Il valore introdotto non é corretto');
        Kmax=input('Introducilo nuovamente : ');
      else
        err=1;
      end
  end
  for k=0:Kmax
     y=lurisol(L,U,x,n);
     j=0;
     i=1;
     while j==0
         if abs(x(i))==1
            j=i;
         end
         i=i+1;
     end
     l=y(j)/x(j);
     x=y/max(abs(y));
  end
  l=1/l;
end
```

6. Autminsi.m

```
% QUESTO ALGORITMO DETERMINA L'AUTOVALORE DI MODULO PIU' PICCOLO DI UNA MATRICE
% SIMMETRICA

function [l,risp]=autminsi(A,n);
% l         é l'autovalore da calcolare
% risp      controlla se il metodo é applicabile
% A         é la matrice in cui calcoliamo l'autovalore
% n         é la dimensione della matrice A

risp=0;
fc=1;
[L,U,C,risp,r]=lusimm(A,n,fc);                  % fattorizzazione LU per matrici simmetriche
if risp==0
  x=diag(eye(n));
  Kmax=input('Introduci il numero massimo di iterate : ');
  err=0;
  while err==0
      if Kmax<=0 | round(Kmax)~=Kmax
        disp(' ');
        disp(' Il valore introdotto non é corretto');
        Kmax=input('Introducilo nuovamente : ');
      else
        err=1;
      end
  end
  for k=0:Kmax
     y=lurisol(L,U,x,n);
     l=(x'*y')/(norm(x))^2;
     x=y'/norm(y);
  end
  l=1/l;
end
```

7. Auttrsim.m

```
% CALCOLO DI TUTTI GLI AUTOVALORI REALI DELLE MATRICI TRIDIAGONALI SIMMETRICHE

function l=auttrsim(A,n,e);
% l         é l'autovalore da calcolare
% A         é la matrice in cui calcoliamo l'autovalore
% n         é la dimensione della matrice A
% e         é l'errore massimo che si deve commettere

inf1(1)=A(1,1)-abs(A(1,2));
sup1(1)=A(1,1)+abs(A(1,2));
for i=2:n-1                                      % calcolo dell'intervallo contenente
   k=abs(A(i,i-1))+abs(A(i,i+1));               % tutti gli autovalori della matrice
   inf1(i)=A(i,i)-k;
   sup1(i)=A(i,i)+k;
end
inf1(n)=A(n,n)-abs(A(n,n-1));
sup1(n)=A(n,n)+abs(A(n,n-1));
inf1=min(inf1);
sup1=max(sup1);
k=(log((sup1-inf1)/e))/log(2);                   % calcolo del numero di iterate da effettuare
k=ceil(k);
for i=1:n
   inf2=inf1;                                    % determinazione di tutti gli
```

```
    sup2=sup1;                                          % autovalori della matrice
    for j=1:k
        c=(inf2+sup2)/2;
        w=camsegn(A,n,c);
        if w>=i
            sup2=c;
        else
            inf2=c;
        end
    end
    l(i)=(inf2+sup2)/2;
end
```

8. Camsegn.m

```
% CALCOLO DEL NUMERO DI CAMBIAMENTI DI SEGNO DELLA SUCCESSIONE DI STURM

function w=camsegn(A,n,x);
% w     rappresenta il numero di cambiamenti di segno della successione di Sturm calcolata in x
% A     é la matrice triangolare simmetrica per la quale calcoliamo tale successione
% n     é la dimensione della matrice A
% x     punto in cui calcoliamo la successione di Sturm

p(1)=1;
p(2)=A(1,1)-x;
if p(2)==0
   p(2)=1;
end
for i=3:n+1
    p(i)=(A(i-1,i-1)-x)*p(i-1)-((A(i-1,i-2))^2)*p(i-2);
    if p(i)==0
       p(i)=p(i-1);
    end
end
w=0;
for i=2:n+1
    if p(i)*p(i-1)<0
       w=w+1;
    end
end
```

9. Chocomp.m

```
% FATTORIZZAZIONE DI CHOLESKY CON LA TECNICA COMPATTA

function L=chocomp(A,n);
% L     matrice ottenuta dalla fattorizzazione di Cholesky
% A     matrice da fattorizzare
% n     dimensione della matrice

for i=1:n
    for j=1:i-1
        a=0;
        for k=1:j-1
            a=a+L(i,k)*L(j,k);
        end
        L(i,j)=(A(i,j)-a)/L(j,j);
    end
    a=0;
    for k=1:i-1
        a=a+(L(i,k))^2;
    end
    L(i,i)=sqrt(A(i,i)-a);
```

```
end
```

10. Chodool.m

```
% FATTORIZZAZIONE DI CHOLESKY CON IL METODO DI DOOLITTLE

function L=chodool(A,n);
% L     matrice ottenuta dalla fattorizzazione di Cholesky
% A     matrice da fattorizzare
% n     dimensione della matrice

for j=1:n
   a=0;
   for k=1:j-1
      a=a+(L(j,k))^2;
   end
   L(j,j)=sqrt(A(j,j)-a);
   for i=j+1:n
      a=0;
      for k=1:j-1
         a=a+L(i,k)*L(j,k);
      end
      L(i,j)=(A(i,j)-a)/L(j,j);
   end
end
```

11. Eulespl.m

```
% METODO DI EULERO ESPLICITO PER LA RISOLUZIONE DELLE EQUAZIONI DIFFERENZIALI

function y=eulespl(F,t,y,n,h);
% y     vettore soluzione dell'equazione differenziale
% F     é la funzione a due variabili F(t,y)
% n     é il numero di suddivisioni dell'intervallo [a,b] fissato
% h     é l'ampiezza delle n suddivisioni

for i=2:n+1
   y(i)=y(i-1)+h*feval(F,t(i-1),y(i-1));
end
```

12. Eulimpl.m

```
% METODO DI EULERO IMPLICITO PER LA RISOLUZIONE DELLE EQUAZIONI DIFFERENZIALI
function [y,risp]=eulimpl(F,G,t,y,n,h,e);
% y       vettore soluzione dell'equazione differenziale
% risp    controlla se il metodo é applicabile
% F       é la funzione a due variabili F(t,y)
% G       é la derivata di F
% t       é la variabile indipendente della funzione y
% n       é il numero di suddivisioni dell'intervallo [a,b] fissato
% h       é l'ampiezza delle n suddivisioni
% e       é l'errore massimo che si deve commettere

risp=0;
i=2;
while i<=n+1 & risp==0
   sol1=y(i-1);
   wd=feval(G,t(i),sol1);
   if wd==1/h
      risp=1;
   else
      wf=sol1-y(i-1)-h*(feval(F,t(i),sol1));
      sol2=sol1-wf/(1-h*wd);
```

```
    end
    while risp==0 & (abs(sol2-sol1))>=e | (abs(wf)>=e)
      sol1=sol2;
      wd=feval(G,t(i),sol1);
      if wd==1/h
        risp=1;
      else
        wf=sol1-y(i-1)-h*feval(F,t(i),sol1);
        sol2=sol1-wf/(1-h*wd);
      end
    end
    y(i)=sol2;
    i=i+1;
end
```

13. Fattqr.m

```
% FATTORIZZAZIONE QR

function [Q,R,risp]=fattqr(A,m,n);
% Q ,R    sono le matrici ottenute dalla fattorizzazione
% A       matrice da fattorizzare
% n ,m    dimensioni della matrice

risp=0;
R=A;
Q=eye(m);
if risp==0
 for k=1:n
    c=sign(R(k,k));
    if c==0
      c=1;
    end
    a=0;                                % calcolo della norma del vettore R(k:m,k)
    for i=k:m
       a=a+(R(i,k))^2;
    end
    a=sqrt(a);                          % calcolo delle matrici Q(k) e R(k)
    b=a*(abs(R(k,k))+a);
    if b==0
      risp=1;
    else
      b=1/b;
      u=1;
      u(1)=R(k,k)+c*a;
      u(2:m-k+1)=R(k+1:m,k);
      P=eye(m-k+1,m-k+1)-b*u'*u;
      S=0;
      if k>1
        S(1:k-1,1:k-1)=eye(k-1,k-1);
      end
      S(k:m,k:m)=P;
      R(k:m,k:n)=P*R(k:m,k:n);
      Q=S*Q;
    end
 end
else
 Q=0;
 R=0;
end
Q=Q';
```

14. Gauss.m

```
   % METODO DI GAUSS

function [x,risp]=gauss(A,b,n);
% x       vettore soluzione del sistema
% risp    controlla se il metodo é applicabile
% A       vettore dei coefficienti del sistema
% b       vettore dei termini noti
% n       dimensione di A

risp=0;
if b==zeros(n,1);
  x=zeros(n,1);
else
  k=1;
  while k<n
    if A(k,k)==0
      risp=1;
      k=n;
    end
    if risp==0
      for i=k+1:n
        for j=k+1:n
          L=A(i,k)/A(k,k);
          A(i,j)=A(i,j)-L*A(k,j);
        end
        b(i)=b(i)-L*b(k);
        for j=1:k
          A(i,j)=0;
        end
      end
    end
    k=k+1;
  end

% APPLICAZIONE DEL METODO DI GAUSS PER LA RICERCA DEL VETTORE X:

  if risp==0
    x(n)=b(n)/A(n,n);
    for i=n-1:-1:1
      a=0;
      for j=i+1:n
        a=a+A(i,j)*x(j);
      end
      x(i)=(b(i)-a)/A(i,i);
    end
  end
end
if risp==1
  x=0;
end
```

15. Intherm.m

```
% INTERPOLAZIONE DI HERMITE

function P=intherm(x,y,z,xi,n,m);
% P    vettore soluzione dei punti da interpolare
% x    vettore dei nodi interpolanti
% y    valori della funzione nei nodi x
% z    valori della derivata prima della funzione nei nodi x
% xi   vettore dei punti in cui calcolare la funzione interpolante
% n    dimensione di x , y e z
% m    dimensione di xi

for ki=1:m
   p=0;
   for k=1:n
      a=1;
      c=1;
      for i=1:k-1
         c=c*(x(k)-x(i));
         a=a*(xi(ki)-x(i));
      end
      for i=k+1:n
         c=c*(x(k)-x(i));
         a=a*(xi(ki)-x(i));
      end
      a=a/c;

%   Calcolo della derivata prima dei polinomi fondamentali

      l=0;
      for j=1:n
         if j~=k
            d=1;
            for i=1:n
               if i~=k
                  if i~=j
                     d=d*(x(k)-x(i));
                  end
               end
            end
            l=l+d;
         end
      end
      l=l/c;
      p=p+a*a*(y(k)+(z(k)-2*l*y(k))*(xi(ki)-x(k)));
   end
   P(ki)=p;
end
P=P';
```

16. Intlagr.m

```
% INTERPOLAZIONE DI LAGRANGE

function L=intlagr(x,y,xi,n,m);
% L     vettore soluzione dei punti da interpolare
% x     vettore dei nodi interpolanti
% y     valori della funzione nei nodi x
% xi    vettore dei punti in cui calcolare la funzione interpolante
% n     dimensione di x , y
% m     dimensione di xi

q=1;                                      % ordinamento dei nodi
while q==1
    q=0;
    for i=1:n-1
       if x(i)>x(i+1)
          u=x(i);
          x(i)=x(i+1);
          x(i+1)=u;
          u=y(i);
          y(i)=y(i+1);
          y(i+1)=u;
          q=1;
       end
    end
end
for i=1:m                                 % interpolazione di Lagrange
   L(i)=0;
   for k=1:n
      a=1;
      for j=1:k-1
         a=a*((xi(i)-x(j))/(x(k)-x(j)));
      end
      for j=k+1:n
         a=a*((xi(i)-x(j))/(x(k)-x(j)));
      end
      L(i)=L(i)+a*y(k);
   end
end
L=L';
asc=x(1):.1:x(n);                         % Algoritmo seguente utilizzato
[h,m2]=size(asc);                         % unicamente per tracciare il grafico
for i=1:m2
   ord(i)=0;
   for k=1:n
      a=1;
      for j=1:k-1
         a=a*((asc(i)-x(j))/(x(k)-x(j)));
      end
      for j=k+1:n
         a=a*((asc(i)-x(j))/(x(k)-x(j)));
      end
      ord(i)=ord(i)+a*y(k);
   end
end
plot(asc,ord,'w',xi,L,'+r',x,y,'og');
title('Interpolazione di Lagrange');
```

17. Intminqu.m

```
% APPROSSIMAZIONE POLINOMIALE AI MINIMI QUADRATI

function [p,risp]=intminqu(x,y,xi,n,m,w);
% p       vettore soluzione dei punti da interpolare
% risp    controlla se il metodo é applicabile
% x       vettore dei nodi interpolanti
% y       valori della funzione nei nodi x
% xi      vettore dei punti in cui calcolare la funzione interpolante
% n       dimensione di x , y
% w       dimensione di xi
% m       grado del polinomio interpolante

B=diag(eye(n));                          % determinazione della matrice
for i=1:n                                % dei coefficienti del sistema
  for j=2:m                              % da risolvere
     B(i,j)=(x(i))^(j-1);
  end
end
[a,f,risp]=minquad(B,y,n,m);
if risp==0
a=a';
for i=1:w                                % regola di Horner per valutare
   p(i)=0;                               % il polinomio interpolante
   p(i)=valpol(xi(i),a);                 % nei nodi
end
asc=x(1):.1:x(n);                        % Algoritmo seguente utilizzato
[h,m2]=size(asc);                        % unicamente per tracciare il grafico
for i=1:m2
   ord(i)=0;
   for j=m:-1:1
      ord(i)=ord(i)*asc(i)+a(j);
   end
end
plot(asc,ord,'w',xi,p,'+r',x,y,'og');
title('Approssimazione polinomiale');
else
p=0;
end
```

18. Intnevil.m

```
% INTERPOLAZIONE CON SCHEMA DI NEVILLE

function z=intnevil(x,y,xi,n,m);
% z     vettore soluzione dei punti da interpolare
% x     vettore dei nodi interpolanti
% y     valori della funzione nei nodi x
% xi    vettore dei punti in cui calcolare la funzione interpolante
% n     dimensione di x , y
% m     dimensione di xi

for k=1:m
   L=0;
   for i=1:n
      L(i,1)=y(i);
   end
   for j=2:n
      for i=j:n
```

```
            L(i,j)=(L(i,j-1)*(xi(k)-x(i-j+1))-L(i-1,j-1)*(xi(k)-x(i)))/(x(i)-x(i-j+1));
        end
    end
    z(k)=L(n,n);
end
asc=x(1):.1:x(n);                              % Algoritmo seguente utilizzato
[h,m2]=size(asc);                              % unicamente per tracciare il grafico
for k=1:m2
    L=0;
    for i=1:n
        L(i,1)=y(i);
    end
    for j=2:n
        for i=j:n
            L(i,j)=(L(i,j-1)*(asc(k)-x(i-j+1))-L(i-1,j-1)*(asc(k)-x(i)))/(x(i)-x(i-j+1));
        end
    end
    ord(k)=L(n,n);
end
plot(asc,ord,'w',xi,z,'+r',x,y,'og');
title('Interpolazione con schema di Neville');
```

19. Intnewt.m

```
% INTERPOLAZIONE DI NEWTON

function L=intnewt(x,y,xi,n,m);
% L     vettore soluzione dei punti da interpolare
% x     vettore dei nodi interpolanti
% y     valori della funzione nei nodi x
% xi    vettore dei punti in cui calcolare la funzione interpolante
% n     dimensione di x , y
% m     dimensione di xi

for i=1:n
    F(i,1)=y(i);
end
for k=1:m
    for j=2:n
        for i=j:n
            F(i,j)=(F(i,j-1)-F(i-1,j-1))/(x(i)-x(i-j+1));
        end
    end
    L(k)=F(1,1);
    w=1;
    for i=2:n
        w=w*(xi(k)-x(i-1));
        L(k)=L(k)+F(i,i)*w;
    end
end
asc=x(1):.1:x(n);                              % Algoritmo seguente utilizzato
[h,m2]=size(asc);                              % unicamente per tracciare il grafico
for k=1:m2
    for j=2:n
        for i=j:n
            F(i,j)=(F(i,j-1)-F(i-1,j-1))/(x(i)-x(i-j+1));
        end
    end
    ord(k)=F(1,1);
    w=1;
```

```
   for i=2:n
      w=w*(asc(k)-x(i-1));
      ord(k)=ord(k)+F(i,i)*w;
   end
end
plot(asc,ord,'w',xi,L,'+r',x,y,'og');
title('Interpolazione di Newton');
```

20. Jacobi.m

```
% METODO DI JACOBI PER LA RISOLUZIONE DI SISTEMI LINEARI

function [x,risp]=jacobi(A,b,n);
% x       vettore soluzione del sistema
% risp    controlla se il metodo é applicabile
% A       vettore dei coefficienti del sistema
% b       vettore dei termini noti
% n       dimensione di A

risp=0;
for i=1:n
   D(i,i)=1/A(i,i);
end
disp('Per una verifica molto precisa sulla convergenza del metodo iterativo');
disp('adottato introdurre un valore abbastanza elevato del seguente numero');
D=eye(n)-D*A;
l=autmax(D,n);
if abs(l)>=1
  risp=1;
  x=0;
else
  Kmax=input('Introduci il numero massimo di iterate per applicare il metodo di Jacobi :');
  err=0;
  while err==0
      if Kmax<=0 | round(Kmax)~=Kmax
        disp(' ');
        disp(' Il valore introdotto non é corretto');
        Kmax=input('Introducilo nuovamente : ');
      else
        err=1;
      end
  end
  x=diag(eye(n));
  for k=1:Kmax
     for i=1:n
        a=0;
        for j=1:i-1
           a=a-A(i,j)*x(j);
        end
        for j=i+1:n
           a=a-A(i,j)*x(j);
        end
        x(i)=(b(i)+a)/A(i,i);
     end
  end
end
```

21. Lucompat.m

```
% FATTORIZZAZIONE LU CON TECNICA COMPATTA

function [L,U,risp]=lucompat(A,n);
% L U      matrici ottenute dalla fattorizzazione LU
% risp     controlla se il metodo é applicabile
% A        matrice da fattorizzare
% n        dimensione della matrice

risp=0;
U=A(1,:);
L=eye(n,n);
i=2;
while i<=n
   for j=1:i-1
      if U(j,j)==0
         risp=1;
         j=n+1;
         i=n+1;
      end
      if risp==0
         a=0;
         for k=1:j-1
            a=a+L(i,k)*U(k,j);
         end
         L(i,j)=(1/U(j,j))*(A(i,j)-a);
      end
   end
   for j=i:n
      a=0;
      for k=1:i-1
         a=a+L(i,k)*U(k,j);
      end
      U(i,j)=A(i,j)-a;
   end
   i=i+1;
end
```

22. Ludimost.m

```
% FATTORIZZAZIONE LU
% UTILIZZANDO LA SEGUENZA DI PASSI DEL TEOREMA DI ESISTENZA ED UNICITA' DELLA STESSA

function [L,U,risp]=ludimost(A,n);
% L,U   matrici ottenute dalla fattorizzazione LU
% risp  controlla se il metodo é applicabile
% A     matrice da fattorizzare
% n     dimensione della matrice

risp=0;
L(1,1)=1;
U(1,1)=A(1,1);
k=2;
while k<=n
   if U(k-1,k-1)==0
      risp=1;
      k=n+1;
   end
   if risp==0
```

```
      L(k,1)=A(k,1)/U(1,1);                     % RISOLUZIONE DEL SISTEMA l'* Un-1 = r'
      for i=2:k-1
        a=0;
        for j=1:i-1
          a=a+U(j,i)*L(k,j);
        end
        L(k,i)=(A(k,i)-a)/U(i,i);
      end
      L(k,k)=1;                                 % RISOLUZIONE DEL SISTEMA c = Ln-1 * u :
      U(1,k)=A(1,k)/L(1,1);
      for i=2:k-1
        a=0;
        for j=1:i-1
          a=a+L(i,j)*U(j,k);
        end
        U(i,k)=(A(i,k)-a)/L(i,i);
      end
      U(k,k)=A(k,k)-L(k,1:k-1)*U(1:k-1,k);
    end
    k=k+1;
end
if risp==1
   L=0;
   U=0;
end
```

23. Ludoolit.m

```
% FATTORIZZAZIONE LU CON METODO DI DOOLITTLE

function [L,U,risp]=ludoolit(A,n);
% L U   matrici ottenute dalla fattorizzazione LU
% risp  controlla se il metodo é applicabile
% A     matrice da fattorizzare
% n     dimensione della matrice

risp=0;
U=A(1,:);
L=eye(n,n);
j=1;
while j<n
    if U(j,j)==0
      risp=1;
      j=n;
    end
    if risp==0
     for i=j+1:n
       a=0;
       for k=1:j-1
         a=a+L(i,k)*U(k,j);
       end
       L(i,j)=(1/U(j,j))*(A(i,j)-a);
     end
     for i=2:j+1
       a=0;
       for k=1:i-1
         a=a+L(i,k)*U(k,j+1);
       end
       U(i,j+1)=A(i,j+1)-a;
     end
```

```
    end
    j=j+1;
end
if risp==1
   L=0;
   U=0;
end
```

24. Lugauss.m

```
% FATTORIZZAZIONE LU UTILIZZANDO IL METODO DI ELIMINAZIONE DI GAUSS

function [L,U,risp]=lugauss(A,n);
% L U      matrici ottenute dalla fattorizzazione LU
% risp     controlla se il metodo é applicabile
% A        matrice da fattorizzare
% n        dimensione della matrice

risp=0;
L=eye(n,n);
U=A;
k=1;
while k<n
    if U(k,k)==0
       risp=1;
       k=n;
    end
    if risp==0
       for i=k+1:n
          for j=k+1:n
             L(i,k)=U(i,k)/U(k,k);
             U(i,j)=U(i,j)-L(i,k)*U(k,j);
          end
          for j=1:k
             U(i,j)=0;
          end
       end
    end
    k=k+1;
end
if risp==1
   x=0;
end
```

25. Lupivpar.m

```
% RISOLUZIONE DI SISTEMI LINEARI CON STRATEGIA DI PIVOTING PARZIALE

function x=lupivpar(A,b,n);
% x     vettore soluzione del sistema
% A     vettore dei coefficienti del sistema
% b     vettore dei termini noti
% n     dimensione di A

for k=1:n-1
    max=0;
    j=0;
    for i=k:n                              % ricerca dell'elemento
       if abs(A(i,k))>max                  % di massimo modulo sulla
          max=abs(A(i,k));                 % prima colonna del minore
```

```
        j=i;                                          % A(k,k)
      end
    end
    if k~=j                                           % scambio di righe
      u=A(k,:);
      A(k,:)=A(j,:);
      A(j,:)=u;
      u=0;
      u=b(k);
      b(k)=b(j);
      b(j)=u;
    end
    for i=k+1:n
      for j=k+1:n
        L=A(i,k)/A(k,k);
        A(i,j)=A(i,j)-L*A(k,j);
      end
      b(i)=b(i)-L*b(k);
    end
end

% APPLICAZIONE DEL METODO DI GAUSS PER LA RICERCA DEL VETTORE X:

x(n)=b(n)/A(n,n);
for i=n-1:-1:1
    a=0;
    for j=i+1:n
      a=a+A(i,j)*x(j);
    end
    x(i)=(b(i)-a)/A(i,i);
end
```

26. Lupivtot.m

```
% RISOLUZIONE DI SISTEMI LINEARI CON STRATEGIA DI PIVOTING TOTALE

function x=lupivtot(A,b,n);
% x       vettore soluzione del sistema
% risp    controlla se il metodo é applicabile
% A       vettore dei coefficienti del sistema
% b       vettore dei termini noti
% n       dimensione di A

for k=1:n-1
    max=0;                                            % Ricerca dell'elemento massimo
    for i=k:n                                         % della matrice A(k,k)
      for j=k:n
        if abs(A(i,k))>max
          max=abs(A(i,k));
          y=i;
          z(k)=j;
        end
      end
    end
    if k~=y                                           % Scambio di righe
      u=A(k,:);
      A(k,:)=A(y,:);
      A(y,:)=u;
      u=0;
      u=b(k);
```

```
      b(k)=b(y);
      b(y)=u;
   end
   if k~=z(k)                                  % Scambio di colonne
      u=0;
      u=A(:,k);
      A(:,k)=A(:,z(k));
      A(:,z(k))=u;
   end
   for i=k+1:n
      for j=k+1:n
         L=A(i,k)/A(k,k);
         A(i,j)=A(i,j)-L*A(k,j);
      end
      b(i)=b(i)-L*b(k);
   end
end

% APPLICAZIONE DEL METODO DI GAUSS PER LA RICERCA DEL VETTORE X:

x(n)=b(n)/A(n,n);
for i=n-1:-1:1
   a=0;
   for j=i+1:n
      a=a+A(i,j)*x(j);
   end
   x(i)=(b(i)-a)/A(i,i);
end
u=0;

for i=1:n-1                                    % ordinamento del vettore soluzione
   if i~=z(i)
      u=x(i)
      x(i)=x(z(i));
      x(z(i))=u;
   end
end
```

27. Lurischo.m

```
% RISOLUZIONE DEI SISTEMI LINEARI IN CUI E' STATA CALCOLATA LA FATTORIZZAZIONE DI
% CHOLESKY

function x=lurischo(L,b,n);
% x     vettore soluzione del sistema
% L     vettore dei coefficienti del sistema
% b     vettore dei termini noti
% n     dimensione di A

U=L';
y(1)=b(1)/L(1,1);                              % RISOLUZIONE DEL SISTEMA L * y = b
for i=2:n
   a=0;
   for j=1:i-1
      a=a+L(i,j)*y(j);
   end
   y(i)=(b(i)-a)/L(i,i);
end
x(n)=y(n)/U(n,n);                              % RISOLUZIONE DEL SISTEMA U * x = y
for i=n-1:-1:1
```

```
   a=0;
   for j=i+1:n
      a=a+U(i,j)*x(j);
   end
   x(i)=(y(i)-a)/U(i,i);
end
```

28. Lurisol.m

```
% RISOLUZIONE DEI SISTEMI LINEARI IN CUI E' STATA CALCOLATA LA FATTORIZZAZIONE LU

function x=lurisol(L,U,b,n);
% x       vettore soluzione del sistema
% L,U     fattorizzazione LU della matrice dei coefficienti del sistema
% b       vettore dei termini noti
% n       dimensione di A

y(1)=b(1);                              % RISOLUZIONE DEL SISTEMA L * y = b
for i=2:n
   a=0;
   for j=1:i-1
      a=a+L(i,j)*y(j);
   end
   y(i)=b(i)-a;
end
x(n)=y(n)/U(n,n);                       % RISOLUZIONE DEL SISTEMA U * x = y
for i=n-1:-1:1
   a=0;
   for j=i+1:n
      a=a+U(i,j)*x(j);
   end
   x(i)=(y(i)-a)/U(i,i);
end
```

29. Lusimm.m

```
% FATTORIZZAZIONE LU PER LE MATRICI SIMMETRICHE

function [L,U,C,risp,r]=lusimm(A,n,fc);
% L,U      matrici ottenute dalla fattorizzazione LU
% C        matrice ottenuta dalla fattorizzazione di Cholesky
% risp     controlla se il metodo é applicabile
% r        indica se si vuole o meno calcolare la fattorizzazione di Cholesky
% A        matrice da fattorizzare
% n        dimensione della matrice
% fc       considera o meno l'intero algoritmo del calcolo della fattorizzazione di Cholesky

risp=0;
C=0;
U=triu(A);
L=eye(n);
p=0;
k=1;
while k<n
   if U(k,k)==0
      risp=1;
      k=n;
   else
      if U(k,k)<0
         p=1;
      end
      for i=k+1:n
```

```
          L(i,k)=U(k,i)/U(k,k);
          for j=i:n
             U(i,j)=U(i,j)-L(i,k)*U(k,j);
          end
       end
    end
    k=k+1;
end
if U(n,n)<0
   p=1;
end
r=0;
if p==1 & risp==0 & fc==0
   disp('La matrice non é anche definita positivamente');
   disp('per cui non é fattorizzabile secondo CHOLESKI');
   r=0;
end
if p==0 & risp==0 & fc==0
     disp('La matrice puo" essere fattorizzabile secondo CHOLESKY');
     r=input('vuoi calcolarla? (se SI premere " 1 ") : ');
end
if r==1
   for k=1:n
      D(k,k)=sqrt(U(k,k));
   end
   C=L*D;
else
   C=0;
end
```

30. Lutrid.m

```
% FATTORIZZAZIONE LU PER MATRICI TRIDIAGONALI

function [L,U,risp]=lutrid(A,n);
% L,U     matrici ottenute dalla fattorizzazione LU
% risp    controlla se il metodo é applicabile
% A       matrice da fattorizzare
% n       dimensione della matrice

risp=0;
U(1,1)=A(1,1);
L(1,1)=1;
i=2;
while i<=n
    if U(i-1,i-1)==0
       risp=1;
       i=n+1;
    end
    if risp==0
       L(i,i)=1;
       L(i,i-1)=A(i,i-1)/U(i-1,i-1);
       U(i-1,i)=A(i-1,i);
       U(i,i)=A(i,i)-L(i,i-1)*U(i-1,i);
    end
    i=i+1;
end
if risp==1
   L=0;
   U=0;
end
```

31. Meripdis.m

```
% METODO DI PIU' RIPIDA DISCESA PER LA RISOLUZIONE DI SISTEMI LINEARI

function x=meripdis(A,b,n,e);
% x      vettore soluzione del sistema
% A      vettore dei coefficienti del sistema
% b      vettore dei termini noti
% n      dimensione di A
% e       é l'errore massimo che si deve commettere

for i=1:n
   somma=0;
   for j=1:n                                   % determinazione di un intervallo contenente
      if j~=i                                   % tutti gli autovalori della matrice
         somma=somma+abs(A(i,j));
      end
   end
   p1(i)=A(i,i)-somma;
   p2(i)=A(i,i)+somma;
end
m=min(p1);
M=max(p2);
k=log(e*sqrt(m/M))/log((M-m)/(M+m));
x=diag(eye(n));
for ki=1:k
   r=b-A*x;
   x=x+(((norm(r))^2)/(r'*A*r))*r;
end
x=x';
```

32. Mettrap.m

```
% METODO DEI TRAPEZI PER LA RISOLUZIONE DELLE EQUAZIONI DIFFERENZIALI

function [y,risp]=mettrap(F,G,t,y,n,h,e);
% y         vettore soluzione dell'equazione differenziale
% risp      controlla se il metodo é applicabile
% F         é la funzione a due variabili F(t,y)
% G         é la derivata di F
% t         é la variabile indipendente della funzione y
% n         é il numero di suddivisioni dell'intervallo [a,b] fissato
% h         é l'ampiezza delle n suddivisioni
% e         é l'errore massimo che si deve commettere

risp=0;
wf1=feval(F,t(1),y(1));
i=2;
while i<=n+1 & risp==0
   sol1=y(i-1);
   wd=feval(G,t(i),sol1);
   if wd==2/h
     risp=1;
   else
     wf2=feval(F,t(i),sol1);
     wf3=sol1-y(i-1)-(h/2)*(wf1+wf2);
     sol2=sol1-(wf3/(1-(h/2)*wd));
     while risp==0 & (abs(sol2-sol1))>=e | (abs(wf3)>=e)
        sol1=sol2;
        wf2=feval(F,t(i),sol1);
```

```
        wf3=sol1-y(i-1)-(h/2)*(wf1+wf2);
        sol2=sol1-(wf3/(1-(h/2)*wd));
      end
      wf1=wf2;
      y(i)=sol2;
   end
   i=i+1;
end
```

33. Minquad.m

```
% PROBLEMA LINEARE AI MINIMI QUADRATI

function [x,a,risp]=minquad(A,b,m,n);
% x       vettore soluzione del sistema
% a       norma 2 del residuo
% risp    controlla se il metodo é applicabile
% A       matrice dei coefficienti del sistema
% b       vettore dei termini noti
% m,n     dimensioni di A

[Q,R,risp]=fattqr(A,m,n);                    % Fattorizzazione QR
if risp==0
   c=Q'*b;
   x(n)=c(n)/R(n,n);                         % Risoluzione del sistema R * x = c1 :
   for i=n-1:-1:1
      a=0;
      for j=i+1:n
         a=a+R(i,j)*x(j);
      end
      x(i)=(c(i)-a)/R(i,i);
   end
   a=0;
   for i=n+1:m                               % Modulo del vettore residuo
      a=a+(c(i))^2;
   end
   a=sqrt(a);
else
   a=0;
   x=0;
end
```

34. Splcubco.m

```
% SPLINE CUBICA COMPLETA INTERPOLANTE

function [s,risp]=splcubco(x,y,xi,n,m1,k0,k1);
% s       vettore soluzione dei punti da interpolare
% risp    controlla se il metodo é applicabile
% x       vettore dei nodi interpolanti
% y       valori della funzione nei nodi x
% xi      vettore dei punti in cui calcolare la funzione interpolante
% n       dimensione di x , y
% m1      dimensione di xi
% k0      valore della derivata prima nel nodo più piccolo
% k0      valore della derivata prima nel nodo più grande

% COSTRUZIONE DELLA MATRICE TRIDIAGONALE

h(1)=x(2)-x(1);
```

```
A(1,1)=h(1)/3;
A(1,2)=h(1)/6;
A(2,1)=h(1);
b(1)=(y(2)-y(1))/h(1)-k0;
for i=1:n-2
   h(i+1)=x(i+2)-x(i+1);
   A(i+1,i+1)=2*(h(i)+h(i+1));
   A(i+1,i+2)=h(i+1);
   if i<n-2
      A(i+2,i+1)=A(i+1,i+2);
   end
   b(i+1)=6*((y(i+2)-y(i+1))/h(i+1)+(y(i)-y(i+1))/h(i));
end
A(n,n-1)=h(n-1)/6;
A(n,n)=h(n-1)/3;
b(n)=k1-(y(n)-y(n-1))/h(n-1);
[L,U,risp]=lutrid(A,n);                    % Fattorizzazione LU per matrici tridiagonali
if risp==0

% RISOLUZIONE DEL SISTEMA L * U * m = b

z(1)=b(1);                                 % RISOLUZIONE DEL SISTEMA L * z = b
for i=2:n
   z(i)=b(i)-L(i,i-1)*z(i-1);
end

m(n)=z(n)/U(n,n);                          % RISOLUZIONE DEL SISTEMA U * m = z
for i=n-1:-1:1
   m(i)=(z(i)-U(i,i+1)*m(i+1))/U(i,i);
end

for k=1:m1
   if xi(k)<=x(2) & xi(k)>=x(1)
      b=xi(k)-x(1);
      s(k)=y(1)+k0*b+(m(1)*(b^2))/2+(((m(2)-m(1))/h(1))*(b^3))/6;
   end
   for i=2:n-1
      if xi(k)<=x(i+1) & xi(k)>=x(i)
         c=(y(i+1)-y(i))/h(i)-m(i)*(h(i)/3)-m(i+1)*(h(i)/6);
         b=xi(k)-x(i);
         s(k)=y(i)+c*b+(m(i)*(b^2))/2+(((m(i+1)-m(i))/h(i))*(b^3))/6;
      end
   end
end
else
  s=0;
end
s=s';

if risp==0                                 % Algoritmo seguente utilizzato
  asc=x(1):.1:x(n);                        % unicamente per tracciare il grafico
  asc=asc';
  [m2,h1]=size(asc);
  for k=1:m2
   if asc(k)<=x(2) & asc(k)>=x(1)
      b=asc(k)-x(1);
      ord(k)=y(1)+k0*b+(m(1)*(b^2))/2+(((m(2)-m(1))/h(1))*(b^3))/6;
   end
   for i=2:n-1
      if asc(k)<=x(i+1) & asc(k)>=x(i)
```

```
          c=(y(i+1)-y(i))/h(i)-m(i)*(h(i)/3)-m(i+1)*(h(i)/6);
          b=asc(k)-x(i);
          ord(k)=y(i)+c*b+(m(i)*(b^2))/2+(((m(i+1)-m(i))/h(i))*(b^3))/6;
        end
    end
  end
  ord=ord';
  plot(asc,ord,'w',xi,s,'+r',x,y,'og');
  title('Spline cubica completa');
else
  disp('il grafico non é stato possibile disegnarlo');
end
```

35. Splcubna.m

```
% SPLINE CUBICA NATURALE INTERPOLANTE

function [s,risp]=splcub(x,y,xi,n,m1);
% s        vettore soluzione dei punti da interpolare
% risp     controlla se il metodo é applicabile
% x        vettore dei nodi interpolanti
% y        valori della funzione nei nodi x
% xi       vettore dei punti in cui calcolare la funzione interpolante
% n        dimensione di x , y
% m1       dimensione di xi

% COSTRUZIONE DELLA MATRICE TRIDIAGONALE

h(1)=x(2)-x(1);
h(2)=x(3)-x(2);
A(1,1)=2*(h(2)+h(1));
A(1,2)=h(2);
A(2,1)=A(1,2);
b(1)=6*((y(3)-y(2))/h(2)+(y(1)-y(2))/h(1));
for i=2:n-3
   h(i+1)=x(i+2)-x(i+1);
   A(i,i)=2*(h(i)+h(i+1));
   A(i,i+1)=h(i+1);
   A(i+1,i)=A(i,i+1);
   b(i)=6*((y(i+2)-y(i+1))/h(i+1)+(y(i)-y(i+1))/h(i));
end
h(n-1)=x(n)-x(n-1);
A(n-2,n-2)=2*(h(n-1)+h(n-2));
b(n-2)=6*((y(n)-y(n-1))/h(n-1)+(y(n-2)-y(n-1))/h(n-2));
[L,U,risp]=lutrid(A,n-2);                    % Fattorizzazione LU per matrici tridiagonali
if risp==0

% RISOLUZIONE DEL SISTEMA L * U * m = b

z(1)=b(1);                                   % RISOLUZIONE DEL SISTEMA L * z = b
for i=2:n-2
   z(i)=b(i)-L(i,i-1)*z(i-1);
end

m(n-2)=z(n-2)/U(n-2,n-2);                    % RISOLUZIONE DEL SISTEMA U * m = z
for i=n-3:-1:1
   m(i)=(z(i)-U(i,i+1)*m(i+1))/U(i,i);
end

for k=1:m1
```

```
    if xi(k)<=x(2) & xi(k)>=x(1)
      c=(y(2)-y(1))/h(1)-(h(1)*m(1))/6;
      s(k)=y(1)+c*(xi(k)-x(1))+((m(1)/h(1))*(xi(k)-x(1))^3)/6;
    end
    for i=2:n-2
      if xi(k)<=x(i+1) & xi(k)>=x(i)
        c=(y(i+1)-y(i))/h(i)-m(i-1)*(h(i)/3)-m(i)*(h(i)/6);
        b=xi(k)-x(i);
        s(k)=y(i)+c*b+(m(i-1)*(b^2))/2+(((m(i)-m(i-1))/h(i))*(b^3))/6;
      end
    end
    if xi(k)<=x(n) & xi(k)>=x(n-1)
      c=(y(n)-y(n-1))/h(n-1)-m(n-2)*(h(n-1)/3);
      s(k)=y(n-1)+c*(xi(k)-x(n-1))+(m(n-2)*(xi(k)-x(n-1))^2)/2-((m(n-2)/h(n-1))*(xi(k)-x(n-1))^3)/6;
    end
end
else
  s=0;
end
s=s';
if risp==0                                    % Algoritmo seguente utilizzato
  asc=x(1):.1:x(n);                           % unicamente per tracciare il grafico
  asc=asc';
  [m2,h1]=size(asc);
  for k=1:m2
    if asc(k)<=x(2) & asc(k)>=x(1)
      c=(y(2)-y(1))/h(1)-(h(1)*m(1))/6;
      ord(k)=y(1)+c*(asc(k)-x(1))+((m(1)/h(1))*(asc(k)-x(1))^3)/6;
    end
    for i=2:n-2
      if asc(k)<=x(i+1) & asc(k)>=x(i)
        c=(y(i+1)-y(i))/h(i)-m(i-1)*(h(i)/3)-m(i)*(h(i)/6);
        b=asc(k)-x(i);
        ord(k)=y(i)+c*b+(m(i-1)*(b^2))/2+(((m(i)-m(i-1))/h(i))*(b^3))/6;
      end
    end
    if asc(k)<=x(n) & asc(k)>=x(n-1)
      c=(y(n)-y(n-1))/h(n-1)-m(n-2)*(h(n-1)/3);
      ord(k)=y(n-1)+c*(asc(k)-x(n-1))+(m(n-2)*(asc(k)-x(n-1))^2)/2-((m(n-2)/h(n-1))*(asc(k)-x(n-1))^3)/6;
    end
  end
  ord=ord';
  plot(asc,ord,'w',xi,s,'+r',x,y,'og');
  title('Spline cubica naturale');
else
  disp('il grafico non é stato possibile disegnarlo');
end
```

36. Spllin.m

```
% CALCOLO DELLA SPLINE LINEARE INTERPOLANTE

function s=spllin(x,y,xi,n,m);
% s     vettore soluzione dei punti da interpolare
% x     vettore dei nodi interpolanti
% y     valori della funzione nei nodi x
% xi    vettore dei punti in cui calcolare la funzione interpolante
% n     dimensione di x , y
% m     dimensione di xi
```

```
for k=1:m
   s(k)=0;
   if xi(k)<=x(2) & xi(k)>=x(1)
      s(k)=s(k)+((xi(k)-x(2))/(x(1)-x(2)))*y(1);
   end
   for i=2:n-1
      if xi(k)>=x(i-1) & xi(k)<x(i)
         s(k)=s(k)+((xi(k)-x(i-1))/(x(i)-x(i-1)))*y(i);
      end
      if xi(k)>=x(i) & xi(k)<=x(i+1)
         s(k)=s(k)+((xi(k)-x(i+1))/(x(i)-x(i+1)))*y(i);
      end
   end
   if xi(k)>=x(n-1) & xi(k)<=x(n)
      s(k)=s(k)+((xi(k)-x(n-1))/(x(n)-x(n-1)))*y(n);
   end
end
s=s';
plot(x,y,'w',xi,s,'+r',x,y,'og');
title('Spline lineare passante per i nodi immessi');
```

37. Valpol.m

```
% REGOLA DI HORNER PER CALCOLARE IL VALORE DEL POLINOMIO IN UN PUNTO
% per il calcolo del valore del polinomio in un punto conoscendo i suoi coefficienti

function p=valpol(x,a);
% p    valore del polinomio nel punto x
% a    coefficienti del polinomio

[n,m]=size(a);
if n==1
  n=m;
end
p=0;
for i=n:-1:1
   p=p*x+a(i);
end
```

38. Zerpol.m

```
% CALCOLO DEGLI ZERI DI UN POLINOMIO
function [a,risp,dom]=zerpol(v1,num,err);
% a        vettore degli zeri del polinomio
% risp     controlla se il metodo é applicabile
% dom      tiene conto del fatto che si é risolto una equazione di secondo grado
% v1       coefficienti del polinomio
% num      numero di zeri reali del polinomio
% err      é l'errore massimo che si deve commettere

risp=0;
dom=0;
v3=v1;
[m,n]=size(v3);
n=m;
j1=0;
while j1<m
    if v3(1)==0
       v3=v3(2:n);                          % ricerca di eventuali zeri nulli
       a(j1+1)=0;
```

```
        n=n-1;
        j1=j1+1;
      else
        m=j1;
      end
end
[n,i]=size(v3);
if n==2
   a(j1+1)=-v3(1)/v3(2);                          % caso di un solo zero non nullo
end
if n==3
   Delta=sqrt((v3(2))^2-4*v3(3)*v3(1));           % caso di due zeri non nulli :
   a(j1+1)=(-v3(2)-Delta)/(2*v3(3));              % solo in questo caso vengono
   a(j1+2)=(-v3(2)-Delta)/(2*v3(3));              % calcolati anche gli zeri
   dom=1;                                         % complessi
end
if n>3
a(j1+1)=0;
                                                  % fa in modo che non si considerino
w=zeros(num,1);                                   % gli zeri ai quali é stata applicata
                                                  % la regola di Ruffini
for i=2:n                                         % calcolo di una maggiorazione
    a(j1+1)=a(j1+1)+abs(v3(i)/v3(1));             % dello zero di modulo massimo
end
a(j1+1)=max(1,a(j1+1));
for i=1:n-1                                       % calcolo dei coefficienti della
    v2(i)=i*v3(i+1);                              % derivata prima della funzione
end
c=valpol(a(j1+1),v3);
for j=j1:num-1
    if c~=0
       d=valpol(a(j+1),v2);                       % prima iterata
       e=0;
       for i=j1+1:j
          if w(i)==0
             e=e+c/(a(j+1)-a(i));
          end
       end
       e=d-e;
       if e==0
          risp=1;
       else
          l=a(j+1);
          a(j+1)=a(j+1)-c/e;
          f=valpol(a(j+1),v3);
          de=valpol(a(j+1),v2);
       end
       while (c*f>0) & (de*d>0) & (risp==0)
            l=a(j+1);
            c=f;
            d=de;                                 % metodo di Newton a doppio passo
            e=0;
            for i=j1+1:j
               if w(i)==0
                  e=e+c/(a(j+1)-a(i));
               end
            end
            e=d-e;
            if e==0
               risp=1;
```

```
            else
               a(j+1)=a(j+1)-2*c/e;
               f=valpol(a(j+1),v3);
               de=valpol(a(j+1),v2);

            end
        end
        if j~=num-1 & risp==0
           a(j+2)=a(j+1);

        end
        g=f;
        if f~=0 & risp==0
           while (((abs(f)>err) | (abs(l-a(j+1))>err))) & (risp==0)
               c=f;
               l=a(j+1);
               d=valpol(a(j+1),v2);                    % metodo di Newton -Rapson
               e=0;
               for i=j1+1:j
                  if w(i)==0
                     e=e+c/(a(j+1)-a(i));
                  end
               end
               e=d-e;
               if e==0
                  risp=1;
               else
                  a(j+1)=a(j+1)-c/e;
                  f=valpol(a(j+1),v3);
               end
           end
        end
        c=g;
     else
        if j~=num-1
           v4(n-1)=v3(n);
           for i=n-2:-1:1
              v4(i)=v3(i+1)+a(j+1)*v4(i+1);
           end
           v3=v4'
           v4=0;                                  % regola di Ruffini
           n=n-1;                                 % quando si conosce
           c=valpol(a(j+1),v3);                   % con precisione lo
           a(j+2)=a(j+1);                         % zero del polinomio
           w(j+1)=1;
           v2=0;
           for i=1:n-1
              v2(i)=i*v3(i+1);
           end
        end
     end
end
end
a=a';
```

ESECUZIONE DEI PROGRAMMI

Algebra

RISOLUZIONE DI SISTEMI LINEARI E FATTORIZZAZIONI DI MATRICI

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 1
Introduci la matrice da fattorizzare : [0 0 0;6 7 8;0 8 9]
Scegli il tipo di fattorizzazione :
1. Fattorizzazione LU (non applicabile per questo tipo di matrice)
2. Fattorizzazione QR
3. Fattorizzazione di Cholesky (non applicabile per questo tipo di matrice)
Introduci il numero corrispondente : 1
Il valore introdotto é errato.Introducilo di nuovo : 3
Il valore introdotto é errato.Introducilo di nuovo : 2
La fattorizzazione QR non si può ottenere
Ricomincia e utilizza un altro metodo.
Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente :-6.8
Il valore introdotto é errato.Ricomincia
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente :2
Introduci la matrice dei coefficienti : [0 0 0;6 7 8;0 8 9]
Il sistema non si può risolvere perché la matrice dei coefficienti non ha rango massimo
Infatti ha rango :

rango =

2

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 1
Introduci la matrice da fattorizzare : [0 6 0;6 7 8;0 8 9]
Scegli il tipo di fattorizzazione :
1. Fattorizzazione LU
2. Fattorizzazione QR
3. Fattorizzazione di Cholesky (non applicabile per questo tipo di matrice)
Introduci il numero corrispondente : 1
Scegli il metodo di calcolo della fattorizzazione LU :
1. Utilizzando il metodo di eliminazione di Gauss
2. Utilizzando la dimostrazione del teorema di esistenza ed unicità della fattorizzazione
3. Tecnica compatta
4. Metodo di Doolittle
5. Metodo per matrici simmetriche
6. Metodo per matrici tridiagonali
Introduci il numero corrispondente : 1
La fattorizzazione LU per questa matrice non si può ottenere
Scegli tra :

```
1. Calcolare la fattorizzazione QR
2. Uscire dal programma
Introduci il numero corrispondente : 1
la fattorizzazione QR della matrice :

     0     6     0
     6     7     8
     0     8     9

é la seguente :

Q =

         0    0.6000    0.8000
   -1.0000         0         0
         0    0.8000   -0.6000



R =

   -6.0000   -7.0000   -8.0000
         0   10.0000    7.2000
         0    0.0000   -5.4000


Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 1
Introduci la matrice da fattorizzare : [0 6 0;6 7 8;0 8 9]
Scegli il tipo di fattorizzazione :
1. Fattorizzazione LU
2. Fattorizzazione QR
3. Fattorizzazione di Cholesky (non applicabile per questo tipo di matrice)
Introduci il numero corrispondente : 1
Scegli il metodo di calcolo della fattorizzazione LU :
1. Utilizzando il metodo di eliminazione di Gauss
2. Utilizzando la dimostrazione del teorema di esistenza ed unicità della fattorizzazione
3. Tecnica compatta
4. Metodo di Doolittle
5. Metodo per matrici simmetriche
6. Metodo per matrici tridiagonali
Introduci il numero corrispondente : 5
La fattorizzazione LU per questa matrice non si può ottenere
Scegli tra :
1. Calcolare la fattorizzazione QR
2. Uscire dal programma
Introduci il numero corrispondente : 2
Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 1
Introduci la matrice da fattorizzare : [0 6 0;6 7 8;0 8 9]
Scegli il tipo di fattorizzazione :
1. Fattorizzazione LU
```

2. Fattorizzazione QR
3. Fattorizzazione di Cholesky (non applicabile per questo tipo di matrice)
Introduci il numero corrispondente : 1
Scegli il metodo di calcolo della fattorizzazione LU :
1. Utilizzando il metodo di eliminazione di Gauss
2. Utilizzando la dimostrazione del teorema diesistenza ed unicità della fattorizzazione
3. Tecnica compatta
4. Metodo di Doolittle
5. Metodo per matrici simmetriche
6. Metodo per matrici tridiagonali
Introduci il numero corrispondente : 6
La fattorizzazione LU per questa matrice non si può ottenere
Scegli tra :
1. Calcolare la fattorizzazione QR
2. Uscire dal programma
Introduci il numero corrispondente : 2
Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 2
Introduci la matrice dei coefficienti : [0 6 0;6 7 8;0 8 9]
Introduci il vettore colonna : [1;1;1]
Scegli il metodo di risoluzione :
Utilizzare :
1. - Il metodo di eliminazione Gauss
2. - La strategia di pivoting parziale nel metodo di eliminazione di Gauss
3. - La strategia di pivoting totale nel metodo di eliminazione di Gauss
4. - La fattorizzazione delle matrici
5. - Metodo di Jacobi (Non applicabile in questo caso.Fare un opportuno scambio di righe)
6. - Metodo di più ripida discesa (non applicabile per questo tipo di matrici)
Introduci il numero corrispondente : 1
Metodo non applicabile
Utilizzare le strategie di pivoting oppure la fattorizzazione QR
Utilizzare :
1. - Il metodo di eliminazione Gauss (Non applicabile in questo caso)
2. - La strategia di pivoting parziale nel metodo di eliminazione di Gauss
3. - La strategia di pivoting totale nel metodo di eliminazione di Gauss
4. - La fattorizzazione delle matrici
5. - Metodo di Jacobi (Non applicabile in questo caso.Fare un opportuno scambio di righe)
6. - Metodo di più ripida discesa (non applicabile per questo tipo di matrici)
Introduci il numero corrispondente : 4
Scegli il tipo di fattorizzazione :
1.Fattorizzazione LU (non applicabile in questo caso)
2. Fattorizzazione QR
3. Fattorizzazione di Cholesky (non applicabile in questo caso)
Introduci il numero corrispondente : 2
Il vettore soluzione del sistema lineare é :

x =

    0.0216   0.1667   -0.0370

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 2

Introduci la matrice dei coefficienti : [0 6 0;6 7 8;0 8 9]
Introduci il vettore colonna : [1;1;1]
Scegli il metodo di risoluzione :
Utilizzare :
1. - Il metodo di eliminazione Gauss
2. - La strategia di pivoting parziale nel metodo di eliminazione di Gauss
3. - La strategia di pivoting totale nel metodo di eliminazione di Gauss
4. - La fattorizzazione delle matrici
5. - Metodo di Jacobi (Non applicabile in questo caso.Fare un opportuno scambio di righe)
6. - Metodo di più ripida discesa (non applicabile per questo tipo di matrici)
Introduci il numero corrispondente : 4
Scegli il tipo di fattorizzazione :
1. Fattorizzazione LU
2. Fattorizzazione QR
3. Fattorizzazione di Cholesky (non applicabile in questo caso)
Introduci il numero corrispondente : 1
Metodo non applicabile
Utilizzare le strategie di pivoting oppure la fattorizzazione QR
Utilizzare :
1. - Il metodo di eliminazione Gauss (Non applicabile in questo caso)
2. - La strategia di pivoting parziale nel metodo di eliminazione di Gauss
3. - La strategia di pivoting totale nel metodo di eliminazione di Gauss
4. - La fattorizzazione delle matrici
5. - Metodo di Jacobi (Non applicabile in questo caso.Fare un opportuno scambio di righe)
6. - Metodo di più ripida discesa (non applicabile per questo tipo di matrici)
Introduci il numero corrispondente : 5
Il valore introdotto é errato.Introducilo di nuovo :
Introduci il numero corrispondente : 3
Il vettore soluzione del sistema lineare é :

x =

    0.0216   0.1667   -0.0370

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 6
Il valore introdotto é errato.Ricomincia
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 2
Introduci la matrice dei coefficienti : [0 6 0;6 7 8;0 8 9]
Introduci il vettore colonna : [1;1;1]
Scegli il metodo di risoluzione :
Utilizzare :
1. - Il metodo di eliminazione Gauss
2. - La strategia di pivoting parziale nel metodo di eliminazione di Gauss
3. - La strategia di pivoting totale nel metodo di eliminazione di Gauss
4. - La fattorizzazione delle matrici
5. - Metodo di Jacobi (Non applicabile in questo caso.Fare un opportuno scambio di righe)
6. - Metodo di più ripida discesa (non applicabile per questo tipo di matrici)
Introduci il numero corrispondente : 2
Il vettore soluzione del sistema lineare é :

```
x =

   0.0216    0.1667   -0.0370

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 1
Introduci la matrice da fattorizzare : [3 2 -1 4;-4 2 -3 -2;1 0 1 5;3 7 -2 1]
Scegli il tipo di fattorizzazione :
1. Fattorizzazione LU
2. Fattorizzazione QR
3. Fattorizzazione di Cholesky (non applicabile per questo tipo di matrice)
Introduci il numero corrispondente : 1
Scegli il metodo di calcolo della fattorizzazione LU :
1. Utilizzando il metodo di eliminazione di Gauss
2. Utilizzando la dimostrazione del teorema di esistenza ed unicità della fattorizzazione
3. Tecnica compatta
4. Metodo di Doolittle
5. Metodo per matrici simmetriche (non applicabile in questo caso)
6. Metodo per matrici tridiagonali (non applicabile in questo caso)
Introduci il numero corrispondente : 1
 la fattorizzazione LU della matrice :
     3     2    -1     4
    -4     2    -3    -2
     1     0     1     5
     3     7    -2     1

é la seguente :

L =

    1.0000         0         0         0
   -1.3333    1.0000         0         0
    0.3333   -0.1429    1.0000         0
    1.0000    1.0714    5.1000    1.0000

U =

    3.0000    2.0000   -1.0000    4.0000
         0    4.6667   -4.3333    3.3333
         0         0    0.7143    4.1429
         0         0         0  -27.7000


Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 1
Introduci la matrice da fattorizzare : [3 2 -1 4;-4 2 -3 -2;1 0 1 5;3 7 -2 1]
Scegli il tipo di fattorizzazione :
1. Fattorizzazione LU
2. Fattorizzazione QR
3. Fattorizzazione di Cholesky (non applicabile per questo tipo di matrice)
Introduci il numero corrispondente : 1
Scegli il metodo di calcolo della fattorizzazione LU :
```

1. Utilizzando il metodo di eliminazione di Gauss
2. Utilizzando la dimostrazione del teorema di esistenza ed unicità della fattorizzazione
3. Tecnica compatta
4. Metodo di Doolittle
5. Metodo per matrici simmetriche (non applicabile in questo caso)
6. Metodo per matrici tridiagonali (non applicabile in questo caso)

Introduci il numero corrispondente : 2
la fattorizzazione LU della matrice :

```
     3     2    -1     4
    -4     2    -3    -2
     1     0     1     5
     3     7    -2     1
```

é la seguente :

L =

```
    1.0000         0         0         0
   -1.3333    1.0000         0         0
    0.3333   -0.1429    1.0000         0
    1.0000    1.0714    5.1000    1.0000
```

U =

```
    3.0000    2.0000   -1.0000    4.0000
         0    4.6667   -4.3333    3.3333
         0         0    0.7143    4.1429
         0         0         0  -27.7000
```

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari

Introduci il numero corrispondente : 1
Introduci la matrice da fattorizzare : [3 2 -1 4;-4 2 -3 -2;1 0 1 5;3 7 -2 1]
Scegli il tipo di fattorizzazione :
1. Fattorizzazione LU
2. Fattorizzazione QR
3. Fattorizzazione di Cholesky (non applicabile per questo tipo di matrice)

Introduci il numero corrispondente : 1
Scegli il metodo di calcolo della fattorizzazione LU :
1. Utilizzando il metodo di eliminazione di Gauss
2. Utilizzando la dimostrazione del teorema di esistenza ed unicità della fattorizzazione
3. Tecnica compatta
4. Metodo di Doolittle
5. Metodo per matrici simmetriche (non applicabile in questo caso)
6. Metodo per matrici tridiagonali (non applicabile in questo caso)

Introduci il numero corrispondente : 3
la fattorizzazione LU della matrice :

```
     3     2    -1     4
    -4     2    -3    -2
     1     0     1     5
     3     7    -2     1
```

é la seguente :

```
L =

    1.0000         0         0         0
   -1.3333    1.0000         0         0
    0.3333   -0.1429    1.0000         0
    1.0000    1.0714    5.1000    1.0000

U =

    3.0000    2.0000   -1.0000    4.0000
         0    4.6667   -4.3333    3.3333
         0         0    0.7143    4.1429
         0         0         0  -27.7000
```

Se vuoi ricominciare nuovamente premi il tasto'1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 2
Introduci la matrice dei coefficienti : [3 2 -1 4;-4 2 -3 -2;1 0 1 5;3 7 -2 1]
Introduci il vettore colonna : [-1;0;6;1]
Scegli il metodo di risoluzione :
Utilizzare :
1. - Il metodo di eliminazione Gauss
2. - La strategia di pivoting parziale nel metodo di eliminazione di Gauss
3. - La strategia di pivoting totale nel metodo di eliminazione diGauss
4. - La fattorizzazione delle matrici
5. - Metodo di Jacobi
6. - Metodo di più ripida discesa (non applicabile per questo tipo di matrici)
Introduci il numero corrispondente : 1
Il vettore soluzione del sistema lineare é :

```
x =

   -1.7942    1.5560    2.7581    1.0072
```

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 2
Introduci la matrice dei coefficienti : [3 2 -1 4;-4 2 -3 -2;1 0 1 5;3 7 -2 1]
Introduci il vettore colonna : [-1;0;6;1]
Scegli il metodo di risoluzione :
Utilizzare :
1. - Il metodo di eliminazione Gauss
2. - La strategia di pivoting parziale nel metodo di eliminazione di Gass
3. - La strategia di pivoting totale nel metodo di eliminazione di Gauss
4. - La fattorizzazione delle matrici
5. - Metodo di Jacobi
6. - Metodo di più ripida discesa (non applicabile per questo tipo di matrici)
Introduci il numero corrispondente : 6
Il valore introdotto é errato.Introducilo di nuovo : 5
Per una verifica molto precisa sulla convergenza del metodo iterativo
adottato introdurre un valore abbastanza elevato del seguente numero
Introduci il numero massimo di iterate : 1000
Metodo non applicabile perché non convergente

Utilizzare altri metodi
Utilizzare :
1. - Il metodo di eliminazione Gauss
2. - La strategia di pivoting parziale nel metodo di eliminazione di Gauss
3. - La strategia di pivoting totale nel metodo di eliminazione di Gauss
4. - La fattorizzazione delle matrici
5. - Metodo di Jacobi
6. - Metodo di più ripida discesa (non applicabile per questo tipo di matrici)
Introduci il numero corrispondente : 3
Il vettore soluzione del sistema lineare é :

```
x =

   -1.7942    1.5560    2.7581    1.0072
```

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 1
Introduci la matrice da fattorizzare : [1 2 0 0;2 -3 -1 0;0 -1 -4 5;0 0 5 2]
Scegli il tipo di fattorizzazione :
1. Fattorizzazione LU
2. Fattorizzazione QR
3. Fattorizzazione di Cholesky (non applicabile per questo tipo di matrice)
Introduci il numero corrispondente : 1
Scegli il metodo di calcolo della fattorizzazione LU :
1. Utilizzando il metodo di eliminazione di Gauss
2. Utilizzando la dimostrazione del teorema di esistenza ed unicità della fattorizzazione
3. Tecnica compatta
4. Metodo di Doolittle
5. Metodo per matrici simmetriche
6. Metodo per matrici tridiagonali
Introduci il numero corrispondente : 5
La matrice non é anche definita positivamente
per cui non é fattorizzabile secondo CHOLESKI
la fattorizzazione LU della matrice :

```
     1     2     0     0
     2    -3    -1     0
     0    -1    -4     5
     0     0     5     2
```

é la seguente :

```
L =

    1.0000         0         0         0
    2.0000    1.0000         0         0
         0    0.1429    1.0000         0
         0         0   -1.2963    1.0000


U =

    1.0000    2.0000         0         0
         0   -7.0000   -1.0000         0
         0         0   -3.8571    5.0000
         0         0         0    8.4815
```

```
Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 1
Introduci la matrice da fattorizzare : [1 2 0 0;2 -3 -1 0;0 -1 -4 5;0 0 5 2 ]
Scegli il tipo di fattorizzazione :
1. Fattorizzazione LU
2. Fattorizzazione QR
3. Fattorizzazione di Cholesky (non applicabile per questo tipo di matrice)
Introduci il numero corrispondente : 1
Scegli il metodo di calcolo della fattorizzazione LU :
1. Utilizzando il metodo di eliminazione di Gauss
2. Utilizzando la dimostrazione del teorema di esistenza ed unicità della fattorizzazione
3. Tecnica compatta
4. Metodo di Doolittle
5. Metodo per matrici simmetriche
6. Metodo per matrici tridiagonali
Introduci il numero corrispondente : 6
la fattorizzazione LU della matrice :
     1     2     0     0
     2    -3    -1     0
     0    -1    -4     5
     0     0     5     2

é la seguente :

L =

    1.0000         0         0         0
    2.0000    1.0000         0         0
         0    0.1429    1.0000         0
         0         0   -1.2963    1.0000


U =

    1.0000    2.0000         0         0
         0   -7.0000   -1.0000         0
         0         0   -3.8571    5.0000
         0         0         0    8.4815


Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 1
Introduci la matrice da fattorizzare : [3 2 0 0;2 4 -1 0;0 -1 4 1;0 0 1 2 ]
Scegli il tipo di fattorizzazione :
1. Fattorizzazione LU
2. Fattorizzazione QR
3. Fattorizzazione di Cholesky
Introduci il numero corrispondente : 3
Scegli il metodo di calcolo della fattorizzazione di Cholesky :
1. Tecnica compatta
2. Metodo di Doolittle
Introduci il numero corrispondente : 1
```

La matrice di Cholesky della matrice :

```
     3     2     0     0
     2     4    -1     0
     0    -1     4     1
     0     0     1     2
```

é la seguente :

L =

```
    1.7321         0         0         0
    1.1547    1.6330         0         0
         0   -0.6124    1.9039         0
         0         0    0.5252    1.3131
```

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 1
Introduci la matrice da fattorizzare : [3 2 0 0;2 4-1 0;0 -1 4 1;0 0 1 2]
Scegli il tipo di fattorizzazione :
1. Fattorizzazione LU
2. Fattorizzazione QR
3. Fattorizzazione di Cholesky
Introduci il numero corrispondente : 3
Scegli il metodo di calcolo della fattorizzazione di Cholesky :
1. Tecnica compatta
2. Metodo di Doolittle
Introduci il numero corrispondente : 5
Il valore introdotto é errato.Ricomincia
1. Tecnica compatta
2. Metodo di Doolittle
Introduci il numero corrispondente : 2
La matrice di Cholesky della matrice :

```
     3     2     0     0
     2     4    -1     0
     0    -1     4     1
     0     0     1     2
```

é la seguente :

L =

```
    1.7321         0         0         0
    1.1547    1.6330         0         0
         0   -0.6124    1.9039         0
         0         0    0.5252    1.3131
```

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 1
Introduci la matrice da fattorizzare : [3 2 0 0;2 4-1 0;0 -1 4 1;0 0 1 2]
Scegli il tipo di fattorizzazione :

```
1. Fattorizzazione LU
2. Fattorizzazione QR
3. Fattorizzazione di Cholesky
Introduci il numero corrispondente : 1
Scegli il metodo di calcolo della fattorizzazione LU :
1. Utilizzando il metodo di eliminazione di Gauss
2. Utilizzando la dimostrazione del teorema di esistenza ed unicità della fattorizzazione
3. Tecnica compatta
4. Metodo di Doolittle
5. Metodo per matrici simmetriche
6. Metodo per matrici tridiagonali
Introduci il numero corrispondente : 5
La matrice puo' essere fattorizzabile secondo CHOLESKY
vuoi calcolarla? (se SI premere ' 1 ') : 1
La matrice di Cholesky di A é :

C =

    1.7321         0         0         0
    1.1547    1.6330         0         0
         0   -0.6124    1.9039         0
         0         0    0.5252    1.3131

la fattorizzazione LU della matrice :
     3     2     0     0
     2     4    -1     0
     0    -1     4     1
     0     0     1     2

é la seguente :


L =

    1.0000         0         0         0
    0.6667    1.0000         0         0
         0   -0.3750    1.0000         0
         0         0    0.2759    1.0000


U =

    3.0000    2.0000         0         0
         0    2.6667   -1.0000         0
         0         0    3.6250    1.0000
         0         0         0    1.7241

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 2
Introduci la matrice dei coefficienti : [3 -2 0 0;-2 4 -1 0;0 -1 4 1;0 0 1 2]
Introduci il vettore colonna  : [-1;0;6;1]
Scegli il metodo di risoluzione :
Utilizzare :
1. - Il metodo di eliminazione Gauss
2. - La strategia di pivoting parziale nel metodo di eliminazione di Gauss
3. - La strategia di pivoting totale nel metodo di eliminazione di Gauss
```

4. - La fattorizzazione delle matrici
5. - Metodo di Jacobi
6. - Metodo di più ripida discesa
Introduci il numero corrispondente : 4
Scegli il tipo di fattorizzazione :
1. Fattorizzazione LU
2. Fattorizzazione QR
3. Fattorizzazione di Cholesky
Introduci il numero corrispondente : 1
Il vettore soluzione del sistema lineare é :

x =

   -0.0800   0.3800   1.6800   -0.3400

Se vuoi ricominciare nuovamente premi il tasto '1' : 1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 2
Introduci la matrice dei coefficienti : [3 -2 0 0;-2 4 -1 0;0 -1 4 1;0 0 1 2]
Introduci il vettore colonna : [-1;0;6;1]
Scegli il metodo di risoluzione :
Utilizzare :
1. - Il metodo di eliminazione Gauss
2. - La strategia di pivoting parziale nel metodo di eliminazione di Gauss
3. - La strategia di pivoting totale nel metodo di eliminazione di Gauss
4. - La fattorizzazione delle matrici
5. - Metodo di Jacobi
6. - Metodo di più ripida discesa
Introduci il numero corrispondente : 4
Scegli il tipo di fattorizzazione :
1. Fattorizzazione LU
2. Fattorizzazione QR
3. Fattorizzazione di Cholesky
Introduci il numero corrispondente : 3
Il vettore soluzione del sistema lineare é :

x =

   -0.0800   0.3800   1.6800   -0.3400

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 2
Introduci la matrice dei coefficienti : [3 -2 0 0;-2 4 -1 0;0 -1 4 1;0 0 1 2]
Introduci il vettore colonna : [-1;0;6;1]
Scegli il metodo di risoluzione :
Utilizzare :
1. - Il metodo di eliminazione Gauss
2. - La strategia di pivoting parziale nel metodo di eliminazione di Gauss
3. - La strategia di pivoting totale nel metodo di eliminazione di Gauss
4. - La fattorizzazione delle matrici
5. - Metodo di Jacobi
6. - Metodo di più ripida discesa
Introduci il numero corrispondente : 5

Per una verifica molto precisa sulla convergenza del metodo iterativo
adottato introdurre un valore abbastanza elevato del seguente numero
Introduci il numero massimo di iterate : 100
Introduci il numero massimo di iterate per applicare il metodo di Jacobi : 50
Il vettore soluzione del sistema lineare é :

x =

-0.0800 0.3800 1.6800 -0.3400

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 2
Introduci la matrice dei coefficienti : [3 -2 0 0;-2 4 -1 0;0 -1 4 1;0 0 1 2]
Introduci il vettore colonna : [-1;0;6;1]
Scegli il metodo di risoluzione :
Utilizzare :
1. - Il metodo di eliminazione Gauss
2. - La strategia di pivoting parziale nel metodo di eliminazionedi Gauss
3. - La strategia di pivoting totale nel metodo di eliminazione di Gauss
4. - La fattorizzazione delle matrici
5. - Metodo di Jacobi
6. - Metodo di più ripida discesa
Introduci il numero corrispondente : 6
Introduci l'errore massimo che si può commettere (al massimo 0.1) : -7
Il valore dell'errore massimo introdotto non é corretto
Introducilo nuovamente : 1
Il valore dell'errore massimo introdotto non é corretto
Introducilo nuovamente : 0.000001
Il vettore soluzione del sistema lineare é :

x =

-0.0800 0.3800 1.6800 -0.3400

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 2
Introduci la matrice dei coefficienti : [3 2 -1 4;-4 2 -3 -2;1 0 1 5;3 7 -2 1;-1 0 2 5]
Introduci il vettore colonna : [-1;0;6;1]
Il vettore introdotto ha dimensioni errate.Ricomincia
Introduci nuovamente il vettore colonna : [-1;0;6;1;0]
Il sistema non ammette soluzioni
Verrà calcolata quindi la soluzione di minima norma
utilizzando il problema dei minimi quadrati
la soluzione ai minimi quadrati é :

x =

-0.1277 0.0639 0.0306 0.4053

dove il vettore residuo ha norma 2 pari a :

```
a =

    5.2298

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 1
Introduci la matrice da fattorizzare : [1 24 2;-3 -1 5 -3;4 0 -2 4;-5 3 0 5;6 1 -4 -6]
Scegli il tipo di fattorizzazione :
1. Fattorizzazione LU (non applicabile per questo tipo di matrice)
2. Fattorizzazione QR
3. Fattorizzazione di Cholesky (non applicabile per questo tipo di matrice)
Introduci il numero corrispondente : 1
Il valore introdotto é errato.Introducilo di nuovo : 3
Il valore introdotto é errato.Introducilo di nuovo : 2
la fattorizzazione QR della matrice :
     1     2     4     2
    -3    -1     5    -3
     4     0    -2     4
    -5     3     0     5
     6     1    -4    -6

é la seguente :

Q =

   -0.1072    0.5315    0.7840   -0.1284   -0.2736
    0.3216   -0.2956    0.5256    0.3367    0.6477
   -0.4288    0.0478    0.0022   -0.6838    0.5884
    0.5361    0.7197   -0.3060   -0.0136    0.3176
   -0.6433    0.3315   -0.1244    0.6343    0.2419

R =

   -9.3274    0.4288    4.6101    3.6452
    0.0000    3.8492   -0.7734    3.7506
    0.0000    0.0000    6.2569   -0.7837
    0.0000    0.0000    0.0000   -7.8760
    0.0000    0.0000    0.0000    0.0000

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli il problema da risolvere :
1. Calcolo di fattorizzazioni di matrici
2. Risoluzione di sistemi lineari
Introduci il numero corrispondente : 2
Introduci la matrice dei coefficienti : [1 2 4 2;-3 -1 5 -3;4 0 -2 4;-5 3 0 5;6 1 -4 -6]
Introduci il vettore colonna  : [0;-5;8;-13;11]
la soluzione é :

x =

    2.0000   -1.0000    0.0000    0.0000

Se vuoi ricominciare nuovamente premi il tasto '1' :0
diary off
```

```
autoval
  CALCOLO DEGLI AUTOVALORI DI UNA MATRICE

Introduci la matrice (di ordine almeno 2x2) : [1 2 4 2 0;-3 -1 5 -3 -5;4 0 -2 4 8;-5 3 0 5 -13;6 1 -4 -6 11]
Scegliere tra i seguenti metodi di calcolo degli autovalori:
1. Metodo delle potenze (per il calcolo dell'autovalore di modulo minimo)
                              (non applicabile per questo tipo di matrice)
2. Metodo delle potenze (per il calcolo dell'autovalore di modulo massimo)
3. Metodo delle potenze inverse (per il calcolo di un particolare autovalore)
4. Metodo di calcolo di tutti gli autovalori reali di matrici triangolari simmetriche
                              (non applicabile per questo tipo di matrice)
5. Metodo di calcolo dell'autovalore di modulo minimoper matrici simmetriche
                              (non applicabile per questo tipo di matrice)
6. Metodo di calcolo dell'autovalore massimo per matrici simmetriche
                              (non applicabile per questo tipo di matrice)
7. Metodo QR (per il calcolo di tutti gli autovalori)
 ATTENZIONE : i primi 3 metodi convergono sempre SOLO SE la matrice ha solo autovalori reali
Introduci il numero corrispondente : 1
Il valore introdotto é errato.Introducilo di nuovo : 2
Introduci il numero massimo di iterate : 50
L'autovalore di modulo massimo della matrice :
     1     2     4     2     0
    -3    -1     5    -3    -5
     4     0    -2     4     8
    -5     3     0     5   -13
     6     1    -4    -6    11

é il seguente :

l =

   16.3216

Se vuoi ricominciare nuovamente premi il tasto '0' :0

Introduci la matrice (di ordine almeno 2x2) : [1 2 4 2 0;-3 -1 5 -3 -5;4 0 -2 4 8;-5 3 0 5 -13;6 1 -4 -6 11]
Scegliere tra i seguenti metodi di calcolo degli autovalori:

1. Metodo delle potenze (per il calcolo dell'autovalore di modulo minimo)
                              (non applicabile per questo tipo di matrice)
2. Metodo delle potenze (per il calcolo dell'autovalore di modulo massimo)
3. Metodo delle potenze inverse (per il calcolo di un particolare autovalore)
4. Metodo di calcolo di tutti gli autovalori reali di matrici triangolari simmetriche
                              (non applicabile per questo tipo di matrice)
5. Metodo di calcolo dell'autovalore di modulo minimo per matrici simmetriche
                              (non applicabile per questo tipo di matrice)
6. Metodo di calcolo dell'autovalore massimo per matrici simmetriche
                              (non applicabile per questo tipo di matrice)
7. Metodo QR (per il calcolo di tutti gli autovalori)
 ATTENZIONE : i primi 3 metodi convergono sempre SOLO SE la matrice ha solo autovalori reali
Introduci il numero corrispondente : 3
Inserisci il valore approssimato dell'autovalore da calcolare : 1
Metodo non applicabile
Utilizzare altri metodi se possibile oppure fissare un autovalore approssimato diverso :
Scegli nuovamente :
1. Metodo delle potenze (per il calcolo dell'autovalore di modulo minimo)
                              (non applicabile per questo tipo di matrice)
2. Metodo delle potenze (per il calcolo dell'autovalore di modulo massimo)
3. Metodo delle potenze inverse (per il calcolo di un particolare autovalore)
4. Metodo di calcolo di tutti gli autovalori reali di matrici triangolari simmetriche
                                 (non applicabile per questo tipo di matrice)
```

5. Metodo di calcolo dell'autovalore di modulo minimo per matrici simmetriche
(non applicabile per questo tipo di matrice)
6. Metodo di calcolo dell'autovalore massimo per matrici simmetriche
(non applicabile per questo tipo di matrice)
7. Metodo QR (per il calcolo di tutti gli autovalori)

ATTENZIONE : i primi 3 metodi convergono sempre SOLO SE la matrice ha solo autovalori reali
Introduci il numero corrispondente : 3
Inserisci il valore approssimato dell'autovalore da calcolare : 0.9
Introduci il numero massimo di iterate : 50
L'autovalore della matrice :

```
     1     2     4     2     0
    -3    -1     5    -3    -5
     4     0    -2     4     8
    -5     3     0     5   -13
     6     1    -4    -6    11
```

é il seguente :

H =

5.5511e-016

Se vuoi ricominciare nuovamente premi il tasto '0' :0

Introduci la matrice (di ordine almeno 2x2) : [1 2 4 2 0;-3 -1 5 -3 -5;4 0 -2 4 8;-5 3 0 5 -13;6 1 -4 -6 11]
Scegliere tra i seguenti metodi di calcolo degli autovalori:
1. Metodo delle potenze (per il calcolo dell'autovalore dimodulo minimo)
(non applicabile per questo tipo di matrice)
2. Metodo delle potenze (per il calcolo dell'autovalore di modulo massimo)
3. Metodo delle potenze inverse (per il calcolo di un particolare autovalore)
4. Metodo di calcolo di tutti gli autovalori reali di matrici triangolari simmetriche
(non applicabile per questo tipo di matrice)
5. Metodo di calcolo dell'autovalore di modulo minimo per matrici simmetriche
(non applicabile per questo tipo di matrice)
6. Metodo di calcolo dell'autovalore massimo per matrici simmetriche
(non applicabile per questo tipo di matrice)
7. Metodo QR (per il calcolo di tutti gli autovalori)

ATTENZIONE : i primi 3 metodi convergono sempre SOLO SE la matrice ha solo autovalori reali
Introduci il numero corrispondente : 7
Introduci il numero massimo di iterate : 50
Tutti gli autovalori della matrice :

```
     1     2     4     2     0
    -3    -1     5    -3    -5
     4     0    -2     4     8
    -5     3     0     5   -13
     6     1    -4    -6    11
```

sono i seguenti :

l =

```
  16.3216
  -2.8271 + 6.2244i
  -2.8271 - 6.2244i
   3.3325
   0.0000
```

```
Se vuoi ricominciare nuovamente premi il tasto '0' :0
Introduci la matrice (di ordine almeno 2x2) : [1 2 0 0;2-3 -1 0;0 -1 -4 5;0 0 5 2]
Scegliere tra i seguenti metodi di calcolo degli autovalori:
1. Metodo delle potenze (per il calcolo dell'autovalore di modulo minimo)
2. Metodo delle potenze (per il calcolo dell'autovalore di modulo massimo)
3. Metodo delle potenze inverse (per il calcolo di un particolare autovalore)
4. Metodo di calcolo di tutti gli autovalori reali di matrici triangolari simmetriche
5. Metodo di calcolo dell'autovalore minimo per matrici simmetriche
6. Metodo di calcolo dell'autovalore massimo per matrici simmetriche
7. Metodo QR (per il calcolo di tutti gli autovalori)
Introduci il numero corrispondente: 0
Il valore introdotto é errato.Introducilo di nuovo : 4
Introduci l'errore massimo che si può commettere (massimo 0.1) : 1
Il valore dell'errore massimo introdotto non é corretto
Introducilo nuovamente : 0.00001
Tutti gli autovalori della matrice :

     1     2     0     0
     2    -3    -1     0
     0    -1    -4     5
     0     0     5     2

sono i seguenti :

l =

   -7.0456
   -3.6506
    1.8294
    4.8668

Se vuoi ricominciare nuovamente premi il tasto '0' :0

Introduci la matrice (di ordine almeno 2x2) : 5
La matrice deve essere quadrata e di ordine almeno 2x2
Introduci nuovamente la matrice:[2 3 0 7;6 8-3 -1;0 5 -5 2]
La matrice deve essere quadrata e di ordine almeno 2x2
Introduci nuovamente la matrice:[1 2 0 0;2-3 -1 0;0 -1 -4 5;0 0 5 2]
Scegliere tra i seguenti metodi di calcolo degli autovalori:
1. Metodo delle potenze (per il calcolo dell'autovalore di modulo minimo)
2. Metodo delle potenze (per il calcolo dell'autovalore di modulo massimo)
3. Metodo delle potenze inverse (per il calcolo di un particolare autovalore)
4. Metodo di calcolo di tutti gli autovalori reali di matrici triangolari simmetriche
5. Metodo di calcolo dell'autovalore minimo per matrici simmetriche
6. Metodo di calcolo dell'autovalore massimo per matrici simmetriche
7. Metodo QR (per il calcolo di tutti gli autovalori)
Introduci il numero corrispondente : 5
Introduci il numero massimo di iterate : 50
L'autovalore di modulo minimo della matrice :

     1     2     0     0
     2    -3    -1     0
     0    -1    -4     5
     0     0     5     2

é il seguente :

l =

    1.8294
```

Se vuoi ricominciare nuovamente premi il tasto '0' :0

Introduci la matrice (di ordine almeno 2x2) : [1 2 0 0;2-3 -1 0;0 -1 -4 5;0 0 5 2]
Scegliere tra i seguenti metodi di calcolo degli autovalori:
1. Metodo delle potenze (per il calcolo dell'autovalore di modulo minimo)
2. Metodo delle potenze (per il calcolo dell'autovalore di modulo massimo)
3. Metodo delle potenze inverse (per il calcolo di un particolare autovalore)
4. Metodo di calcolo di tutti gli autovalori reali di matrici triangolari simmetriche
5. Metodo di calcolo dell'autovalore minimo per matrici simmetriche
6. Metodo di calcolo dell'autovalore massimo per matrici simmetriche
7. Metodo QR (per il calcolo di tutti gli autovalori)
Introduci il numero corrispondente : 6
Introduci il numero massimo di iterate : 50
L'autovalore di modulo massimo della matrice :

```
     1     2     0     0
     2    -3    -1     0
     0    -1    -4     5
     0     0     5     2
```

é il seguente :

```
l =

   -7.0456
```

Se vuoi ricominciare nuovamente premi il tasto '0' :0

Introduci la matrice (di ordine almeno 2x2) : [1 2 0 0;2-3 -1 0;0 -1 -4 5;0 0 5 2]
Scegliere tra i seguenti metodi di calcolo degli autovalori:
1. Metodo delle potenze (per il calcolo dell'autovalore di modulo minimo)
2. Metodo delle potenze (per il calcolo dell'autovalore di modulo massimo)
3. Metodo delle potenze inverse (per il calcolo di un particolare autovalore)
4. Metodo di calcolo di tutti gli autovalori reali di matrici triangolari simmetriche
5. Metodo di calcolo dell'autovalore minimo per matrici simmetriche
6. Metodo di calcolo dell'autovalore massimo per matrici simmetriche
7. Metodo QR (per il calcolo di tutti gli autovalori)
Introduci il numero corrispondente : 7
Introduci il numero massimo di iterate : 50
Tutti gli autovalori della matrice :

```
     1     2     0     0
     2    -3    -1     0
     0    -1    -4     5
     0     0     5     2
```

sono i seguenti :

```
l =

   -7.0456
    4.8668
   -3.6506
    1.8294
```

Se vuoi ricominciare nuovamente premi il tasto '0' :0

Introduci la matrice (di ordine almeno 2x2) : [0 8 6 ;0 4-3;0 -1 -5]
Scegliere tra i seguenti metodi di calcolo degli autovalori:
1. Metodo delle potenze (per il calcolo dell'autovalore di modulo minimo)
(non applicabile per questo tipo di matrice)
2. Metodo delle potenze (per il calcolo dell'autovalore di modulo massimo)
3. Metodo delle potenze inverse (per il calcolo di un particolare autovalore)
4. Metodo di calcolo di tutti gli autovalori reali di matrici triangolari simmetriche
(non applicabile per questo tipo di matrice)
5. Metodo di calcolo dell'autovalore di modulo minimo per matrici simmetriche
(non applicabile per questo tipo di matrice)
6. Metodo di calcolo dell'autovalore massimo per matrici simmetriche
(non applicabile per questo tipo di matrice)
7. Metodo QR (per il calcolo di tutti gli autovalori)
ATTENZIONE : i primi 3 metodi convergono sempre SOLO SE la matrice ha solo autovalori reali
Introduci il numero corrispondente : 7
Introduci il numero massimo di iterate : 50

Metodo non applicabile. Provare con gli altri se é possibile

Se vuoi ricominciare nuovamente premi il tasto '0' :0

Introduci la matrice (di ordine almeno 2x2) : [8 -8 -2;4 -3 -2;3 -4 1]
Scegliere tra i seguenti metodi di calcolo degli autovalori:
1. Metodo delle potenze (per il calcolo dell'autovalore di modulo minimo)
2. Metodo delle potenze (per il calcolo dell'autovalore di modulo massimo)
3. Metodo delle potenze inverse (per il calcolo di un particolare autovalore)
4. Metodo di calcolo di tutti gli autovalori reali di matrici triangolari simmetriche
(non applicabile per questo tipo di matrice)
5. Metodo di calcolo dell'autovalore di modulo minimo per matrici simmetriche
(non applicabile per questo tipo di matrice)
6. Metodo di calcolo dell'autovalore massimo per matrici simmetriche
(non applicabile per questo tipo di matrice)
7. Metodo QR (per il calcolo di tutti gli autovalori)
ATTENZIONE : i primi 3 metodi convergono sempre SOLO SE la matrice ha solo autovalori reali

Introduci il numero corrispondente : 1

Introduci il numero massimo di iterate : 50
L'autovalore di modulo minimo della matrice :

```
     8    -8    -2
     4    -3    -2
     3    -4     1
```

é il seguente :

l =

1

Se vuoi ricominciare nuovamente premi il tasto '0' :0

Introduci la matrice (di ordine almeno 2x2) : [0 0 0;0 0 0;0 0 0]
Hai introdotto una matrice nulla per cui i suoi autovalori sono tutti nulli

Se vuoi ricominciare nuovamente premi il tasto '0' :9

diary off

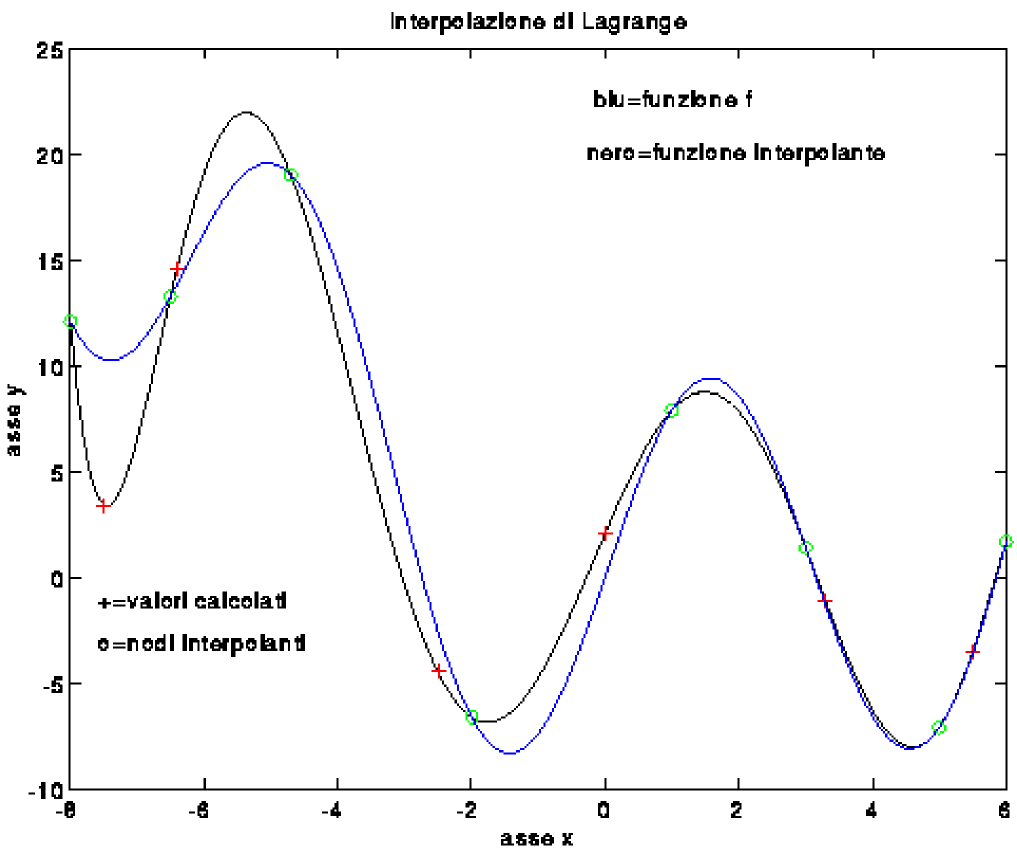
OSS: per ogni tipo di interpolazione eseguito è stato tracciato il grafico corrispondente

Interpol

INTERPOLAZIONE DI FUNZIONI

Scegli il tipo di interpolazione :
1. Interpolazione di Lagrange
2. Interpolazione di Newton
3. Utilizzando lo schema di Neville
4. Interpolazione di Hermite
5. Interpolazione con spline lineari
6. Interpolazione con spline cubiche naturali
7. Interpolazione con spline cubiche complete
8. Approssimazione polinomiale ai minimi quadrati
Introduci il numero corrispondente : 1
Introduci :
- il vettore colonna dei nodi interpolanti : [-2;5;1;3;-8;-4.7;6;-6.5]
- il vettore colonna dei corrispondenti valori della funzione : [-6.59;-7.09;7.91;1.41;12.11;19.05;1.7;13.29]
Se conosci la funzione f corrispondente ai punti introdotti premere "1" : 1
- Il nome della function contenente la funzione : funz
- il vettore colonna con i punti nei quali calcolare la funzione interpolante : [5.5;-7.5;0;3.3;-2.5;-6.4]
Il vettore soluzione y corrispondente ai punti x immessi é :

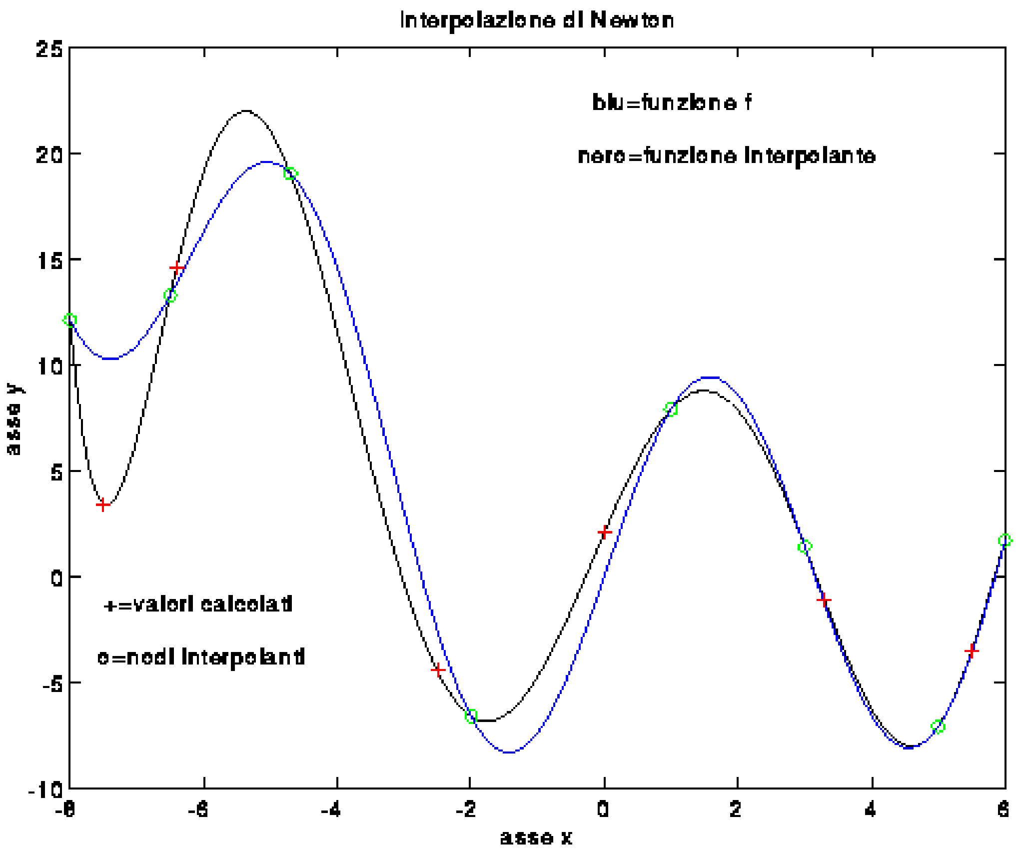
x=	y=
5.5000	-3.4996
-7.5000	3.4252
0	2.0756
3.3000	-1.1198
-2.5000	-4.4667
-6.4000	14.6329



Se vuoi ricominciare premi "0" : 0

Scegli il tipo di interpolazione :
1. Interpolazione di Lagrange
2. Interpolazione di Newton
3. Utilizzando lo schema di Neville
4. Interpolazione di Hermite
5. Interpolazione con spline lineari
6. Interpolazione con spline cubiche naturali
7. Interpolazione con spline cubiche complete
8. Approssimazione polinomiale ai minimi quadrati
Introduci il numero corrispondente : 2
Introduci :
- il vettore colonna dei nodi interpolanti : [-2;5;1;3;-8;-4.7;6;-6.5]
- il vettore colonna dei corrispondenti valori della funzione : [-6.59;-7.09;7.91;1.41;12.11;19.05;1.7;13.29]
Se conosci la funzione f corrispondente ai punti introdotti premere "1" : 1
Introduci :
- Il nome della function contenente la funzione : funzion
La funzione introdotta non esiste,introduci :
Introduci :
- Il nome della function contenente la funzione : funz
- il vettore colonna con i punti nei quali calcolare la funzione interpolante : [4,7;7,8]
Non é stato introdotto un vettore colonna.Ricomincia
Introduci :
- il vettore colonna con i punti nei quali calcolare la funzione interpolante : [5.5;-7.5;0;3.3;-2.5;-6.4]
Il vettore soluzione y corrispondente ai punti x immessi é :

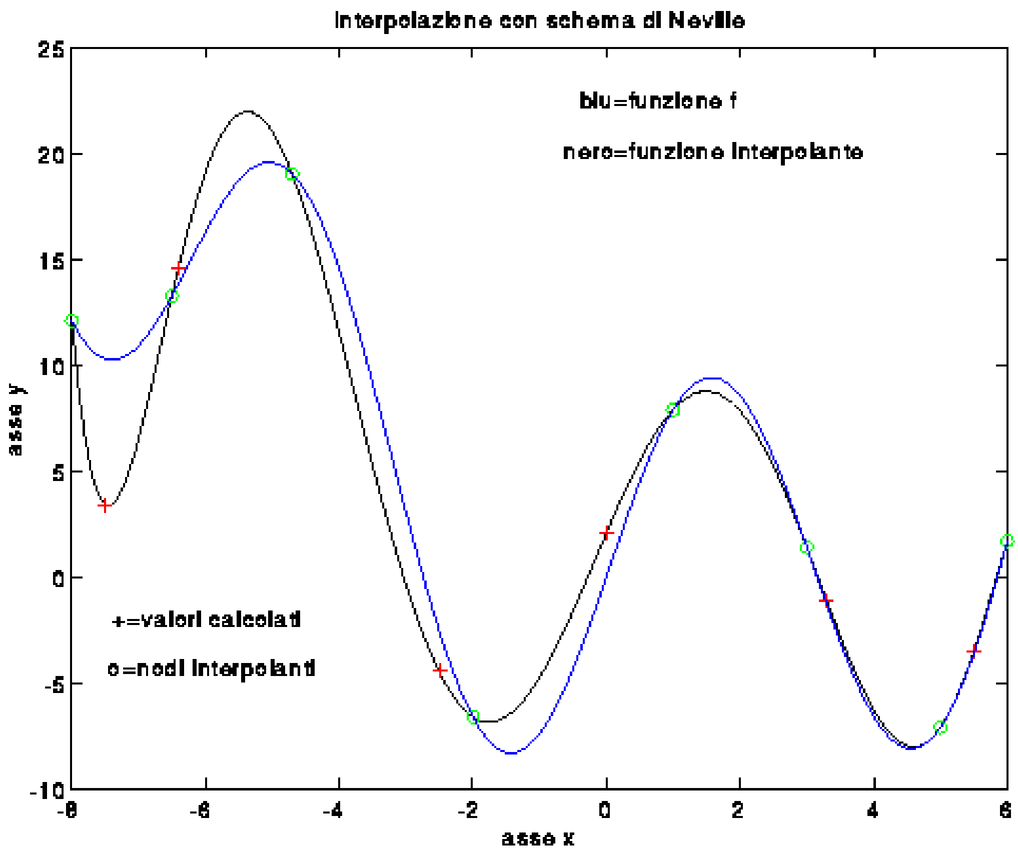
x=	y=
5.5000	-3.4996
-7.5000	3.4252
0	2.0756
3.3000	-1.1198
-2.5000	-4.4667
-6.4000	14.6329



Se vuoi ricominciare premi "0" : 0

Scegli il tipo di interpolazione :
1. Interpolazione di Lagrange
2. Interpolazione di Newton
3. Utilizzando lo schema di Neville
4. Interpolazione di Hermite
5. Interpolazione con spline lineari
6. Interpolazione con spline cubiche naturali
7. Interpolazione con spline cubiche complete
8. Approssimazione polinomiale ai minimi quadrati
Introduci il numero corrispondente : 3
Introduci :
- il vettore colonna dei nodi interpolanti : 6
I nodi devono essere almeno 2.Introduci nuovamente :
- il vettore colonna dei nodi interpolanti : [-2;5;1;3;-8;-4.7;6;-6.5]
- il vettore colonna dei corrispondenti valori della funzione : [-6.59;-7.09;7.91;1.41]
ATTENZIONE : deve avere le stesse dimensioni dell'altro vettore.RICOMINCIA
Introduci :
- il vettore colonna dei corrispondenti valori della funzione : [-6.59;-7.09;7.91;1.41;12.11;19.05;1.7;13.29]
Se conosci la funzione f corrispondente ai punti introdotti premere "1" : 1
Introduci :
- Il nome della function contenente la funzione : funz
- il vettore colonna con i punti nei quali calcolare la funzione interpolante : [5.5;-7.5;0;3.3;-2.5;-6.4]
Il vettore soluzione y corrispondente ai punti x immessi é :

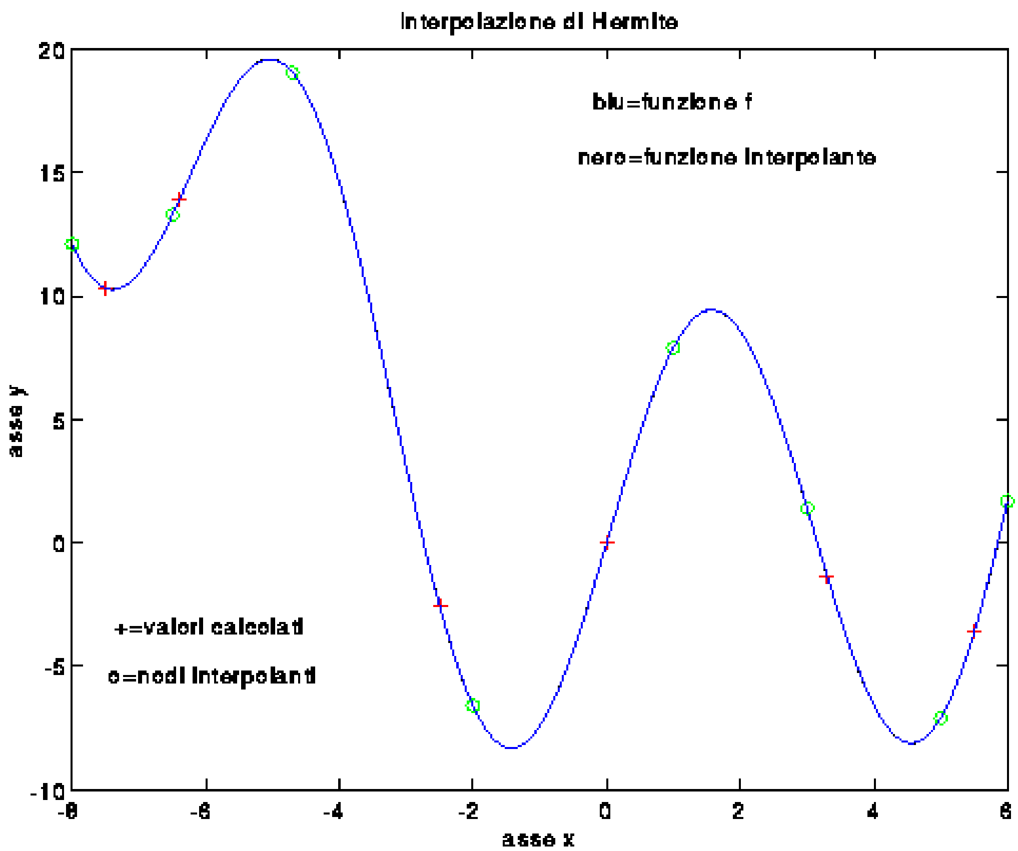
x=	y=
5.5000	-3.4996
-7.5000	3.4252
0	2.0756
3.3000	-1.1198
-2.5000	-4.4667
-6.4000	14.6329



Se vuoi ricominciare premi "0" :0

Scegli il tipo di interpolazione :
1. Interpolazione di Lagrange
2. Interpolazione di Newton
3. Utilizzando lo schema di Neville
4. Interpolazione di Hermite
5. Interpolazione con spline lineari
6. Interpolazione con spline cubiche naturali
7. Interpolazione con spline cubiche complete
8. Approssimazione polinomiale ai minimi quadrati
Introduci il numero corrispondente : 9
Il valore introdotto é errato.Introducilo di nuovo : 4
Introduci :
- il vettore colonna dei nodi interpolanti : [-2;5;1;3;-8;-4.7;6;-6.5]
- il vettore colonna dei corrispondenti valori della funzione : [-6.59;-7.09;7.91;1.41;12.11;19.05;1.7;13.29]
- il vettore colonna con corrispondenti valori delle derivate prime : [-5.91;4.58;5.15;-9.15]
ATTENZIONE : deve avere le stesse dimensioni dei primi 2 vettori.RICOMINCIA
Introduci :
- il vettore colonna con corrispondenti valori delle derivate prime : [-5.91;4.58;5.15;-9.15;-6.20;-3.22;11.85;5.76]
Se conosci la funzione f corrispondente ai punti introdotti premere "1" : 1
Introduci :
- Il nome della function contenente la funzione : funz
- il vettore colonna con i punti nei quali calcolare la funzione interpolante : [5.5;-7.5;0;3.3;-2.5;-6.4]
Il vettore soluzione y corrispondente ai punti x immessi é :

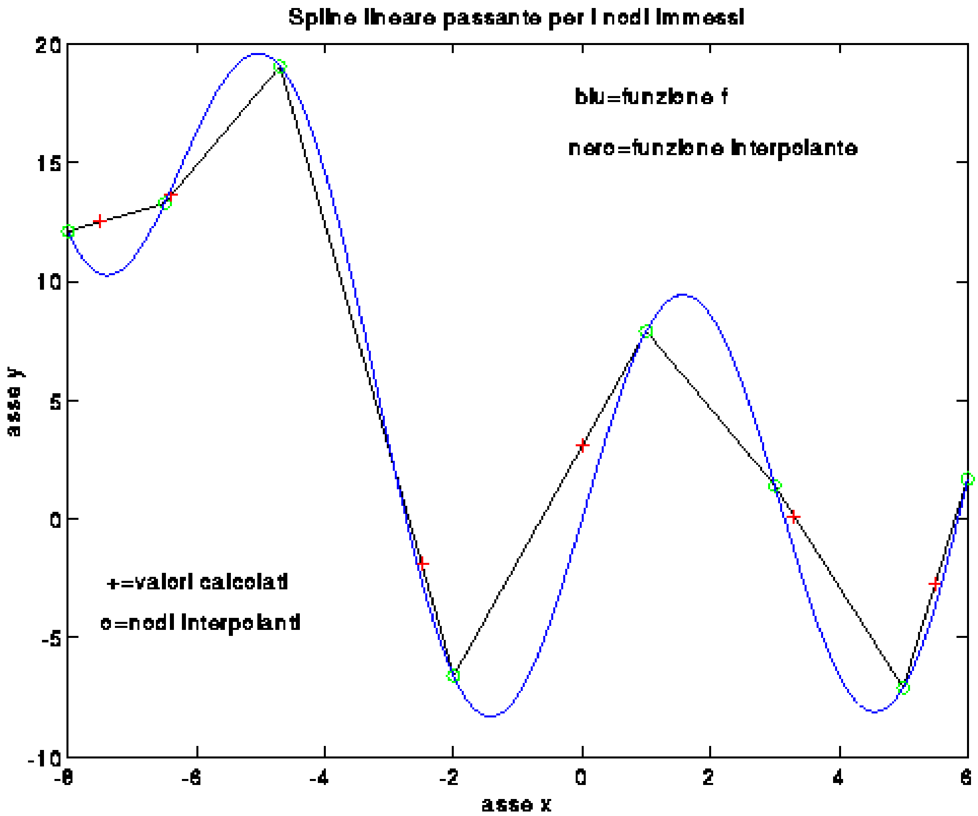
x=	y=	
5.5000	-3.6228	←------ ATTENZIONE : i due grafici coincidono
-7.5000	10.3177	
0	0.0016	
3.3000	-1.3313	
-2.5000	-2.5442	
-6.4000	13.8777	



Se vuoi ricominciare premi "0" : 0

Scegli il tipo di interpolazione :
1. Interpolazione di Lagrange
2. Interpolazione di Newton
3. Utilizzando lo schema di Neville
4. Interpolazione di Hermite
5. Interpolazione con spline lineari
6. Interpolazione con spline cubiche naturali
7. Interpolazione con spline cubiche complete
8. Approssimazione polinomiale ai minimi quadrati
Introduci il numero corrispondente : 5
Introduci :
- il vettore colonna dei nodi interpolanti : [5.5,-7.5;0,3]
Non é stato introdotto un vettore colonna.Ricomincia
Introduci :
- il vettore colonna dei nodi interpolanti : [-2;5;1;3;-8;-4.7;6;-6.5]
- il vettore colonna dei corrispondenti valori della funzione : [-6.59;-7.09;7.91;1.41;12.11;19.05;1.7;13.29]
Se conosci la funzione f corrispondente ai punti introdotti premere "1" : 1
Introduci :
- Il nome della function contenente la funzione : funz
- il vettore colonna con i punti nei quali calcolare la funzione interpolante : [5.5;-7.5;0;3.3;-2.5;-6.4]
Il vettore soluzione y corrispondente ai punti x immessi é :

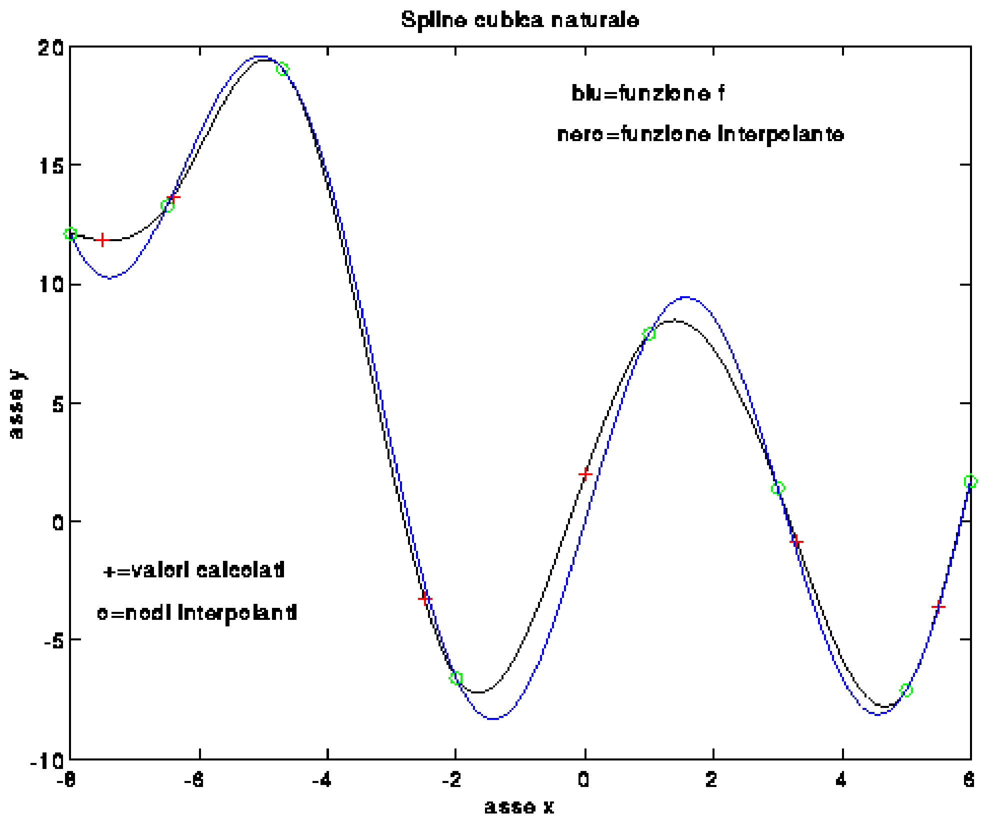
x=	y=
5.5000	-2.6950
-7.5000	12.5033
0	3.0767
3.3000	0.1350
-2.5000	-1.8419
-6.4000	13.6100



Se vuoi ricominciare premi "0" : 0

Scegli il tipo di interpolazione :
1. Interpolazione di Lagrange
2. Interpolazione di Newton
3. Utilizzando lo schema di Neville
4. Interpolazione di Hermite
5. Interpolazione con spline lineari
6. Interpolazione con spline cubiche naturali
7. Interpolazione con spline cubiche complete
8. Approssimazione polinomiale ai minimi quadrati
Introduci il numero corrispondente : 6
Introduci :
- il vettore colonna dei nodi interpolanti : [-2;5;1;3;-8;-4.7;6;-6.5]
- il vettore colonna dei corrispondenti valori della funzione : [-6.59;-7.09;7.91;1.41;12.11;19.05;1.7;13.29]
Se conosci la funzione f corrispondente ai punti introdotti premere "1" : 1
Introduci :
- Il nome della function contenente la funzione : funz
- il vettore colonna con i punti nei quali calcolare la funzione interpolante : [5.5;-7.5;0;3.3;-2.5;-6.4]
Il vettore soluzione y corrispondente ai punti x immessi é :

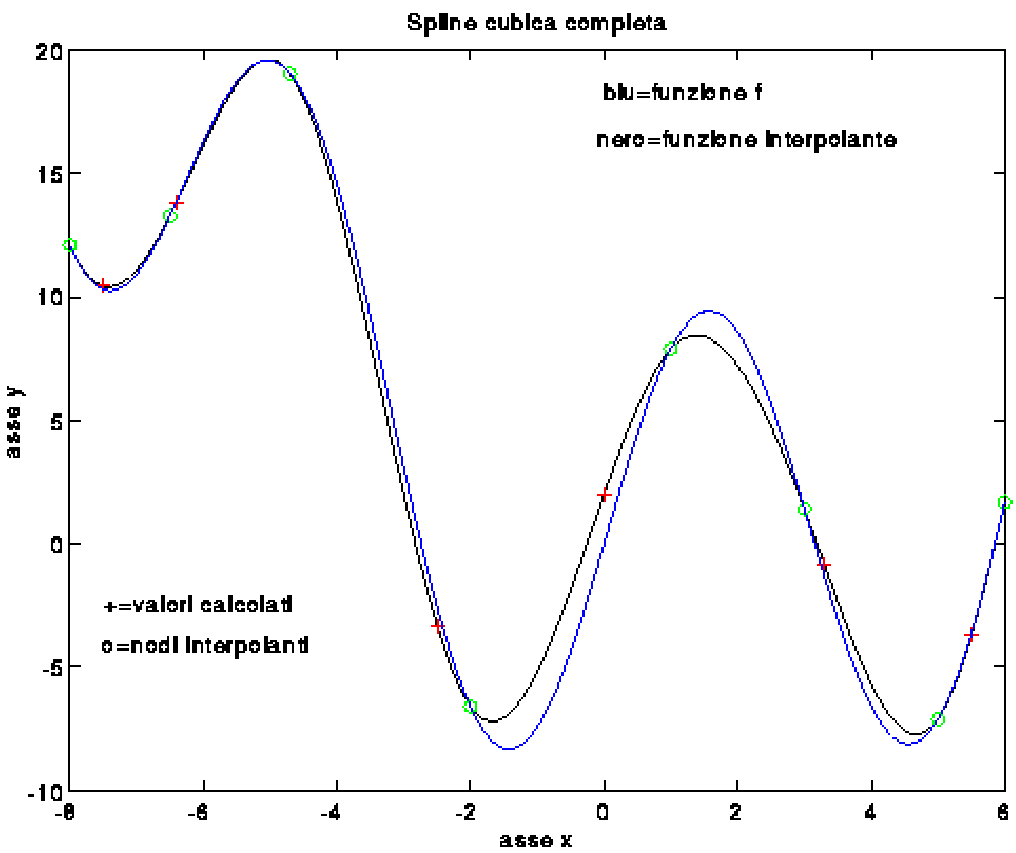
x=	y=
5.5000	-3.5539
-7.5000	11.8499
0	1.9451
3.3000	-0.8767
-2.5000	-3.2063
-6.4000	13.6903



Se vuoi ricominciare premi "0" : 0

Scegli il tipo di interpolazione :
1. Interpolazione di Lagrange
2. Interpolazione di Newton
3. Utilizzando lo schema di Neville
4. Interpolazione di Hermite
5. Interpolazione con spline lineari
6. Interpolazione con spline cubiche naturali
7. Interpolazione con spline cubiche complete
8. Approssimazione polinomiale ai minimi quadrati
Introduci il numero corrispondente : 7
Introduci :
- il vettore colonna dei nodi interpolanti : [-2;5;1;3;-8;-4.7;6;-6.5]
- il vettore colonna dei corrispondenti valori della funzione : [-6.59;-7.09;7.91;1.41;12.11;19.05;1.7;13.29]
Se conosci la funzione f corrispondente ai punti introdotti premere "1" : 1
Introduci :
- Il nome della function contenente la funzione : funz
- il vettore colonna con i punti nei quali calcolare la funzione interpolante : [5.5;-7.5;0;3.3;-2.5;-6.4]
- il valore della derivata prima nel nodo più piccolo fissato :-6.2
- il valore della derivata prima nel nodo più grande fissato : 11.85
Il vettore soluzione y corrispondente ai punti x immessi é :

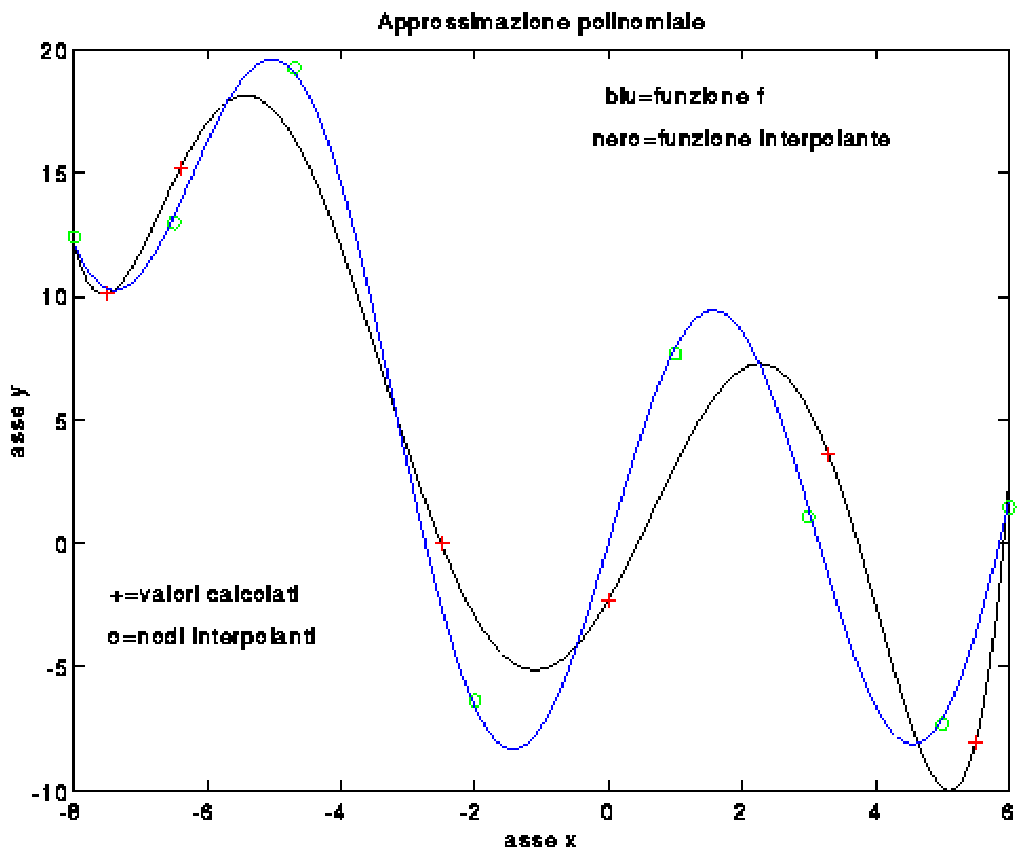
x=	y=
5.5000	-3.6839
-7.5000	10.4438
0	2.0022
3.3000	-0.8489
-2.5000	-3.2946
-6.4000	13.8377



Se vuoi ricominciare premi "0" : 0

Scegli il tipo di interpolazione :
1. Interpolazione di Lagrange
2. Interpolazione di Newton
3. Utilizzando lo schema di Neville
4. Interpolazione di Hermite
5. Interpolazione con spline lineari
6. Interpolazione con spline cubiche naturali
7. Interpolazione con spline cubiche complete
8. Approssimazione polinomiale ai minimi quadrati
Introduci il numero corrispondente : 8
Introduci :
- il vettore colonna dei nodi interpolanti : [-2;5;1;3;-8;-4.7;6;-6.5]
- il vettore colonna dei corrispondenti valori della funzione : [-6.35;-7.3;7.68;1.1;12.4;19.25;1.48;13]
Se conosci la funzione f corrispondente ai punti introdotti premere "1" : 1
Introduci :
- Il nome della function contenente la funzione : funz
- il vettore colonna con i punti nei quali calcolare la funzione interpolante : [5.5;-7.5;0;3.3;-2.5;-6.4]
- Il grado del polinomio interpolante (minore del numero di nodi introdotto) : 7
Il vettore soluzione y corrispondente ai punti x immessi é :

x=	y=
5.5000	-8.0660
-7.5000	10.1192
0	-2.2765
3.3000	3.6124
-2.5000	-0.0247
-6.4000	15.2310



Se vuoi ricominciare premi "0" : 6

diary off

```
zero
  CALCOLO DELLO ZERO DI UNA FUNZIONE

Scegliere tra i seguenti metodi di calcolo dello zero di una funzione:

1. Metodo di bisezione
2. Metodo di Newton-Rapson
3. Metodo delle secanti a due punti
4. Metodo di Newton a doppio passo per il calcolo di tutti gli zeri reali di un generico polinomio
Introduci il numero corrispondente : 1
INTRODUCI:
- Il nome della function contenente la funzione : funz
La funzione non esiste, introduci :
- Il nome della function contenente la funzione : funzione
- L'errore massimo che si può commettere (massimo 0.1) : 10
Il valore dell'errore massimo introdotto non é corretto
Introducilo nuovamente : 0.00001
- L'estremo inferiore dell'intervallo : 1
- L'estremo superiore dell'intervallo tale che f(a)*f(b)<0 : 2.8
Il metodo non è applicabile .
Infatti agli estremi la funzione assume lo stesso segno.


Scegli un altro metodo :

1. Metodo di bisezione
2. Metodo di Newton-Rapson
3. Metodo delle secanti a due punti
4. Metodo di Newton a doppio passo per il calcolo di tutti gli zeri reali di un generico polinomio
Introduci il numero corrispondente : 1
INTRODUCI:
- Il nome della function contenente la funzione : funzione
- L'errore massimo che si può commettere (massimo 0.1) : 0.00001
- L'estremo inferiore dell'intervallo : 0
- L'estremo superiore dell'intervallo tale che f(a)*f(b)<0 : 3

Lo zero è :
     0

Se vuoi ricominciare nuovamente premi il tasto '1' :1


Scegli di nuovo tra :

1. Metodo di bisezione
2. Metodo di Newton-Rapson
3. Metodo delle secanti a due punti
4. Metodo di Newton a doppio passo per il calcolo di tutti gli zeri reali di un generico polinomio
Introduci il numero corrispondente : 1
INTRODUCI:
- Il nome della function contenente la funzione : funzione
- L'errore massimo che si può commettere (massimo 0.1) : 0.00001
- L'estremo inferiore dell'intervallo : 1
- L'estremo superiore dell'intervallo tale che f(a)*f(b)<0 : 4

Lo zero é approssivamente:
    2.9274

Se vuoi ricominciare nuovamente premi il tasto '1' :1
```

Scegli di nuovo tra :
1. Metodo di bisezione
2. Metodo di Newton-Rapson
3. Metodo delle secanti a due punti
4. Metodo di Newton a doppio passo per il calcolo di tutti gli zeri reali di un generico polinomio
Introduci il numero corrispondente : 6
Scelta errata
Ricomincia

1. Metodo di bisezione
2. Metodo di Newton-Rapson
3. Metodo delle secanti a due punti
4. Metodo di Newton a doppio passo per il calcolo di tutti gli zeri reali di un generico polinomio
Introduci il numero corrispondente : 1
INTRODUCI:
- Il nome della function contenente la funzione : funzione
- L'errore massimo che si può commettere (massimo 0.1) : 0.00001
- L'estremo inferiore dell'intervallo : 1
- L'estremo superiore dell'intervallo tale che f(a)*f(b)<0 : 1
L'estremo inferiore deve essere strettamente minore di quello superiore
Ricomincia dall'inizio .

Scegli nuovamente il metodo di calcolo :
1. Metodo di bisezione
2. Metodo di Newton-Rapson
3. Metodo delle secanti a due punti
4. Metodo di Newton a doppio passo per il calcolo di tutti gli zeri reali di un generico polinomio
Introduci il numero corrispondente : 2
INTRODUCI:
- Il nome della function contenente la funzione : funzione
- L'errore massimo che si può commettere (massimo 0.1) : 0.00001
- Il nome della function contenente la derivata prima della funzione : derivata
- L'approssimazione iniziale dello zero da calcolare :-5

Lo zero é approssivamente:
x0 =

 -1.4867e-009

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli di nuovo tra :
1. Metodo di bisezione
2. Metodo di Newton-Rapson
3. Metodo delle secanti a due punti
4. Metodo di Newton a doppio passo per il calcolo di tutti gli zeri reali di un genericopolinomio
Introduci il numero corrispondente : 2
INTRODUCI:
- Il nome della function contenente la funzione : funzione
- L'errore massimo che si può commettere (massimo 0.1) : 0.00001
- Il nome della function contenente la derivata prima della funzione: derivata
- L'approssimazione iniziale dello zero da calcolare : 0

Lo zero é :
x0 =

     0

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli di nuovo tra :

1. Metodo di bisezione
2. Metodo di Newton-Rapson
3. Metodo delle secanti a due punti
4. Metodo di Newton a doppio passo per il calcolo di tutti gli zeri reali di un generico polinomio
Introduci il numero corrispondente : 2
INTRODUCI:
- Il nome della function contenente la funzione : funzione
- L'errore massimo che si può commettere (massimo 0.1) : 0.00001
- Il nome della function contenente la derivata prima della funzione : derivata
- L'approssimazione iniziale dello zero da calcolare : 4

Lo zero é approssivamente:

x0 =

    2.9274

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli di nuovo tra :

1. Metodo di bisezione
2. Metodo di Newton-Rapson
3. Metodo delle secanti a due punti
4. Metodo di Newton a doppio passo per il calcolo di tutti gli zeri reali di un generico polinomio
Introduci il numero corrispondente : 3
INTRODUCI:
- Il nome della function contenente la funzione : funzione
- L'errore massimo che si può commettere (massimo 0.1) : 0.00001
- La prima approssimazione iniziale dello zero da calcolare :-1
- La seconda approssimazione iniziale dello zero da calcolare : -2

Lo zero é approssivamente:
 -1.5417e-011

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli di nuovo tra :

1. Metodo di bisezione
2. Metodo di Newton-Rapson
3. Metodo delle secanti a due punti
4. Metodo di Newton a doppio passo per il calcolo di tutti gli zeri reali di un generico polinomio
Introduci il numero corrispondente : 3
INTRODUCI:
- Il nome della function contenente la funzione : funzione
- L'errore massimo che si può commettere (massimo 0.1) : 0.00001
- La prima approssimazione iniziale dello zero da calcolare : 1
- La seconda approssimazione iniziale dello zero da calcolare : 4

Lo zero é approssivamente:
    2.9274

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli di nuovo tra :

1. Metodo di bisezione
2. Metodo di Newton-Rapson
3. Metodo delle secanti a due punti
4. Metodo di Newton a doppio passo per il calcolo di tutti gli zeri reali di un generico polinomio
Introduci il numero corrispondente : 4

Scegli tra :

1. - Introdurre a priori il numero di zeri reali del polinomio interessato
2. - Applicare altri metodi
Introduci il numero corrispondente : 1
Introduci :
- il vettore dei coefficienti del polinomio a partire dal termine
  noto fino ad arrivare a quello di grado massimo : 4
  Il vettore deve avere almeno 2 elementi.Inseriscili : [0 9;8-1]
  Non é stato introdotto un vettore.Inseriscilo : [0 0 0 0 0 0]
  E'stato introdotto un vettore nullo.Inseriscilo nuovamente : [0 0 0 6 7 8 0 0 0 0]
Introduci :
- il numero di zeri reali del polinomio (nel caso di polinomi di secondo grado
  verranno calcolati tutti gli zeri compreso quelli complessi) : 3
- l'errore massimo di approssimazione (al massimo 0.1): 0
  Valore errato.Inserisci quello giusto : 0.00001

Gli zeri del polinomio sono :
        0
        0
        0
  -0.4375 + 0.7474i
  -0.4375 + 0.7474i

OSS.:in questo caso é stato possibile calcolare anche gli zeri complessi

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli di nuovo tra :

1. Metodo di bisezione
2. Metodo di Newton-Rapson
3. Metodo delle secanti a due punti
4. Metodo di Newton a doppio passo per il calcolo di tutti gli zeri reali di un generico polinomio
Introduci il numero corrispondente : 4

Scegli tra :

1. - Introdurre a priori il numero di zeri reali del polinomio interessato
2. - Applicare altri metodi
Introduci il numero corrispondente : 1
Introduci :
- il vettore dei coefficienti del polinomio a partire dal termine
  noto fino ad arrivare a quello di grado massimo : [0 6 -14 13 -9 2 6 -5 1]
- il numero di zeri reali del polinomio (nel caso di polinomi di secondo grado
  verranno calcolati tutti gli zeri compreso quelli complessi) :-1.5
  Valore errato.Inserisci quello giusto : 6
Introduci :
- l'errore massimo di approssimazione (al massimo 0.1): 0.00001

Gli zeri del polinomio sono :
0
3.0000
1.4142
1.0000
1.0000
-1.4142

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli di nuovo tra :
1. Metodo di bisezione
2. Metodo di Newton-Rapson
3. Metodo delle secanti a due punti
4. Metodo di Newton a doppio passo per il calcolo di tutti gli zeri reali di un generico polinomio

Introduci il numero corrispondente : 4

Scegli tra :
1. - Introdurre a priori il numero di zeri reali del polinomio interessato
2. - Applicare altri metodi

Introduci il numero corrispondente : 1
Introduci :
- il vettore dei coefficienti del polinomio a partire dal termine
  noto fino ad arrivare a quello di grado massimo : [229-77 -37 4 1]
- il numero di zeri reali del polinomio (nel caso di polinomi di secondo grado
  verranno calcolati tutti gli zeri compreso quelli complessi) : 5
  il numero di zeri non può superare il grado del polinomio
  Inserisci quello giusto : 4

Introduci :
- l'errore massimo di approssimazione (al massimo 0.1): .00001

Gli zeri del polinomio sono :
1.8294 <---Attenzione : questi sono proprio gli autovalori della matrice
4.8668 triangolare simmetrica immessa nella
-3.6506 risoluzione del programma Autoval.m .
-7.0456 Infatti questo é il suo polinomio caratteristico

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli di nuovo tra :
1. Metodo di bisezione
2. Metodo di Newton-Rapson
3. Metodo delle secanti a due punti
4. Metodo di Newton a doppio passo per il calcolo di tutti gli zeri reali di un generico polinomio

Introduci il numero corrispondente : 4

Scegli tra :
1. - Introdurre a priori il numero di zeri reali del polinomio interessato
2. - Applicare altri metodi

Introduci il numero corrispondente : 1
Introduci :
- il vettore dei coefficienti del polinomio a partire dal termine
  noto fino ad arrivare a quello di grado massimo : [3 2]
- il numero di zeri reali del polinomio (nel caso di polinomi di secondo grado
  verranno calcolati tutti gli zeri compreso quelli complessi) : 1
- l'errore massimo di approssimazione (al massimo 0.1): 0.1

Gli zeri del polinomio sono :
-1.5000

Se vuoi ricominciare nuovamente premi il tasto'1' :1

Scegliere di nuovo tra :
1. Metodo di bisezione
2. Metodo di Newton-Rapson
3. Metodo delle secanti a due punti
4. Metodo di Newton a doppio passo per il calcolo di tutti gli zeri reali di un generico polinomio
Introduci il numero corrispondente :4

Scegli tra :

1. - Introdurre a priori il numero di zeri reali del polinomio interessato
2. - Applicare altri metodi
Introduci il numero corrispondente : 1
Introduci :
- il vettore dei coefficienti del polinomio a partire dal termine
  noto fino ad arrivare a quello di grado massimo : [1 0 0]
- il numero di zeri reali del polinomio (nel caso di polinomi di secondo grado
  verranno calcolati tutti gli zeri compreso quelli complessi) : 0
- l'errore massimo di approssimazione (al massimo 0.1): 0.1

Il polinomio introdotto ha grado zero

Se vuoi ricominciare nuovamente premi il tasto '1' :1

Scegli di nuovo tra :
1. Metodo di bisezione
2. Metodo di Newton-Rapson
3. Metodo delle secanti a due punti
4. Metodo di Newton a doppio passo per il calcolo di tutti gli zeri reali di un generico polinomio
Introduci il numero corrispondente : 4

Scegli tra :
1. - Introdurre a priori il numero di zeri reali del polinomio interessato
2. - Applicare altri metodi
Introduci il numero corrispondente : 1
Introduci :
- il vettore dei coefficienti del polinomio a partire dal termine
  noto fino ad arrivare a quello di grado massimo : [1 0 0 0 1]
- il numero di zeri reali del polinomio (nel caso di polinomi di secondo grado
  verranno calcolati tutti gli zeri compreso quelli complessi) : 0
- l'errore massimo di approssimazione (al massimo 0.1): 0.1

Il polinomio non ha zeri reali

Se vuoi ricominciare nuovamente premi il tasto '1' :0

diary off

```
formquad
  FORMULE DI QUADRATURA

INTRODUCI:
- Il nome della function contenente la funzione : funz
La funzione introdotta non esiste,introduci :
- Il nome della function contenente la funzione : derivata
- L'estremo inferiore di integrazione : 0
- L'estremo superiore di integrazione : 0

L'integrale definito é nullo

Se vuoi ricominciare nuovamente premi il tasto '0' :0
INTRODUCI:
- Il nome della function contenente la funzione : derivata
- L'estremo inferiore di integrazione : 0
- L'estremo superiore di integrazione : pi/2
- Il numero di suddivisioni dell'intervallo di integrazione :30
Scegliere tra le seguenti formule di quadratura :
1. Trapezi composta
2. Simpson composta
Introduci il numero corrispondente : 1
T =

   -1.2452

Se vuoi ricominciare nuovamente premi il tasto '0' :0

INTRODUCI:
- Il nome della function contenente la funzione : derivata
- L'estremo inferiore di integrazione : 0
- L'estremo superiore di integrazione : pi/2
- Il numero di suddivisioni dell'intervallo di integrazione :30
Scegliere tra le seguenti formule di quadratura :
1. Trapezi composta
2. Simpson composta
Introduci il numero corrispondente : 2
T =

   -1.2450

Se vuoi ricominciare nuovamente premi il tasto '0' :0

INTRODUCI:
- Il nome della function contenente la funzione : derivata
- L'estremo inferiore di integrazione : pi/2
- L'estremo superiore di integrazione : 0
- Il numero di suddivisioni dell'intervallo di integrazione :30
Scegliere tra le seguenti formule di quadratura :
1. Trapezi composta
2. Simpson composta
Introduci il numero corrispondente : 2
T =

    1.2450

Se vuoi ricominciare nuovamente premi il tasto '0' :1
diary off
```

infatti l'integrale tra 0 e $\pi / 2$ della funzione : $\cos x + 2x - 3$ è $1 + (\pi / 2)^2 - 3 (\pi / 2) = -1.2450$

```
Metmulst

  METODI MULTISTEP (ONE-STEP) PER EQUAZIONI DIFFERENZIALI

INTRODUCI:
- Il nome della function contenente la funzione f(t,y): fun
- L'estremo inferiore dell'intervallo di definizione : 0
- L'estremo superiore dell'intervallo di definizione : 0.3
- La condizione iniziale corrispondente all'estremo inferiore dell'intervallo : 0
- Il numero di suddivisioni dell'intervallo di definizione :20
Scegliere tra :
1. Metodo di Eulero implicito
2. Metodo di Eulero esplicito
3. Metodo dei trapezi
Introduci il numero corrispondente : 1
Introduci il nome della function contenente la derivata prima della funzione rispetto a y : fund
Introduci l'errore massimo che si può commettere (massimo 0.1) : 0.0001
In corrispondenza dei seguenti punti t
La funzione soluzione dell'equazione differenziale assume i valori y che sono
t=        y=
   0         0
   0.0150    0.0002
   0.0300    0.0007
   0.0450    0.0013
   0.0600    0.0022
   0.0750    0.0033
   0.0900    0.0047
   0.1050    0.0062
   0.1200    0.0080
   0.1350    0.0099
   0.1500    0.0121
   0.1650    0.0144
   0.1800    0.0169
   0.1950    0.0196
   0.2100    0.0225
   0.2250    0.0255
   0.2400    0.0287
   0.2550    0.0321
   0.2700    0.0355
   0.2850    0.0391
   0.3000    0.0429

Se vuoi ricominciare nuovamente premi il tasto '0' :0
INTRODUCI:
- Il nome della function contenente la funzione f(t,y): fun
- L'estremo inferiore dell'intervallo di definizione : 0
- L'estremo superiore dell'intervallo di definizione : 0.3
- La condizione iniziale corrispondente all'estremo inferiore dell'intervallo : 0
- Il numero di suddivisioni dell'intervallo di definizione :20
Scegliere tra :
1. Metodo di Eulero implicito
2. Metodo di Eulero esplicito
3. Metodo dei trapezi
Introduci il numero corrispondente : 2
In corrispondenza dei seguenti punti t
La funzione soluzione dell'equazione differenziale assume i valori y che sono
t=        y=
   0         0
   0.0150    0
   0.0300    0.0002
```

```
    0.0450    0.0007
    0.0600    0.0013
    0.0750    0.0022
    0.0900    0.0034
    0.1050    0.0047
    0.1200    0.0062
    0.1350    0.0080
    0.1500    0.0099
    0.1650    0.0121
    0.1800    0.0144
    0.1950    0.0170
    0.2100    0.0197
    0.2250    0.0226
    0.2400    0.0256
    0.2550    0.0289
    0.2700    0.0322
    0.2850    0.0357
    0.3000    0.0394

Se vuoi ricominciare nuovamente premi il tasto '0' :0
INTRODUCI:
- Il nome della function contenente la funzione f(t,y): fun
- L'estremo inferiore dell'intervallo di definizione : 0
- L'estremo superiore dell'intervallo di definizione : 0.3
- La condizione iniziale corrispondente all'estremo inferiore dell'intervallo : 0
- Il numero di suddivisioni dell'intervallo di definizione :20
Scegliere tra :
1. Metodo di Eulero implicito
2. Metodo di Eulero esplicito
3. Metodo dei trapezi
Introduci il numero corrispondente : 3
Introduci il nome della function contenente la derivata prima della funzione rispetto a y : fund
Introduci l'errore massimo che si può commettere (massimo 0.1) : 0.0001
In corrispondenza dei seguenti punti t
La funzione soluzione dell'equazione differenziale assume i valori y che sono
t=         y=
     0         0
    0.0150    0.0001        <---------------------------------- Questi valori sono uguali a quelli teorici fino alla
    0.0300    0.0004                                              cifra decimale visualizzata
    0.0450    0.0010
    0.0600    0.0018
    0.0750    0.0028
    0.0900    0.0040
    0.1050    0.0055
    0.1200    0.0071
    0.1350    0.0089
    0.1500    0.0110
    0.1650    0.0132
    0.1800    0.0157
    0.1950    0.0183
    0.2100    0.0211
    0.2250    0.0241
    0.2400    0.0272
    0.2550    0.0305
    0.2700    0.0339
    0.2850    0.0374
    0.3000    0.0411

Se vuoi ricominciare nuovamente premi il tasto '0' :9
diary off
```