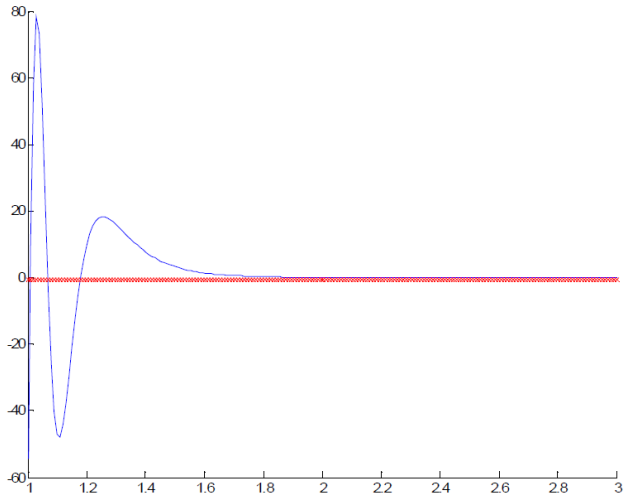
ISSN 2239-7035

dicecca.net

Anno I - № 12 – Supplemento al MONITORE NAPOLETANO - № 154 – Dicembre 2020

GIOVANNI DI CECCA
VIRGINIA BELLINO



Progetto Metodo di Simpson Fisso e Globale

dicecca.net
web site

Direttore Responsabile: Giovanni Di Cecca

Collana dicecca.net – Computer Science

Anno I - № 12 – Supplemento al Numero 154 – Dicembre 2020

Periodico Mensile Registrato presso il Tribunale di Napoli № 45 dell'8 giugno 2011

ISSN: 2239-7035

Indice


Simpson
a schema fisso

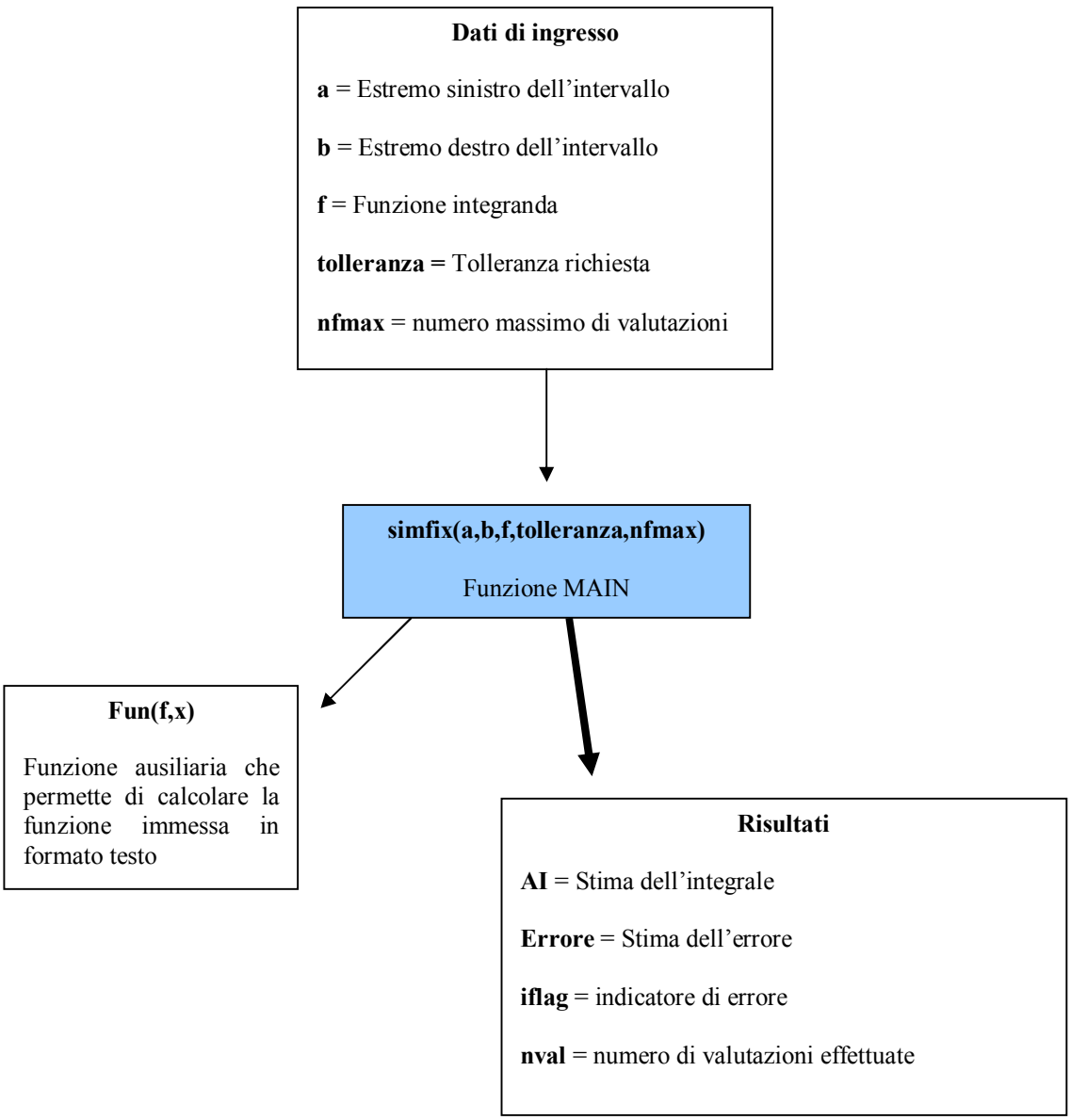
Sezione 1

Documentazione esterna

Progetto Metodo di Simpson a schema fisso

$$\int_a^b f(x)\,dx$$

Schema grafico



- **Scopo:** calcolare il valore numerico approssimato di un integrale definito del tipo

$$\int_a^b f(x)\,dx$$

utilizzando la strategia basata sulla *formula di quadratura composta di Simpson a schema fisso*.

- **Specifiche:**

[AI,Errore,iflag,nval]=simfix(a,b,f,tolleranza,nfmax)

- **Descrizione:**

l'algoritmo esegue i seguenti passi:

1. Inizializzazione delle variabili.

2. Valutazione della funzione negli estremi a et b, e salvataggio di tali valori che verranno poi riutilizzati in seguito.
   (nota: tutte le valutazioni di funzione vengono eseguite richiamando il modulo ausiliario Fun)

3. Suddivisione dell'intervallo di integrazione (a,b) in due sottointervalli e calcolo della prima stima dell'integrale usando la formula di Simpson semplice su 3 punti.

4. Inizio del ciclo while che da luogo al procedimento di Simpson a schema fisso. Tale ciclo termina o quando si raggiunge la tolleranza richiesta, oppure quando si raggiunge il numero massimo di valutazioni di funzione consentito.

5. Quando il ciclo termina, vi è un controllo per stabilire quale delle due suddette condizioni si è verificata, in modo da restituire in output il corrispettivo valore dell'indicatore iflag.

6. Infine, vi sono le istruzioni per eseguire il plot della funzione.

♦ Riferimenti bibliografici:

- James F. Epperson

INTRODUZIONE ALL'ANALISI NUMERICA
McGraw-Hill

- Prof. Eleonora Messina

Appunti del corso di calcolo numerico
A. A. 2004/2005

♦ Lista dei parametri:

Parametri di input:

a = Estremo sinistro dell'intervallo di integrazione
b = Estremo destro dell'intervallo di integrazione
f = Funzione integranda scritta nel seguente modo:

```
f='(100./(x.^7)).*sin(10./(x.^7))'
```

tolleranza = Tolleranza richiesta
nfmax = Massimo numero consentito di valutazioni della funzione

Parametri di output:

AI = Approssimazione dell'integrale
Errore = Stima dell'errore
iflag = Indicatore di errore:

0 Se la condizione di uscita si è verificata per il raggiungimento della tolleranza richiesta;

1 Se la condizione di uscita si è verificata per il raggiungimento del numero massimo di valutazioni;

nval = Numero di valutazioni della funzione effettuato

- **Indicatori di errore:**

iflag: indica se durante la procedura si verificano degli errori,bloccando in tal caso l'esecuzione. Può assumere i seguenti valori:

0 Se la condizione di uscita si è verificata per il raggiungimento della tolleranza richiesta;

1 Se la condizione di uscita si è verificata per il raggiungimento del numero massimo di valutazioni;

- **Funzioni ausiliarie:**

funzione Fun:
questa routine calcola i valori della funzione data in input secondo Matlab.
La specifica è:
function y=Fun(f, x)

- **Complessità computazionale:**

Complessità di tempo: dipende dalla complessità della funzione e dal numero di valutazioni

Complessità di spazio:

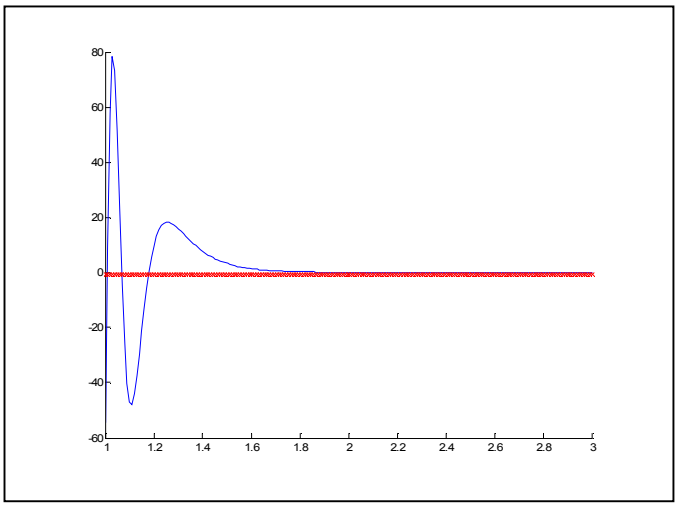
```
EDU>>
EDU>> whos
  Name          Size        Bytes  Class

  AI            1x1             8  double array
  Errore        1x1             8  double array
  a             1x1             8  double array
  b             1x1             8  double array
  fun           1x30           60  char array
  iflag         1x1             8  double array (logical)
  nfmax         1x1             8  double array
  nval          1x1             8  double array
  tolleranza    1x1             8  double array

Grand total is 38 elements using 124 bytes
```

- **Accuratezza fornita:**

l'accuratezza del risultato dipende dalla tolleranza richiesta,che viene fornita in input.

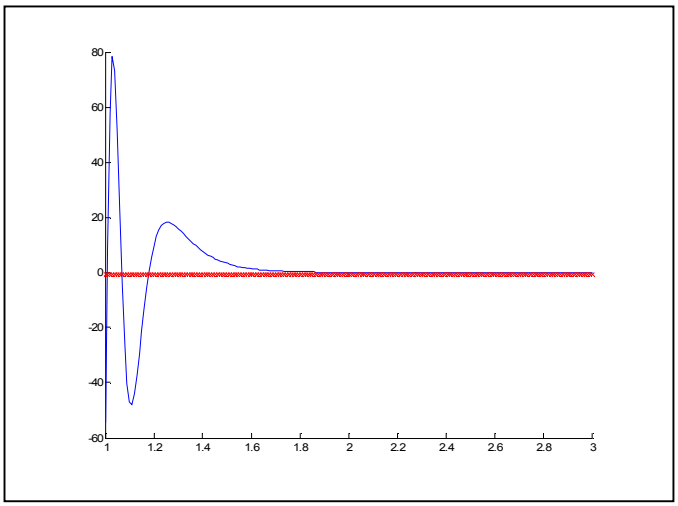
♦ Esempio d'uso:

```
fun =

(100./(x.^7)).*sin(10./(x.^7))


a =

    1


b =

    3


nfmax =

       1000


tolleranza =

   0.00100000000000

soluzioni della simfix

AI =

   3.34836501062069


Errore =

    8.794999028849541e-004


iflag =

     0


nval =

   257
```



- **Esempio d'uso:**

```
Primo TEST

fun =

(100./(x.^7)).*sin(10./(x.^7))


a =

     1


b =

     3


nfmax =

   100


tolleranza =

   0.00100000000000

soluzioni della simfix

AI =

   3.36155750916396


Errore =

   0.01202705155935


iflag =

     1


nval =

   129
```



- **Raccomandazioni d'uso:** I file sono stati implementati su **MATLAB Student Version 6.0**. Durante la fase di test su MATLAB 5.2 si sono verificati degli inconvenienti su alcuni plot. **Si raccomanda l'uso di MATLAB 6.0 o superiore.** Inoltre digitando **help simfix** si ha un rapido help sull'uso del file.

Sezione 2

Codice MATLAB

```
%            Progetto Metodo di Simpson
%                  a schema fisso
%
%              Programma elaborato da
%
%       Giovanni DI CECCA & Virginia BELLINO
%            50 / 887           408 / 466
%
%              http://www.dicecca.net
%
%
% Chiamata della funzione
%
% [AI,Errore,iflag,nval]=simfix(a,b,f,tolleranza,nfmax)
%
%
% Informazioni sul programma
%
% Scopo: Calcola l'integrale definito di f(x) utilizzando la formula
%        di Simpson a schema fisso. La funzione prosegue nel processo
%        iterativo fino a quando:
%        1) Il margine di errore raggiunge la tolleranza richiesta;
%        2) Si sono effettuate nfmax valutazioni;
%
% Parametri:
%
% Input:  a = Estremo sinistro dell'intervallo di integrazione
%         b = Estremo destro dell'intervallo di integrazione
%         f = Funzione integranda scritta nel seguente modo:
%
%         f='(100./(x.^7)).*sin(10./(x.^7))'
%
%         tolleranza = Tolleranza richiesta
%         nfmax = Massimo numero consentito di valutazioni della funzione
%
% Output: AI = Approssimazione dell'integrale
%         Errore = Stima dell'errore
%         iflag = Indicatore di errore:
%                 0 Se la condizione di uscita si è verificata per il
%                   raggiungimento della tolleranza richiesta;
%                 1 Se la condizione di uscita si è verificata per il
%                   raggiungimento del numero massimo di valutazioni;
%
%         nval = Numero di valutazioni della funzione effettuato

function [AI,Errore,iflag,nval]=simfix(a,b,f,tolleranza,nfmax)

% Inizializzazione delle variabili
```

```
Iprecedente=0; % Stima dell'integrale al passo precedente

AI=0; % Inizializzazione dell'approssimazione dell'integrale

% Inializzazione delle variabili usate per la valutazione
% della funzione negli estremi

fa=0;
fb=0;

% Valutazione della funzione negli estremi a et b
fa=Fun(f,a);
fb=Fun(f,b);

Errore=tolleranza;

% Numero di valutazioni della funzione effettuate per calcolare la
% prima approssimazione dell'integrale
nval=3;

iflag=0; % Inizializza l'indicatore di errore (tipo logical)

sommaprecedente=0; % Somma delle valutazioni di funzione all'iterazione k
sommacorrente=0; % Somma delle valutazioni di funzione all'iterazione k+1

h=(b-a)/2;

% Somma delle valutazioni di funzione al passo k+1
% Inizialmente, tale somma contiene il valore della funzione valutata nel
punto medio
% dell'intervallo di integrazione (a,b)
sommacorrente=Fun(f,a+h);

% Calcolo di Simpson semplice su tre punti
AI=(h/3)*(fa + 4*sommacorrente + fb);


k=1; % Inizializza il contatore k


% Inizio procedimento di Simpson a schema fisso
% Il calcolo viene effettuato con un ciclo while.

while (Errore>=tolleranza) & (nval<=nfmax)

  k=k+1; % Passa alla iterazione successiva

  Iprecedente=AI; % Memorizza il valore dell'ultima iterazione
```

```
  % Calcola l'ampiezza degli intervalli
  h=(b-a)/(2^k);

  % Calcola il numero dei nodi in cui andare a valutare la funzione
  m=2^(k-1);

 % Aggiornamento della somma delle valutazioni di funzione al passo k
  sommaprecedente = sommaprecedente + sommacorrente;

  % Aggiornamento della somma delle valutazioni di funzione al passo k+1
  % con l'aiuto della funzione ausiliaria Fun
  sommacorrente=0;

  for i=1:1:m
    sommacorrente = sommacorrente + Fun(f,a+((2*i)-1)*h);
  end

  % Calcolo dell'integrale definito con la formula di Simpson composta
  AI=(h/3)*(fa + 2*sommaprecedente + 4*sommacorrente + fb);

  % Calcola l'errore o resto utilizzando la stima di Richardson
  Errore=abs(AI-Iprecedente)/15;

  % Aggiornamento del numero di valutazioni effettuate
  nval=nval+m;
end

% Controllo sulla possibilità di terminazione del ciclo per il raggiungime
nto
% di una delle due condizioni previste, e cioè:
% a- il numero di valutazioni della funzione effettuate supera il massimo
consentito
% b- l'errore raggiunge la tolleranza richiesta
iflag=(nval>=nfmax & Errore>=tolleranza);


% Plot della funzione

hold on; % Consenti la possibilità di sovrascrivere il grafico

zoom on; % Abilita la funzione di zoom

% Inserisci i valori degli intervallini nel plot a video
x=linspace(a,b,abs(a-b)/0.01);

eval(sprintf('y_arr=%s;',f));

plot(x,y_arr,'b');
```

```
plot(x,-0.5,'rx');

hold off; % Disabilita Hold on
```

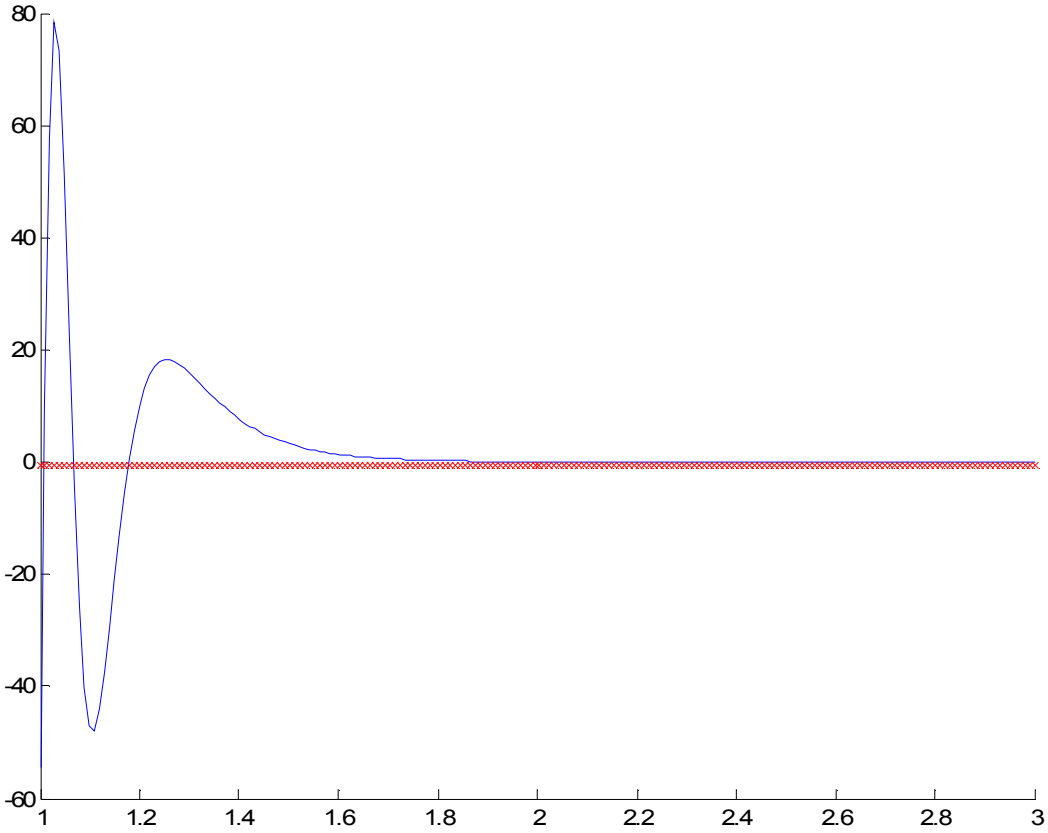
```
%            Progetto Metodo di Simpson
%                  a schema fisso
%
%              Programma elaborato da
%
%       Giovanni DI CECCA & Virginia BELLINO
%            50 / 887           408 / 466
%
%              http://www.dicecca.net
%
% Funzione ausiliaria di simfix
%
% Questa routine calcola i valori della funzione
% data in input secondo Matlab

function y=Fun( f, x )

% crea la funzione
Funzione = sprintf( 'y=%s;', f);

% calcola i valori
eval( Funzione );
```

Sezione 3

Testing

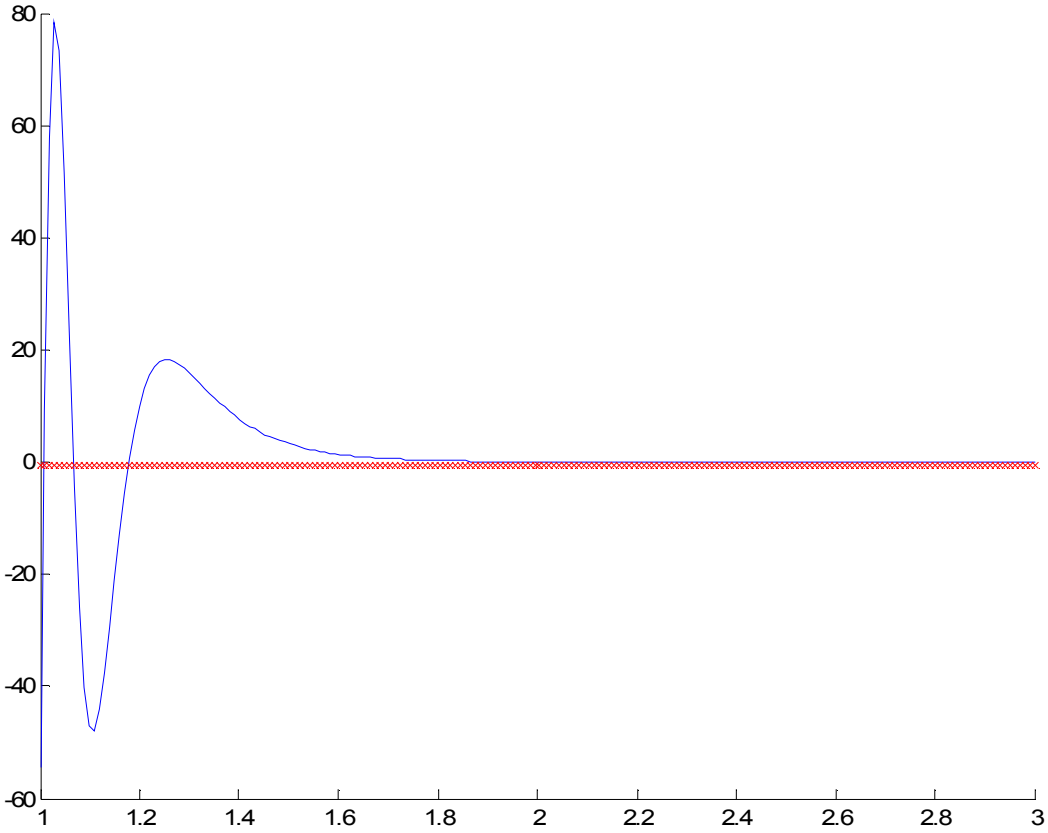
Primo TEST

```
fun =

(100./(x.^7)).*sin(10./(x.^7))

a =

     1

b =

     3

nfmax =

        1000

tolleranza =

   0.00100000000000

soluzioni della simfix

AI =

   3.34836501062069

Errore =

    8.794999028849541e-004

iflag =

     0

nval =

   257

Premi un tasto per continuare
```


80
60
40
20
0
-20
-40
-60
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3

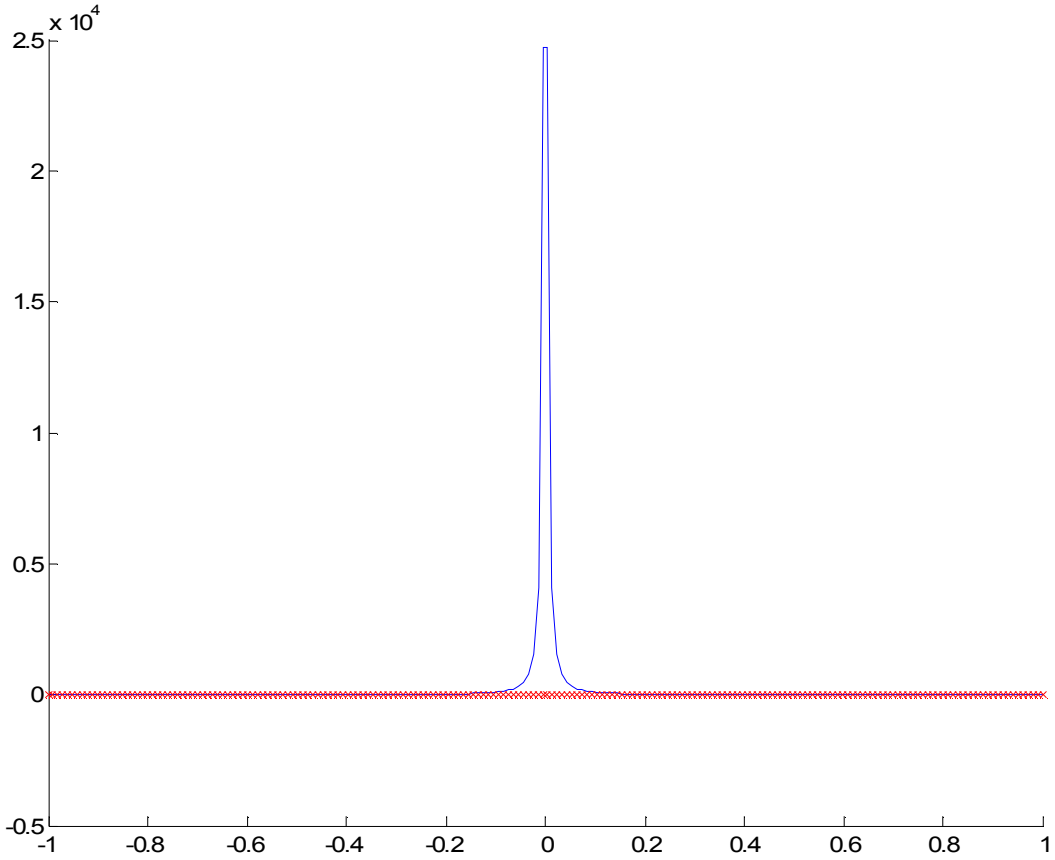
Secondo TEST

```
fun =

(100./(x.^7)).*sin(10./(x.^7))

a =

     1

b =

     3

nfmax =

       10000

tolleranza =

    1.000000000000000e-006

soluzioni della simfix

AI =

   3.34754709533644

Errore =

    1.941855582702582e-007

iflag =

     0

nval =

        2049

Premi un tasto per continuare
```


80
60
40
20
0
-20
-40
-60
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3

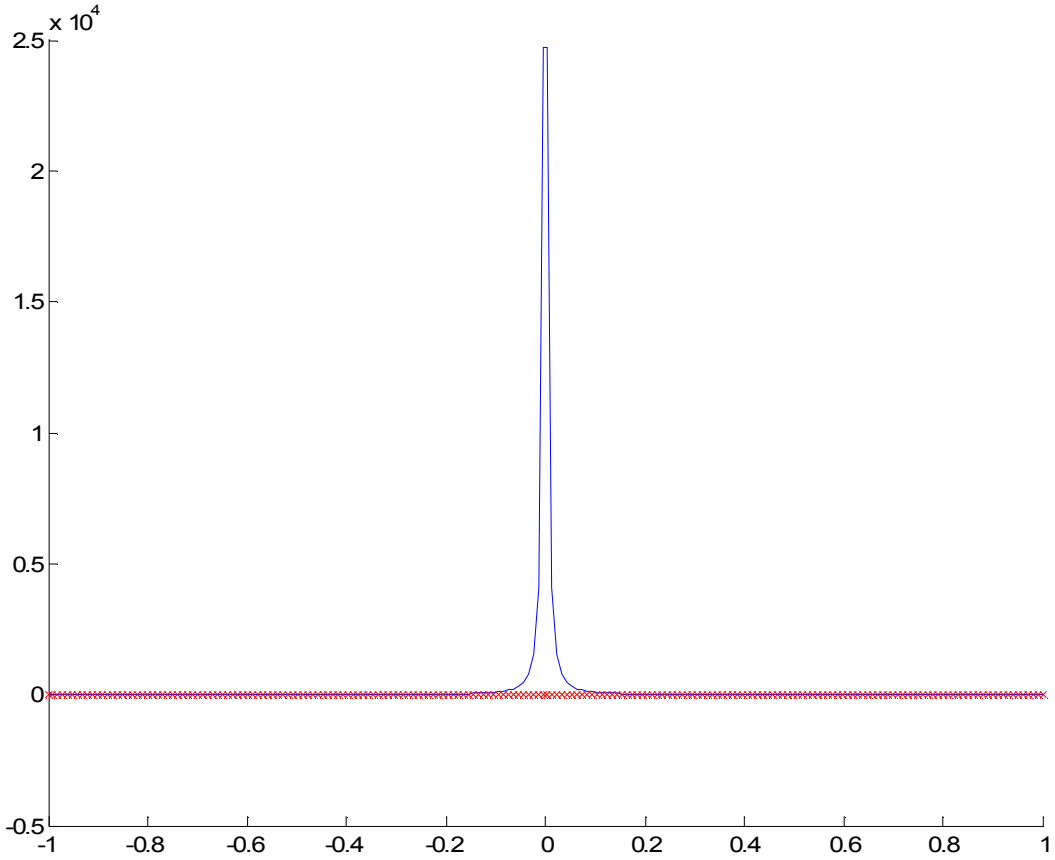
Terzo TEST

```
fun =

(1./((x.^2)+1./(2.^16)))

a =

    -1

b =

     1

nfmax =

       10000

tolleranza =

   0.10000000000000

soluzioni della simfix

AI =

    8.022458597195574e+002

Errore =

   0.06625213542247

iflag =

     0

nval =

       2049

Premi un tasto per continuare
```


x 10^4
2.5
2
1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

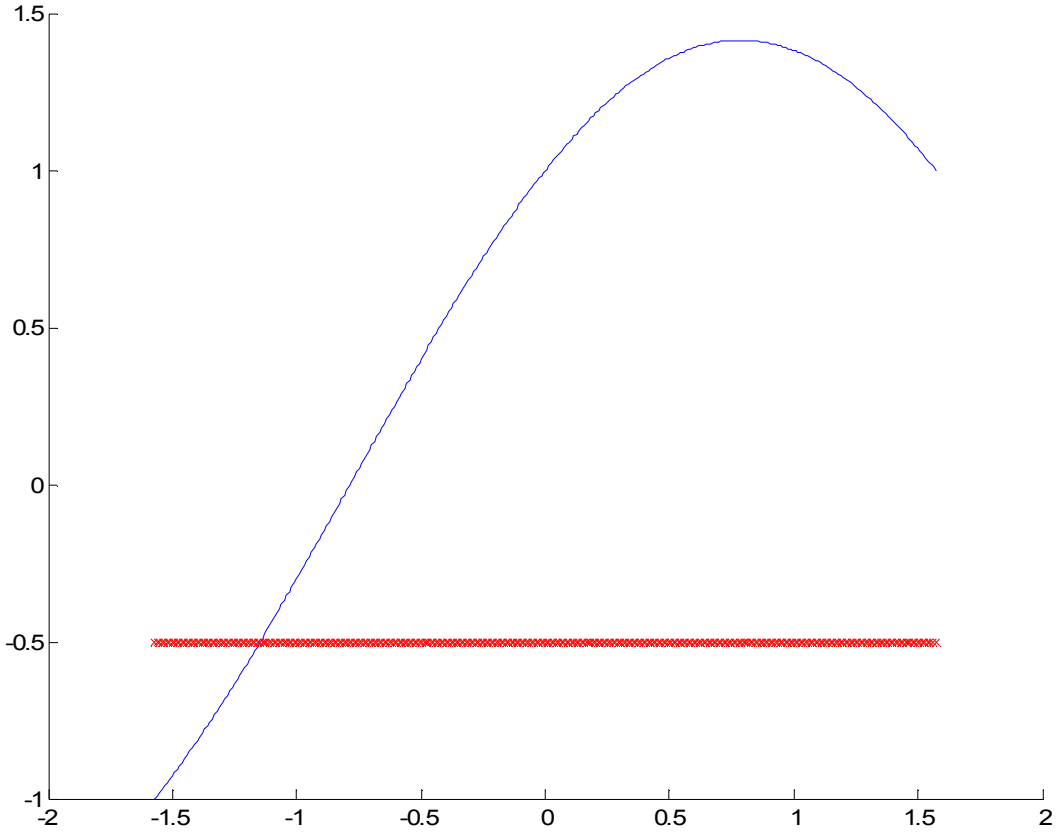
Quarto TEST

```
fun =

(1./((x.^2)+1./(2.^16)))

a =

    -1

b =

     1

nfmax =

       10000

tolleranza =

   0.00100000000000

soluzioni della simfix

AI =

    8.022477294848998e+002

Errore =

    1.246510228232485e-004

iflag =

     0

nval =

       4097

Premi un tasto per continuare
```


x 10^4
2.5
2
1.5
1
0.5
0
-0.5
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

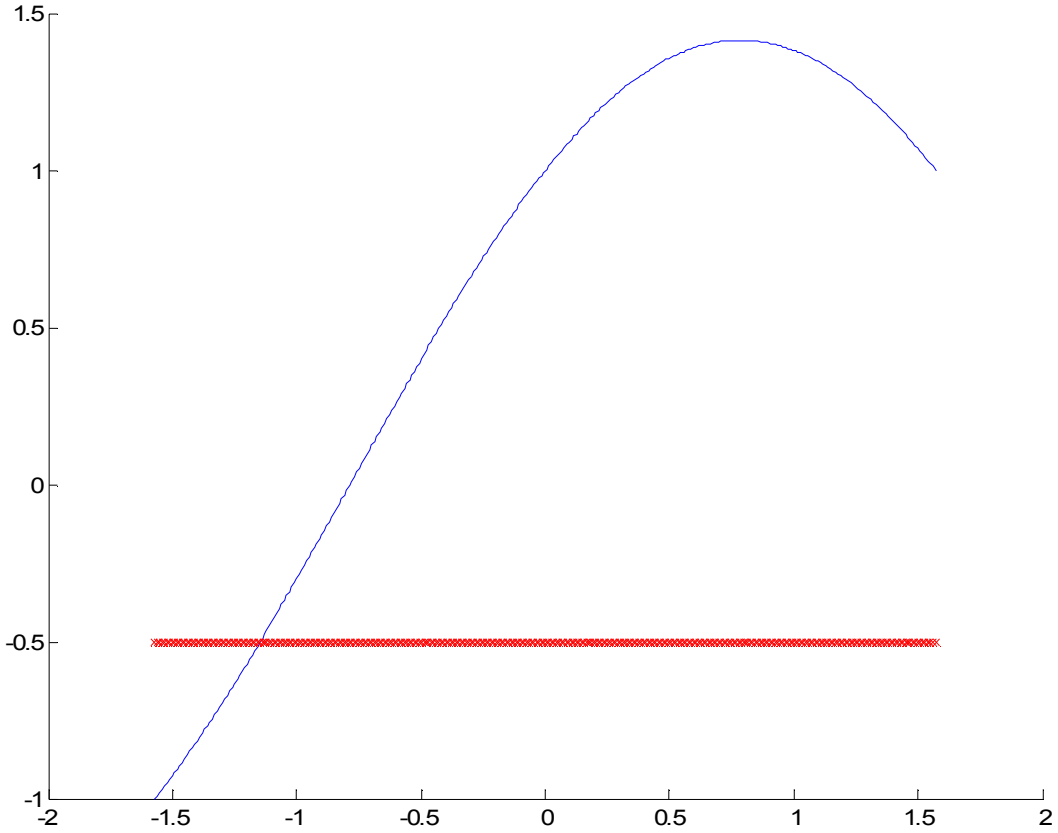
Quinto TEST

```
fun =

sin(x)+cos(x)

a =

  -1.57079632679490

b =

   1.57079632679490

nfmax =

   100

tolleranza =

   0.00100000000000

soluzioni della simfix

AI =

   2.00026916994839

Errore =

    2.860390024022053e-004

iflag =

     0

nval =

     9

Premi un tasto per continuare
```


1.5
1
0.5
0
-0.5
-1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

Sesto TEST

```
fun =

sin(x)+cos(x)

a =

  -1.57079632679490

b =

   1.57079632679490

nfmax =

   100

tolleranza =

    1.000000000000000e-006

soluzioni della simfix

AI =

   2.00000006453000

Errore =

    6.458929409092207e-008

iflag =

     0

nval =

    65

Fine del test
```

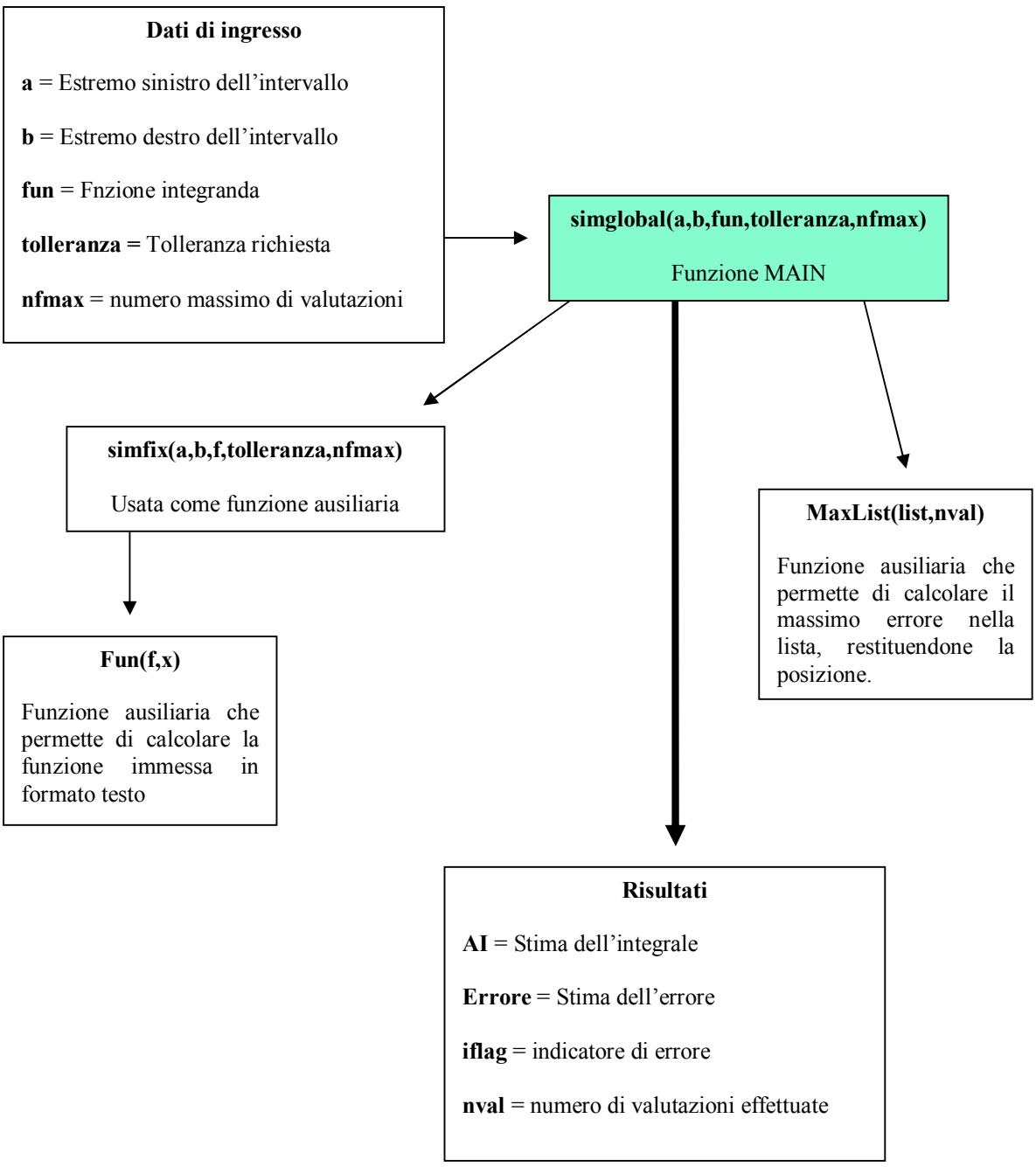

1.5
1
0.5
0
-0.5
-1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

Simpson a schema adattivo con controllo globale dell'errore

Progetto Metodo di Simpson a schema globale

$$\int_a^b f(x)\,dx$$

Schema grafico



Sezione 1

Documentazione esterna

- **Scopo:** calcolare il valore numerico approssimato di un integrale definito del tipo

$$\int_a^b f(x)\,dx$$

utilizzando la strategia basata sulla *formula di quadratura composta di Simpson a schema adattivo con controllo globale dell'errore*.

- **Specifiche:**

function [AI,Errore,iflag,nval]=simglobal(a,b,fun,tolleranza,nfmax)

- **Descrizione:**

dopo aver inizializzato le variabili e calcolato la prima stima dell'integrale richiamando la funzione simfix (eseguita su 3 punti),l'algoritmo crea una lista in cui verranno memorizzati tutti i dati relativi ai sottointervalli.
Successivamente, ha inizio il ciclo while che esegue il procedimento di Simpson a schema adattivo. Tale ciclo esegue le seguenti operazioni:

- Ricerca nella lista dell'intervallo con errore massimo utilizzando il modulo ausiliario MaxList.
- L'intervallo selezionato viene suddiviso in 2 parti per calcolare le nuove approssimazioni di integrale ed errore.
- Rimozione dalla lista della vecchia approssimazione relativa all'intervallo selezionato.
- Calcolo delle nuove approssimazioni sul sottointervallo di sinistra e di destra.
- Aggiornamento della lista con i nuovi dati.
- Aggiornamento delle stime complessive di integrale ed errore

Infine, vi è un controllo sulle condizioni di terminazione del ciclo, cui seguono le istruzioni per il plot della funzione.

♦ Riferimenti bibliografici:

- James F. Epperson

INTRODUZIONE ALL'ANALISI NUMERICA
McGraw-Hill

- Prof. Eleonora Messina

Appunti del corso di calcolo numerico
A.A 2004/2005

♦ Lista dei parametri:

Parametri di input:

a = Estremo sinistro dell'intervallo di integrazione
b = Estremo destro dell'intervallo di integrazione
fun = Funzione integranda scritta nel seguente modo:

```
f='(100./(x.^7)).*sin(10./(x.^7))'
```

tolleranza = Tolleranza richiesta
nfmax = Massimo numero consentito di valutazioni della funzione

Parametri di output:

AI = Approssimazione dell'integrale
Errore = Stima dell'errore
iflag = Indicatore di errore:

0 Se la condizione di uscita si è verificata per il raggiungimento della tolleranza richiesta;

1 Se la condizione di uscita si è verificata per il raggiungimento del numero massimo di valutazioni;

nval = Numero di valutazioni della funzione effettuato

- **Indicatori di errore:**

iflag: indica se durante la procedura si verificano degli errori,bloccando in tal caso l'esecuzione. Può assumere i seguenti valori:

0 Se la condizione di uscita si è verificata per il raggiungimento della tolleranza richiesta;

1 Se la condizione di uscita si è verificata per il raggiungimento del numero massimo di valutazioni;

- **Funzioni ausiliarie:**

funzione MaxList:
funzione che esegue la ricerca nella lista dell'intervallo con errore massimo. Una volta trovato, ne restituisce i dati al programma chiamante.
La specifica è:
function [E,approx,xl,xh,k]=MaxList(list,c)

- **Complessità computazionale:**

Complessità di tempo: dipende dalla complessità della funzione e dal numero di valutazioni

Complessità di spazio:

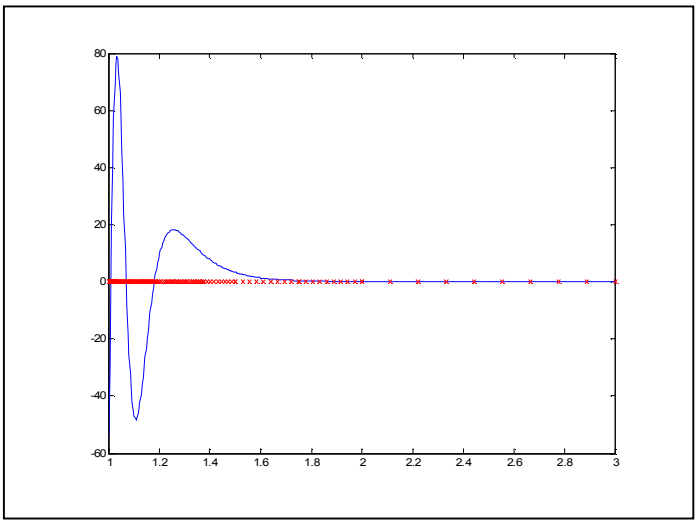
```
EDU>>
EDU>> whos
  Name          Size        Bytes  Class

  AI            1x1             8  double array
  Errore        1x1             8  double array
  a             1x1             8  double array
  b             1x1             8  double array
  fun           1x30           60  char array
  iflag         1x1             8  double array (logical)
  nfmax         1x1             8  double array
  nval          1x1             8  double array
  tolleranza    1x1             8  double array

Grand total is 38 elements using 124 bytes
```

- **Accuratezza fornita:**

l'accuratezza del risultato dipende dalla tolleranza richiesta,che viene fornita in input.

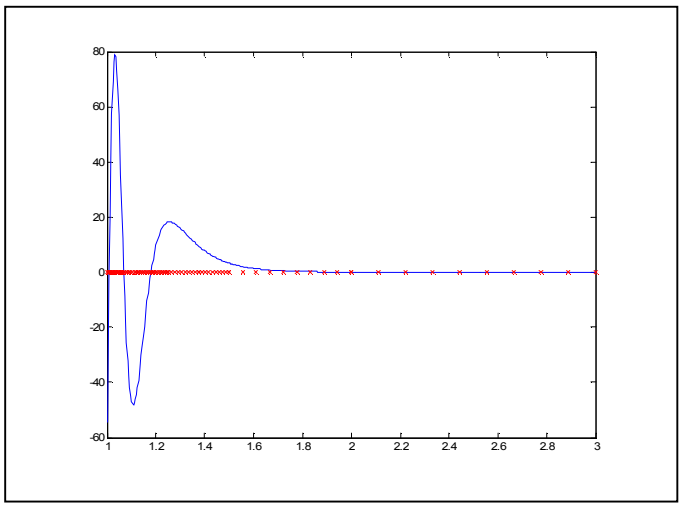
♦ Esempio d'uso:

```
fun =

(100./(x.^7)).*sin(10./(x.^7))

a =

     1

b =

     3

nfmax =

        1000

tolleranza =

   0.00100000000000

soluzioni della simglobal

AI =

   3.34792914607643

Errore =

    7.037416393202394e-004

iflag =

     0

nval =

    14
```



- **Esempio d'uso:**

```
fun =

(100./(x.^7)).*sin(10./(x.^7))

a =

     1

b =

     3

nfmax =

     8

tolleranza =

   0.00100000000000

soluzioni della simglobal

AI =

   3.34925629756928

Errore =

   0.00599895147469

iflag =

     1

nval =

     9
```



- **Raccomandazioni d'uso:** I file sono stati implementati su **MATLAB Student Version 6.0**. Durante la fase di test su MATLAB 5.2 si sono verificati degli anomalie su alcuni plot. **Si raccomanda l'uso di MATLAB 6.0 o superiore.** Inoltre digitando **help simglobal** si ha un rapido help sull'uso del file.

Sezione 2

Codice MATLAB

```
%            Progetto Metodo di Simpson
%                  a schema globale
%
%              Programma elaborato da
%
%       Giovanni DI CECCA & Virginia BELLINO
%            50 / 887           408 / 466
%
%              http://www.dicecca.net
%
%
% Chiamata della funzione
%
% [AI,Errore,iflag,nval]=simglobal(a,b,fun,tolleranza,nfmax)
%
%
% Informazioni sul programma
%
%
% Scopo: Calcola l'integrale definito di f(x) utilizzando la formula
%        di Simpson a schema adattivo con controllo globale dell'errore.
%        La funzione prosegue nel processo iterativo fino a quando:
%
%        1) Il margine di errore raggiunge la tolleranza richiesta;
%        2) Si sono effettuate nfmax valutazioni;
%
% Parametri:
%
% Input : a = Estremo sinistro dell'intervallo di integrazione
%         b = Estremo destro dell'intervallo di integrazione
%         fun = Funzione integranda scritta nel seguente modo:
%
%         fun='(100./(x.^7)).*sin(10./(x.^7))'
%
%         tolleranza = Tolleranza richiesta
%         nfmax = Massimo numero consentito di valutazioni della funzione
%
% Output: AI = Stima dell'integrale
%         Errore = Stima dell'errore
%         iflag = Indicatore di errore:
%                 0 Se la condizione di uscita si è verificata per il
%                   raggiungimento della tolleranza richiesta;
%                 1 Se la condizione di uscita si è verificata per il
%                   raggiungimento del numero massimo di valutazioni;
%
%         nval = Numero di valutazioni di funzione effettuato


function [AI,Errore,iflag,nval]=simglobal(a,b,fun,tolleranza,nfmax)
```

```
%Inizializzo le variabili
nval=1;

% Chiama la funzione simfix per eseguire la prima stima
% dell'integrale (P.S.: z et g non vengono considerati)
[Iprecedente,Errore,z,g]=simfix(a,b,fun,0,3);

% Crea una lista in cui vengono inseriti i dati relativi agli intervalli
list(nval)=struct('xl',a,'xh',b,'approx',Iprecedente,'est',Errore);

AI=Iprecedente;

% Inizio procedimento Simpson adattivo
while (Errore>=tolleranza & nval<=nfmax)

   % Rileva nella lista l'intervallo con il massimo
   % errore mediante una funzione apposita
   [E,Iprecedente,xa,xb,k]=MaxList(list,nval);

   % Suddivide l'intervallo selezionato in due sottointervalli
   h=(xb-xa)/2;

   % Calcola il punto medio
   xmid=xa+h;

   % Rimuove dalla lista la vecchia approssimazione
   % relativa all'intervallo selezionato
   list(k)=[];

   % Calcola la nuova approssimazione sul sottointervallo di sinistra
   % ed aggiorna la lista
   [I1,E1,z,g]=simfix(xa,xmid,fun,0,3);
   list(nval)=struct('xl',xa,'xh',xmid,'approx',I1,'est',E1);

   % Calcola la nuova approssimazione sul sottointervallo di destra
   % ed aggiorna la lista
   nval=nval+1;
   [I2,E2,z,g]=simfix(xmid,xb,fun,0,3);
   list(nval)=struct('xl',xmid,'xh',xb,'approx',I2,'est',E2);

   % Ricava la nuova stima dell'integrale e dell'errore.
   AI=AI-Iprecedente+I1+I2;
   Errore=Errore-E+E1+E2;
end

% Controllo sulla possibilità di terminazione del ciclo per il raggiungime
nto
% di una delle due condizioni previste, e cioè:
```

```
% a- il numero di valutazioni della funzione effettuate supera il massimo
consentito
% b- l'errore raggiunge la tolleranza richiesta
iflag=(nval>=nfmax & Errore>=tolleranza);

% Disegna la funzione f(x)
x=linspace(a,b,500);
eval(sprintf('y_arr=%s;',fun));
plot(x,y_arr);

zoom on
hold on

% Calcola i valori dei nodi
arr_nodi=[];
for (i=1:nval)
  x=linspace(getfield(list(i),'xl'),getfield(list(i),'xh'),10);
  arr_nodi = [arr_nodi x];
end

% Disegna i nodi
plot(arr_nodi,0,'rx');

hold off
```

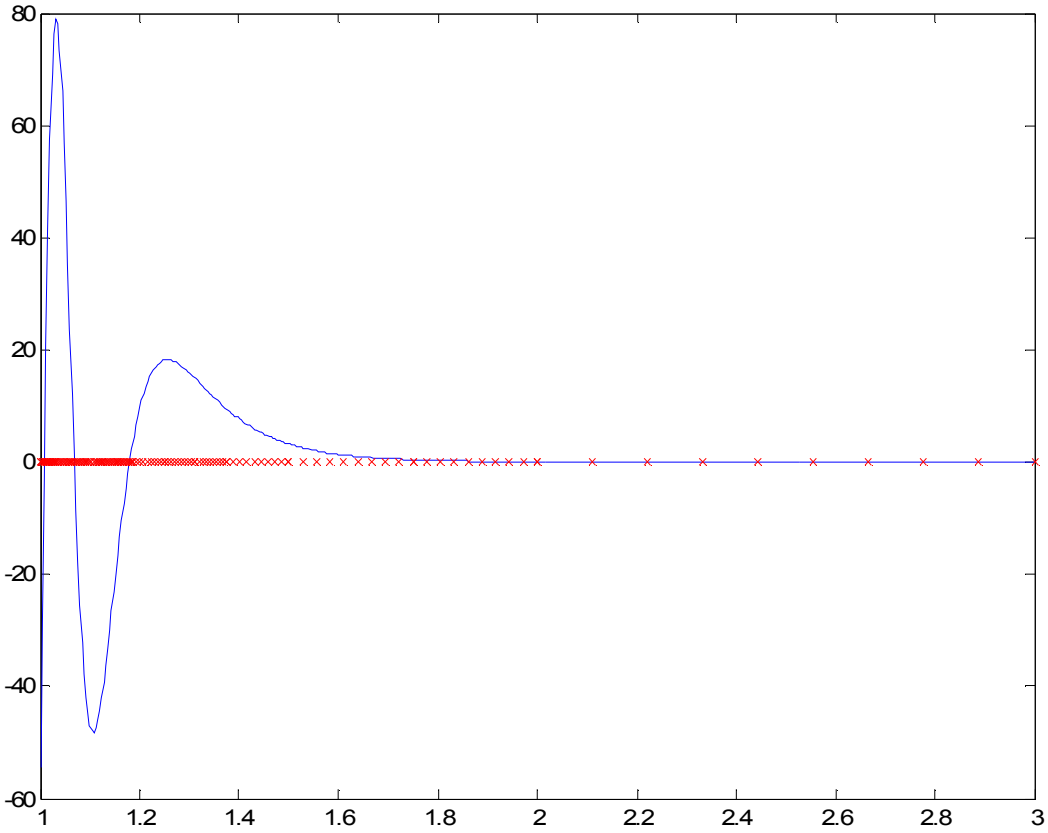
```
%            Progetto Metodo di Simpson
%                  a schema globale
%
%              Programma elaborato da
%
%       Giovanni DI CECCA & Virginia BELLINO
%            50 / 887           408 / 466
%
%              http://www.dicecca.net

% Funzione ausiliaria di simglobal

function [E,approx,xl,xh,k]=MaxList(list,c)

% Salva i valori alla testa della struttura
E=getfield(list(1),'est');

approx=getfield(list(1),'approx');

xl=getfield(list(1),'xl');

xh=getfield(list(1),'xh');

k=1;

% Ricerca dei dati relativi all'intervallo con errore massimo
for (j=2:1:c)

    % Confronta la testa della lista con gli
    % altri valori
    if (E<getfield(list(j),'est'))

        E=getfield(list(j),'est');

        approx=getfield(list(j),'approx');

        xl=getfield(list(j),'xl');

        xh=getfield(list(j),'xh');

        % Indica la posizione nella list adell'intervallo
        % con il massimo errore
        k=j;

    end % End if

end % end for
```

Sezione 3

Testing

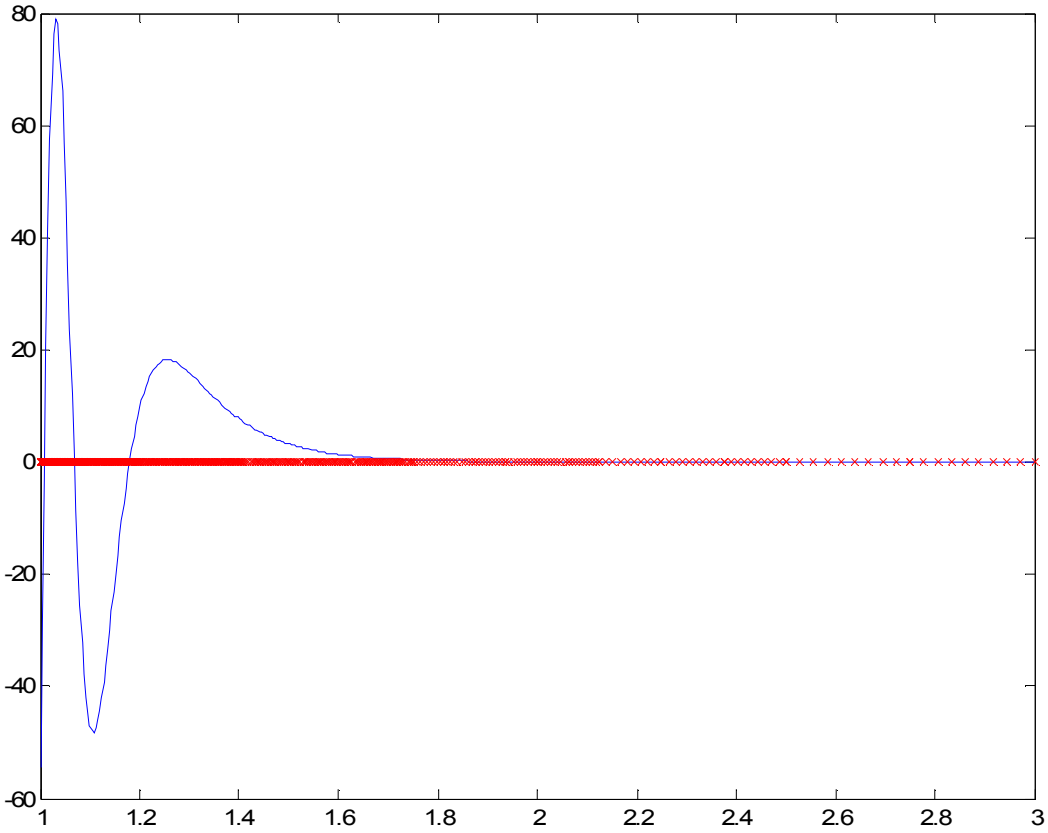
Primo TEST

```
fun =

(100./(x.^7)).*sin(10./(x.^7))

a =

     1

b =

     3

nfmax =

        1000

tolleranza =

   0.00100000000000

soluzioni della simglobal

AI =

   3.34792914607643

Errore =

    7.037416393202394e-004

iflag =

     0

nval =

    14

Premi un tasto per continuare
```


80
60
40
20
0
-20
-40
-60
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3

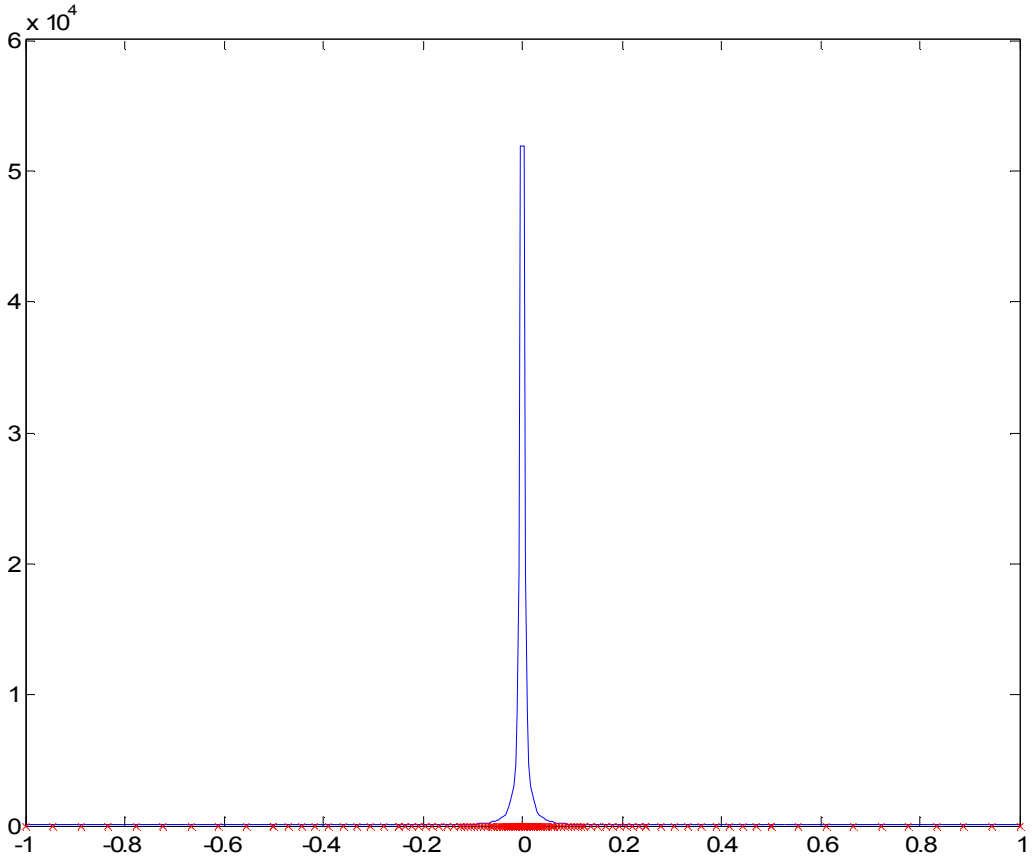
Secondo TEST

```
fun =

(100./(x.^7)).*sin(10./(x.^7))

a =

     1

b =

     3

nfmax =

        1000

tolleranza =

    1.000000000000000e-006

soluzioni della simglobal

AI =

   3.34754720021388

Errore =

    9.519758794661029e-007

iflag =

     0

nval =

    72
```


80
60
40
20
0
-20
-40
-60
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3

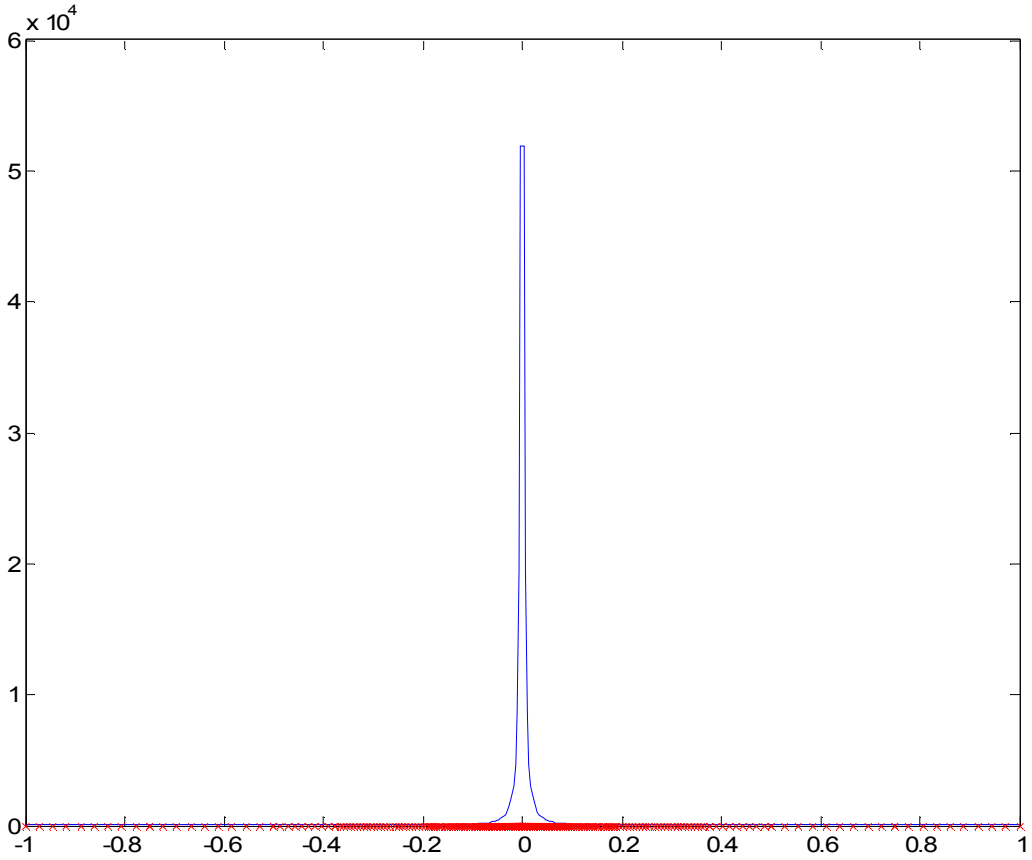
Terzo TEST

```
fun =

(1./((x.^2)+1./(2.^16)))

a =

    -1

b =

     1

nfmax =

        1000

tolleranza =

   0.10000000000000

soluzioni della simglobal

AI =

    8.023725162233195e+002

Errore =

   0.09883973133026

iflag =

     0

nval =

    20

Premi un tasto per continuare
```


x 10^4
6
5
4
3
2
1
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

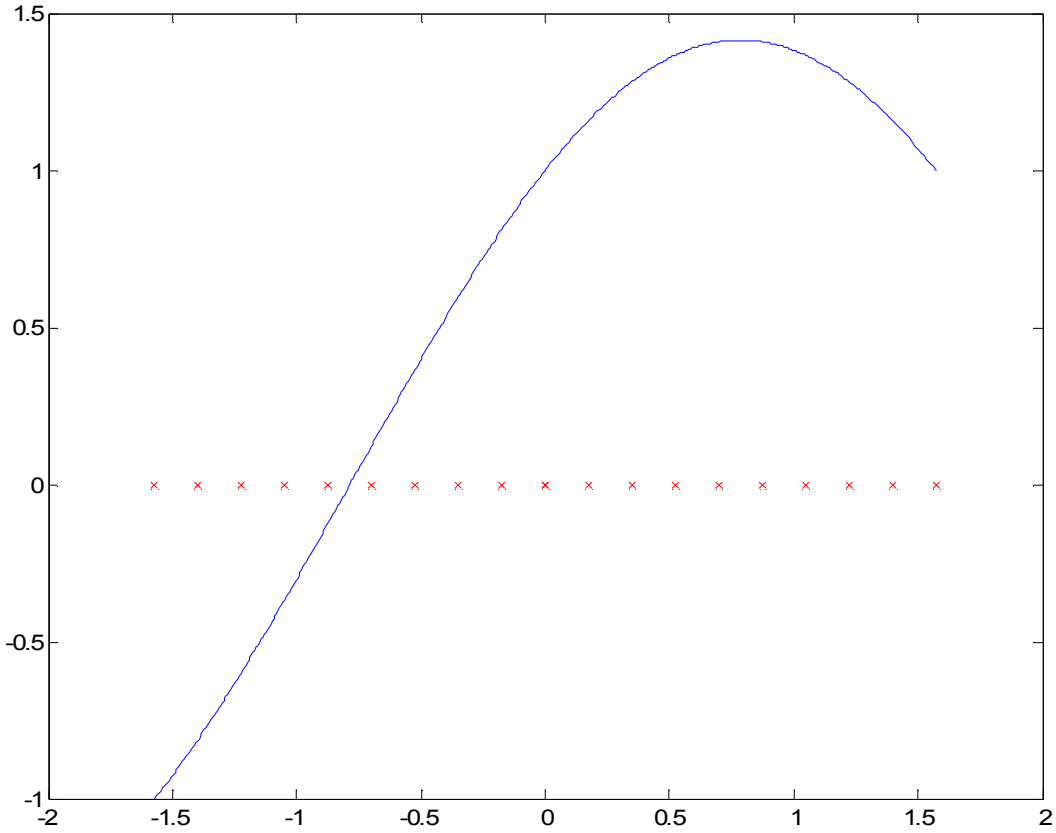
Quarto TEST

```
fun =

(1./((x.^2)+1./(2.^16)))

a =

    -1

b =

     1

nfmax =

        1000

tolleranza =

   0.00100000000000

soluzioni della simglobal

AI =

    8.022483189444289e+002

Errore =

    9.754756183261852e-004

iflag =

     0

nval =

    65

Premi un tasto per continuare
```


x 10^4
6
5
4
3
2
1
0
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

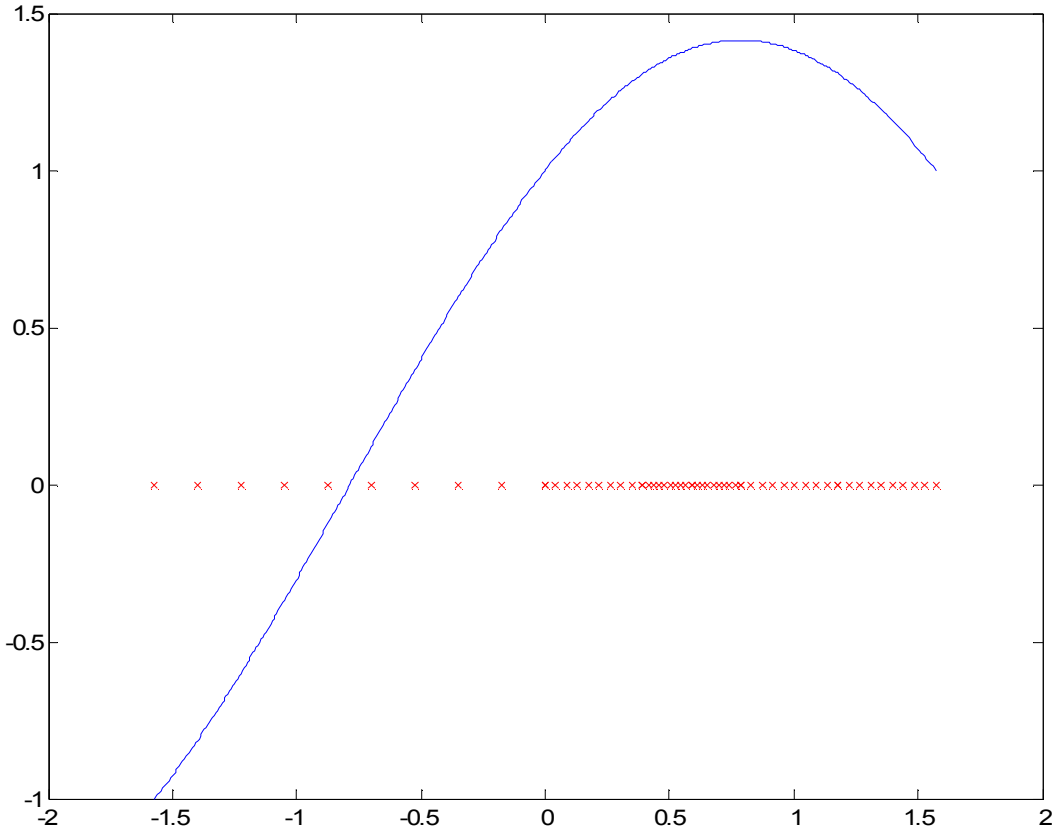
Quinto TEST

```
fun =

sin(x)+cos(x)

a =

  -1.57079632679490

b =

   1.57079632679490

nfmax =

        1000

tolleranza =

   0.00100000000000

soluzioni della simglobal

AI =

   2.00026916994839

Errore =

    2.860390024022120e-004

iflag =

     0

nval =

     2

Premi un tasto per continuare
```


1.5
1
0.5
0
-0.5
-1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

Sesto TEST

```
fun =

sin(x)+cos(x)

a =

  -1.57079632679490

b =

   1.57079632679490

nfmax =

        1000

tolleranza =

    1.000000000000000e-006

soluzioni della simglobal

AI =

   2.00000077120336

Errore =

    7.739978070791364e-007

iflag =

     0

nval =

     6

Fine del test
```


1.5
1
0.5
0
-0.5
-1
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

Osservazioni

Simpson fisso

Poiché l'idea di fondo dell'algoritmo di Simpson a schema fisso è quella di campionare la funzione in punti equidistanti (e ciò si ottiene dimezzando l'ampiezza degli intervalli ad ogni iterazione) , osservando i test si può notare che esso risulta efficiente solo nei casi in cui *la funzione presenta un andamento abbastanza regolare in tutto l'intervallo di integrazione*.

In caso contrario infatti, ovvero quando la funzione presenta brusche oscillazioni in alcuni sottointervalli, per arrivare ad ottenere una precisione maggiore del risultato, vengono effettuate molte valutazioni della funzione assolutamente non necessarie, dovute al fatto che, andando ad infittire il numero dei nodi nelle cosiddette "zone difficili", si aumenta inutilmente anche il numero dei nodi nelle zone ove l'andamento è regolare.

Inoltre, è opportuno osservare anche che, diminuendo la tolleranza, e quindi aumentando la precisione richiesta, cresce il numero di nodi in cui valutare la funzione per ottenere il risultato.

Simpson adattivo

La strategia adattiva, basata sull'idea di scegliere i nodi adeguandosi all'andamento della funzione, consente di aumentare l'efficienza dell'algoritmo quando ci si trova davanti ad una *funzione che presenta un andamento non regolare caratterizzato da bruschi cambiamenti lungo l'intervallo di integrazione*.

In questi casi infatti, il numero dei nodi viene infittito nelle "zone difficili" caratterizzate da bruschi cambiamenti di andamento ed alto tasso di errore, mentre nei sottointervalli più regolari, ove l'errore è già basso, il numero dei nodi si dirada.

In tal modo è possibile raggiungere la precisione richiesta con un numero minore di iterazioni.

@@@@@@

Quando invece ci si trova davanti a funzioni con un andamento oscillante ma abbastanza morbido su tutto l'intervallo di integrazione, i due algoritmi presentano pressappoco la medesima efficienza.

Ma osserviamo ora più in dettaglio i risultati di ciascun test.

PRIMA FUNZIONE

In questo caso, la funzione presenta inizialmente una brusca oscillazione, per poi stabilizzarsi ed assumere un andamento regolare.

Usando lo schema fisso, il numero delle valutazioni effettuate risulta considerevole a causa dei motivi esposti in precedenza (si va da 257 per una tolleranza di 10^{-3} fino a salire a 2049 per una tolleranza di 10^{-6}), mentre scegliendo lo schema adattivo tali cifre scendono sensibilmente (14 per una tolleranza di 10^{-3} e 72 per una tolleranza di 10^{-6}).

In casi di questo tipo dunque, è migliore l'algoritmo adattivo.

SECONDA FUNZIONE

La seconda funzione ha un andamento in gran parte lineare, ma è caratterizzata da un picco altissimo in prossimità dello zero.

Ciò implica che, usando lo schema fisso il numero di valutazioni effettuate per raggiungere il risultato è enorme con entrambe le tolleranze date in input.
Invece, la scelta dello schema adattivo si rivela migliore poiché il risultato viene raggiunto con sole 20 valutazioni per una tolleranza di 10^{-1} (contro le 2049 dello schema fisso) e 65 valutazioni per una tolleranza di 10^{-3} (contro le 4097 dello schema fisso).

Anche questo esempio mostra dunque la migliore efficienza dello schema adattivo in presenza di funzioni non regolari.

TERZA FUNZIONE

La terza funzione ha invece un andamento abbastanza regolare, e ciò implica che sia l'algoritmo a schema fisso che quello a schema adattivo producono risultati non molto dissimili tra loro.

Ciò è evidente soprattutto nel primo caso, ove, con tolleranza 10^{-3} lo schema fisso esegue 9 valutazioni, mentre quello adattivo appena 2.
Aumentando la tolleranza a 10^{-6} il divario diventa più evidente con 65 valutazioni per lo schema fisso e appena 6 per quello adattivo.

LIBERTÀ EGUAGLIANZA

MONITORE NAPOLETANO

Fondato nel 1799 da
Carlo Lauberg ed Eleonora de Fonseca Pimentel

Rifondato nel 2010
Direttore: Giovanni Di Cecca

Anno CCXXI

Contatti

C.Ph.: +39 392 842 76 67

www.monitorenapoletano.it

info@monitorenapoletano.it