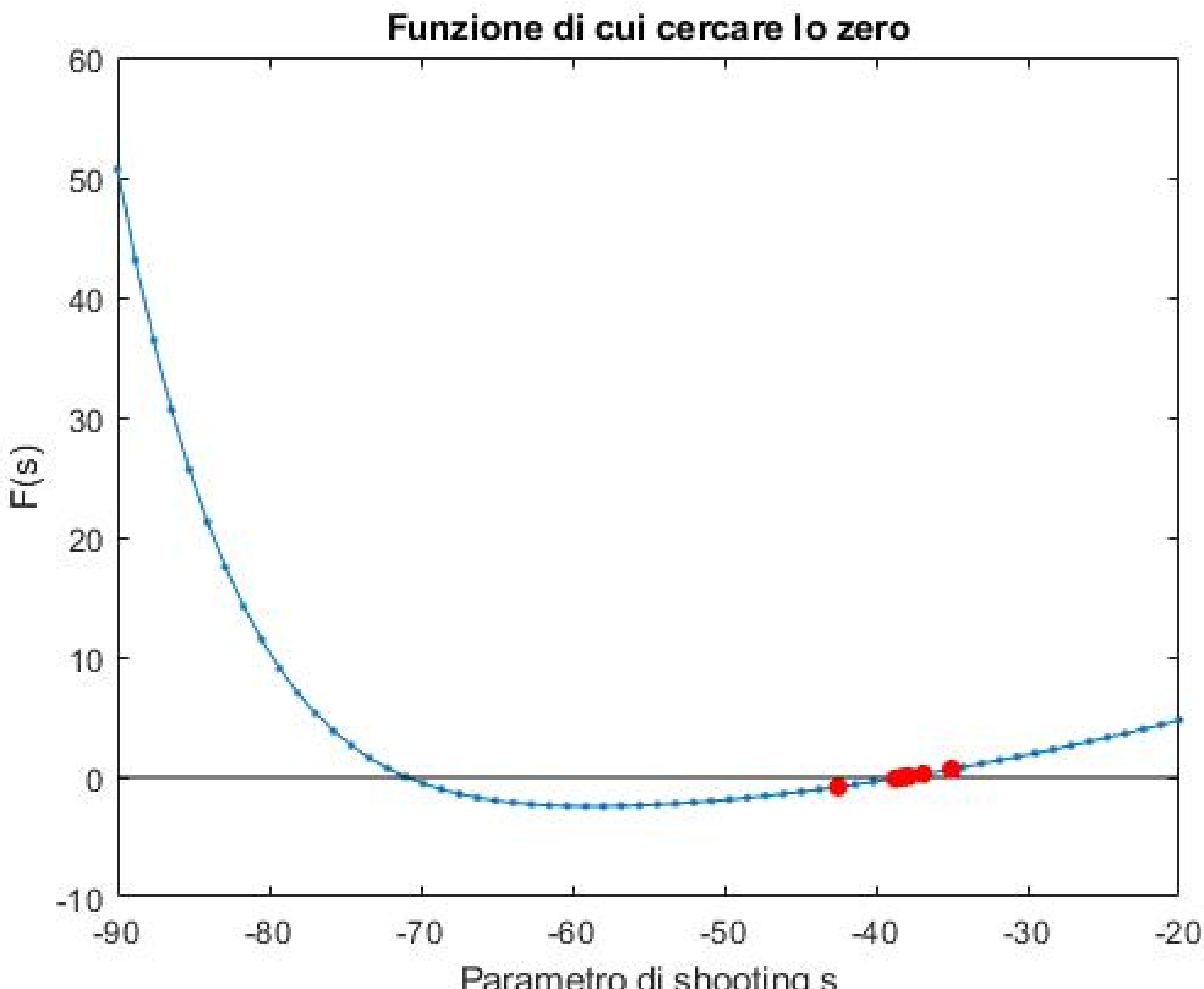

Funzione di cui cercare lo zero
60
50
40
30
20
10
0
-10
F(s)
-90
-80
-70
-60
-50
-40
-30
-20
Parametro di shooting s

Metodi Numerici in Ingegneria Aerospaziale

2 Luglio 2015

Si consideri la equazione di convezione-diffusione lineare instazionaria e non omogenea:

$$\varphi_t + a\varphi_x - k\varphi_{xx} = \pi(x) \tag{1}$$

nel dominio $[0, L]$ con condizione iniziale:

$$\varphi(x, 0) = \frac{L}{\pi}\sin\left(\frac{\pi}{L}x\right)$$

e condizioni al contorno:

$$\begin{aligned} \varphi(0, t) &= \gamma(t) \\ \left.\frac{\partial \varphi}{\partial x}\right|_{x=L} &= 1. \end{aligned}$$

Per le funzioni $\pi(x)$ e $\gamma(t)$ si assumano le leggi:

$$\begin{aligned} \pi(x) &= x(L - x) \\ \gamma(t) &= 1 - e^{-wt} \end{aligned}$$

Si risolvano in sequenza i seguenti punti:

1. Si discretizzi l'equazione assegnata su un mesh uniforme di ampiezza h, utilizzando per il termine convettivo uno schema esplicito di tipo *backward* su tre punti, mentre per il termine diffusivo uno schema implicito alla Crank-Nicolson (collocato a $t^{n+1/2}$) centrale su cinque punti. Si scriva il codice in modo che le costanti L, a, k, w siano dei parametri assegnabili in ingresso.

2. Con riferimento alla versione stazionaria del modello (1), si determini analiticamente la soluzione e la si confronti con la soluzione numerica ottenuta simulando la equazione per tempi sufficientemente lunghi. Si studi la accuratezza della discretizzazione al variare del passo h e per diversi valori del gruppo adimensionale aL/k.

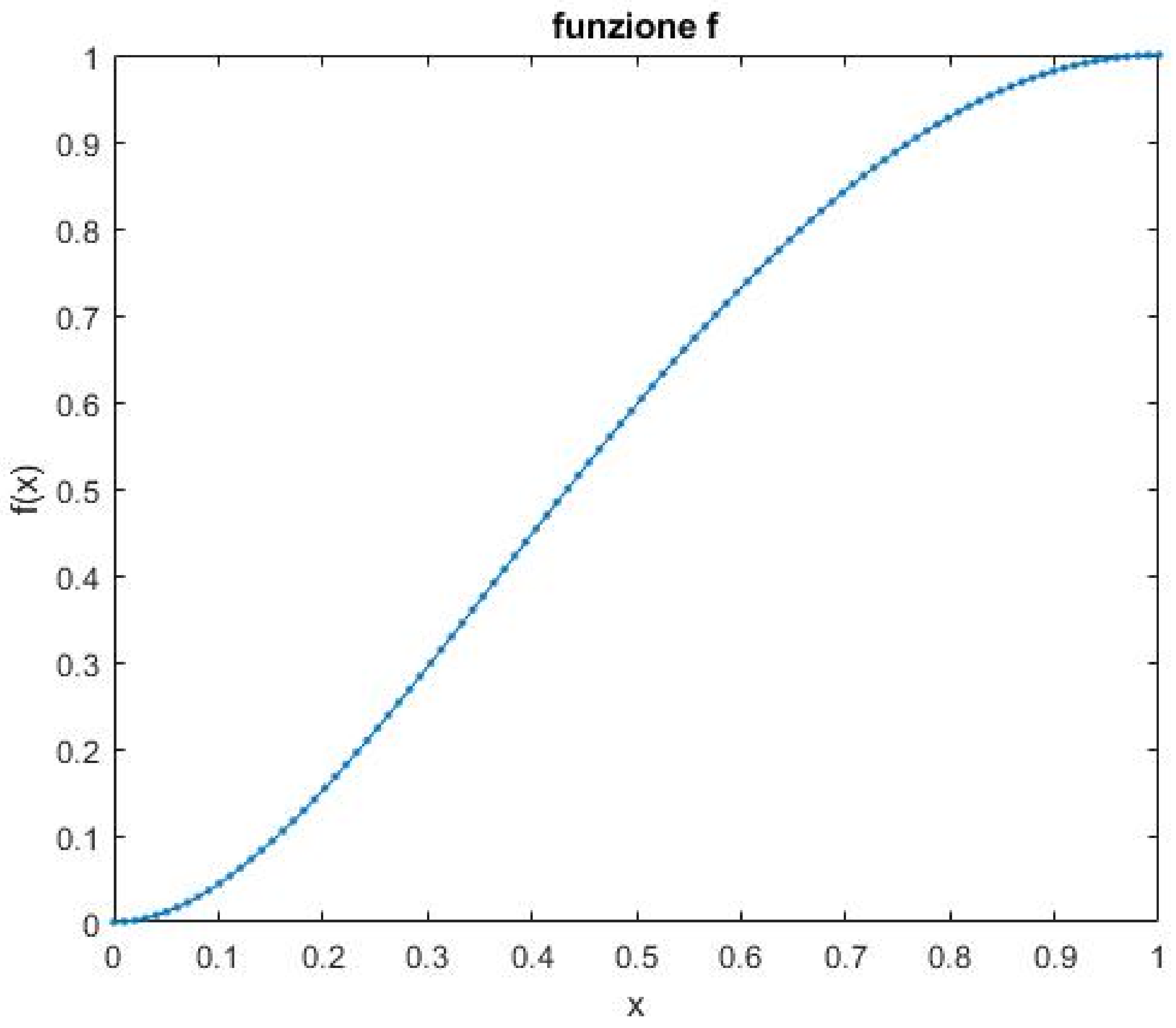
```matlab
function W = PesiDer(x,xc,p)

% Esercitazione di 'Metodi Numerici in Ingegneria Aerospaziale'
% 14 ottobre 2016
% Prof. G. Coppola
%
% Function per il calcolo dei pesi delle formule di derivazione
mediante il
% metodo dei coefficienti indeterminati.
%
% Parametri di ingresso:
% x    -->    Vettore di coordinate dei nodi del mesh.
% xc   -->    Punto di collocazione delle derivate.
% p    -->    Ordine di derivazione della formula richiesta.
%
% Parametri di uscita:
% W    -->    Riga dei pesi delle derivate o, se nargin = 2, matrice
%             contenente sulla j-ma riga i pesi della derivata j-1.

N   = length(x);            % N è il numero di nodi dello stencil.
x   = x(:).';               % Rendiamo il vettore x una riga.
csi = x - xc;               % Definiamo la variabile ausiliaria x-xc.
M   = zeros(N);             % Allocazione della matrice M.
% Composizione della matrice.
for j = 1:N
    M (j,:) = csi.^(j-1)/factorial(j-1);
end
I = eye(N);                 % Matrice identica.
% Risoluzione del sistema per i pesi di tutte le derivate (se è
fornito p,
% il termine noto per i pesi della derivata di ordine p sarà la
(p+1)-ma
% colonna della matrice identica).
W = M\I;                    % Risoluzione del sistema.
W = W.';                    % Mettiamo i pesi sulle righe.
% Se richiesto, estraiamo solo i pesi della p-ma derivata (NB la
presente
% versione non è 'ottimizzata', poichè calcola in ogni caso i pesi di
tutte
% le derivate, anche se è richiesta solo la derivata di ordine p).
if nargin==3
    W = W(p+1,:);
end
```

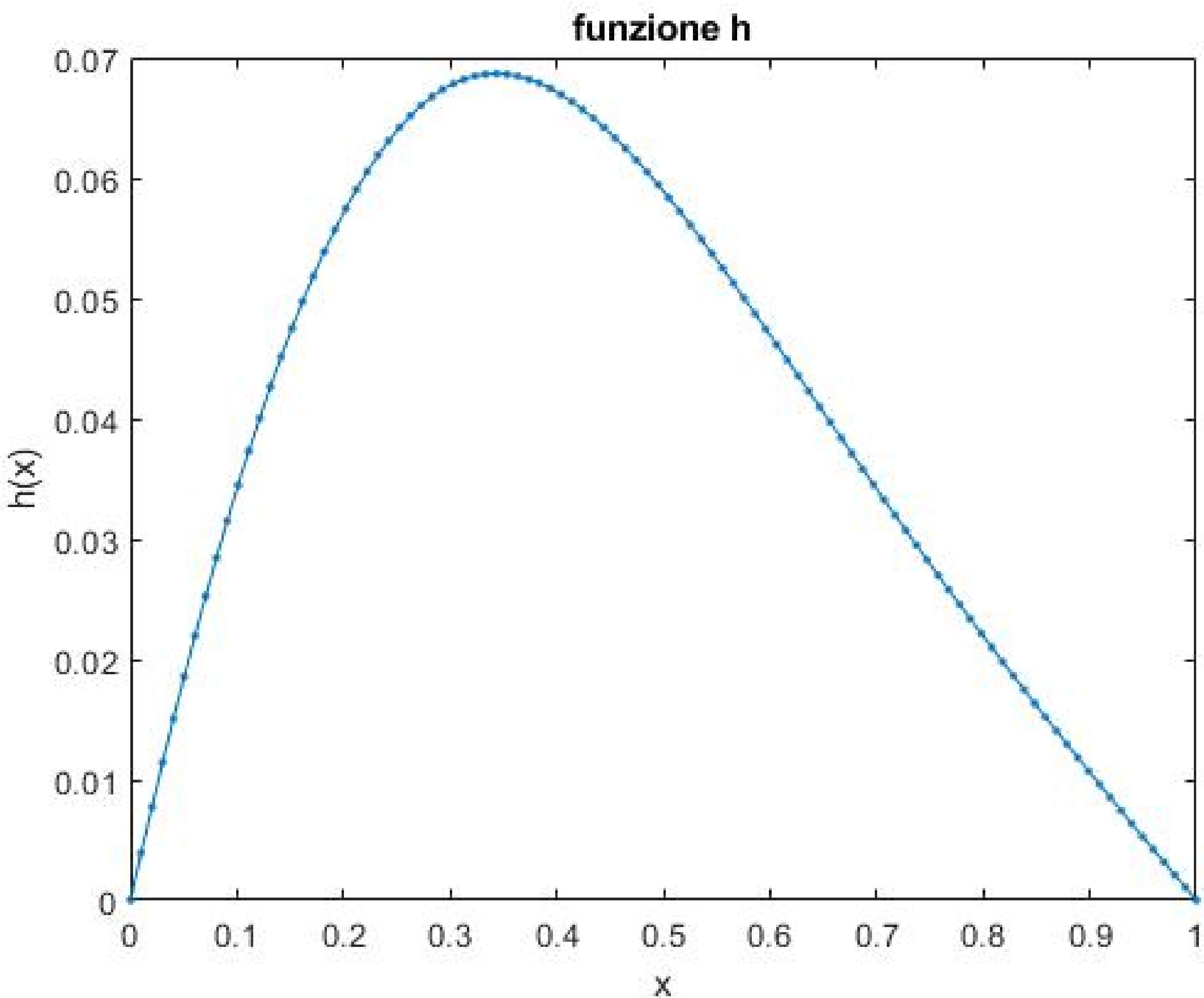
```matlab
%esame 2 Luglio 2015
```

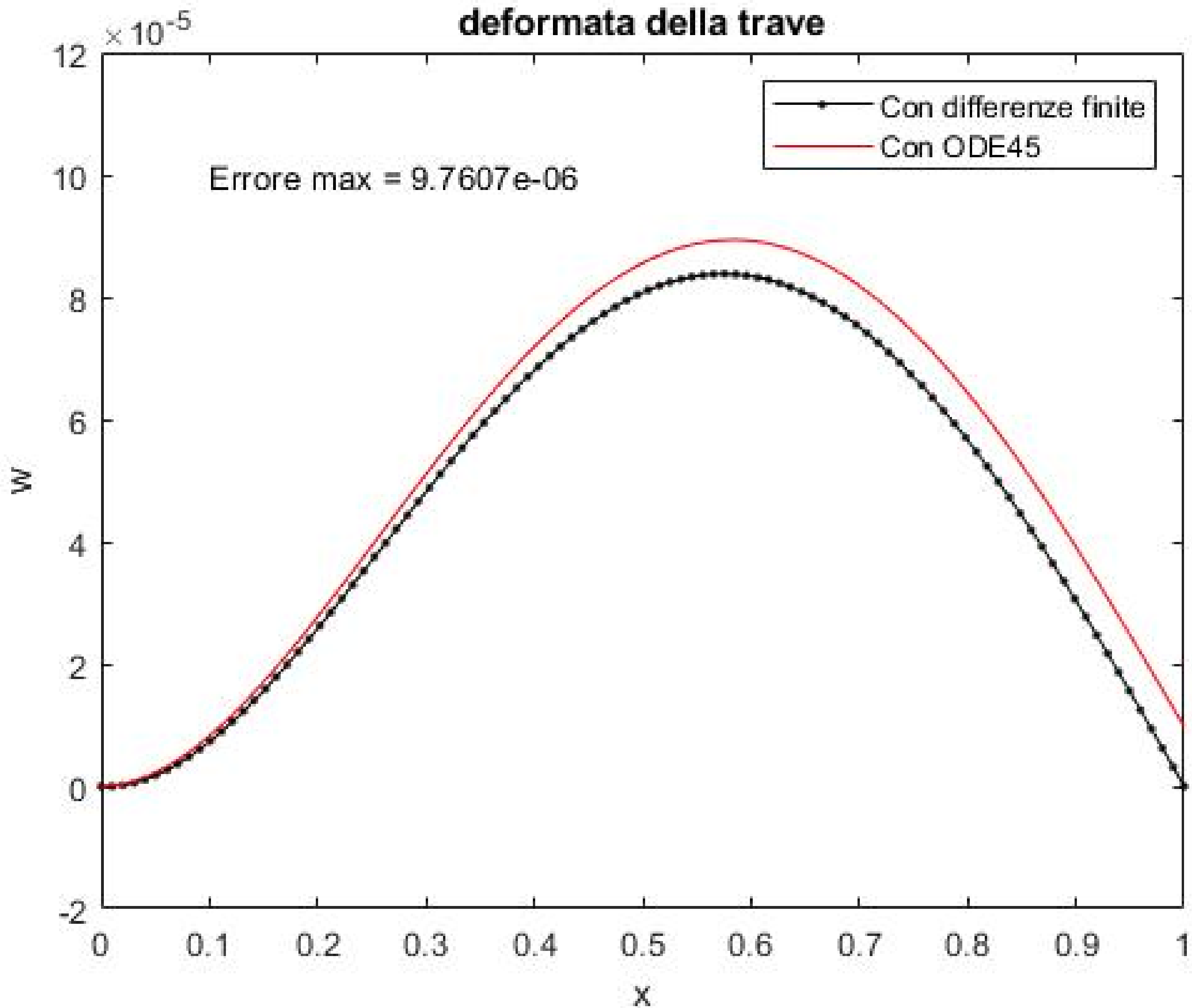
```
close all; clear; clc;

%dati

vN=[50,100,150];
vrapp=[.5,1,5];

L=1;
k=.5;
w=.9;

grafico_ins = 1; %0 per non visualizzare

beta=5;

h=NaN(length(vN),1);
err=NaN(length(vrapp),length(vN));

for irapp=1:length(vrapp)

disp(['iterazione n. ',num2str(irapp),' su
',num2str(length(vrapp))]);
rapp=vrapp(irapp);
a=rapp*k/L;

for iN=1:length(vN)

N=vN(iN);

x=linspace(0,L,N);
dx=x(2)-x(1);
h(iN)=dx;
dt=(beta*dx^2)/k;

phi0=(L/pi)*sin(pi*x/L);
p=x.*(L-x);
dphi_L=1;

%operatori

xs=x(1:3);
xc=x(3);
vb=PesiDer(xs,xc,1);
B=zeros(N,N);
for i=3:N-1
    B(i,i-2:i)=vb;
end
B(2,1:3)=PesiDer(xs,x(2),1);
```

```matlab
xs=x(1:5);
vd=PesiDer(xs,xc,2);
D=zeros(N,N);
for i=3:N-2
    D(i,i-2:i+2)=vd;
end
D(2,1:5)=PesiDer(xs,x(2),2);
D(end-1,end-4:end)=PesiDer(xs,x(4),2);

I=eye(N);

A=I-(k/2)*dt*D;
C=I-(a*dt*B)+(k/2)*dt*D;
q=dt*p;

%BC

A(1,:)=0; A(end,:)=0;
C(1,:)=0; C(end,:)=0;
A(1,1)=1; A(end,end)=1/dx; A(end,end-1)=-1/dx;
q(end)=dphi_L;

%calcolo

T=A\C;

phi1=phi0';
t=0;
dif=1;
tol=1e-4;

while dif > tol

    t=t+dt;
    gamma=1-exp(-w*t);
    q(1,1)=gamma;

    f=A\q';

    phi2 = T*phi1 + f;
    dif=norm(phi2-phi1,inf);
    phi1=phi2;

    if grafico_ins==0 || iN ~= 1 || irapp ~= 1
        continue;
    end
    plot(x,phi0,'-r',x,phi2,'-k');
```

```
    axis([0 L 0 2.7]);
    title(['\phi_2 - \phi_1 = ',num2str(dif,1),...
                                    ' on tollerance =
',num2str(tol)]);
    xlabel('x'); ylabel('\phi');
    drawnow;

end



end



end



syms y(xa) aa ka La

eqn = aa*diff(y) - ka*diff(diff(y)) == xa*(La-xa);
condizione1 = y(0) == 1;
condizione2 = diff(y(La))==1;

y = dsolve (eqn, condizione1, condizione2);
y = subs (y, {xa, aa, ka, La}, {x, a, k, L});

phi_a = eval(y);
```


funzione f
f(x)
x
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


funzione h
h(x)
x
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


deformata della trave
×10^-5
12
10
8
6
4
2
0
-2
w
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
x
Con differenze finite
Con ODE45
Errore max = 9.7607e-06

Metodi Numerici in Ingegneria Aerospaziale

17 Novembre 2016

Si consideri la equazione di convezione lineare:

$$\frac{\partial \varphi}{\partial t} + a\frac{\partial \varphi}{\partial x} = 0 \tag{1}$$

nel dominio $[0, L]$ avente condizioni al contorno periodiche e condizione iniziale:

$$\varphi(x, 0) = e^{\sin(2\pi x/L)}$$

Fissato un mesh uniforme nel dominio $[0, L]$, si discretizzi la equazione assegnata secondo il seguente schema predictor-corrector:

- Si determina preliminarmente una soluzione di previsione φ^* secondo lo schema classico:
$$\frac{\varphi^* - \varphi^n}{\Delta t} = -a\frac{\delta^+ \varphi^n}{\Delta x}$$
- Si calcola uno stadio intermedio secondo la formula:
$$\varphi^{n+1/2} = \frac{1}{2}\left(\varphi^n + \varphi^*\right)$$
- Si corregge la soluzione avanzando di un passo $\Delta t/2$ secondo lo schema:
$$\frac{\varphi^{n+1} - \varphi^{n+1/2}}{\Delta t/2} = -a\frac{\delta^- \varphi^*}{\Delta x}$$

dove naturalmente gli operatori δ^+ e δ^- sono i classici operatori *forward* e *backward* su due punti.

Si programmi la risoluzione numerica della equazione e si analizzino empiricamente le proprietà di stabilità e accuratezza dello schema.

```
close all; clear all; clc;

% Corso di 'Metodi Numerici in Ingegneria Aerospaziale'
% Prof. G. Coppola
%
```