```
------------------- errEulLoc.m --------------------------
%Verifichiamo che l'errore locale del metodo di
% Eulero esplicito sia O(h^2)
%Usiamo come esempio la ODE
%  y' = y(1-y)
%  y(0) = y0 = 0.3
% la cui soluzione esatta è
%  y(t) = y0*exp(t)/(1+y0*(exp(t)-1))

y0=0.3;

H= 2.^ (0:-1:-6);
clear err
for k=1:length(H)
   h=H(k);
   esatta = y0*exp(h)/(1+y0*(exp(h)-1)); % al tempo t=h
   y1 = y0 + h* y0*(1-y0);  %Metodo di E.E.
   err(k) = abs(y1-esatta);
end
loglog(H, err,'o-');
title 'Errore locale del metodo Eulero esplicito'
xlabel 'passo'
ylabel 'errore'

p=polyfit(log(H), log(err) ,1);
disp(sprintf('Ordine di convergenza misurato: %f',p(1)))
----------------------------------------------------------

>> errEulLoc
Ordine di convergenza misurato: 1.921102

% Sufficientemente vicino al valore teorico 2
% Nota: se ripetessimo il test con il metodo di Heun, dovremmo trovare
% un valore prossimo a 3

------------------- errEulGlob.m -------------------------
%Verifichiamo che l'errore globale del metodo di
% Eulero esplicito sia O(h)
%Usiamo come esempio la ODE
%  y' = y(1-y)
```

```
%  y(0) = y0 = 0.3
% la cui soluzione esatta è
%  y(t) = y0*exp(t)/(1+y0*(exp(t)-1))

y0=0.3;
esatta = y0*exp(1)/(1+y0*(exp(1)-1)); % al tempo t=1

clear H err
for k=1:6
   h=2^(-k); % lunghezza del passo
   N=2^k;    % numero dei passi

   %Applico N passi di E.E. di lunghezza h
   y1=y0;
   for i=1:N
      y1 = y1 + h* y1*(1-y1);
   end
   H(k) = h;
   err(k) = abs(y1-esatta);
end
loglog(H, err,'o-');
title 'Errore globale del metodo Eulero esplicito (tfin=1)'
xlabel 'passo'
ylabel 'errore'

p=polyfit(log(H), log(err) ,1);
disp(sprintf('Ordine di convergenza misurato: %f',p(1)))
--------------------------------------------------

>> errEulGlob
Ordine di convergenza misurato: 1.049066

% Sufficientemente vicino al valore teorico 1
% Nota: se ripetessimo il test con il metodo di Heun, dovremmo trovare
% un valore prossimo a 2

--------------- Heun.m ------------------------------
function [T,Y]=Heun(f,y0,tfin,h0)
% Approssima la soluzione di
% y' = f(y) , y(0)=y0
```

```
% per t da 0 a tfin,
% usando il metodo di Heun con passo h0

t=0; % tempo corrente
T=0; % elenco dei tempi t
Y=y0; % elenco dei valori di y(t)
while (t<tfin)
   % 1 passo di Heun lungo h0
   K1 = f(Y(end));
   K2 = f(Y(end)+h0*K1);
   ynew = Y(end) + h0*(0.5*K1+0.5*K2);
   % aggiorno t
   t = t + h0;
   %salvo i valori per l'output
   T(end+1)=t;
   Y(end+1)=ynew;
end
---------------------------------------------

>> T=linspace(0,5);
>> y0=0.3;
>> EXA=y0*exp(T)./(1+y0*(exp(T)-1));
>> plot(T,EXA,'k')
>> hold on
>> f=@(y) y.*(1-y);
>> [t,y]=Heun(f,0.3,5,1);
>> plot(t,y,'bo-')
>> [t,y]=Heun(f,0.3,5,0.5);
>> plot(t,y,'rd-')
>> [t,y]=Heun(f,0.3,5,0.1);
>> plot(t,y,'g.-')

%Notate come nell'ultimo test venga calcolato un passo di troppo:
>> format long
>> t(end-1)
ans =
   4.999999999999998
>> t(end)
ans =
   5.099999999999998
```

```
%i questo caso si tratta di un'accumulazione di errori in t, ma
% potrebbe anche essere causato dall'utente:

>> [t,y]=Heun(f,0.3,5,1/pi );
>> t(end-1)
ans =
   4.774648292756860
>> t(end)
ans =
   5.092958178940650

%Occorre introdurre una modifica per garantire di arrivare al tempo
% finale richiesto dall'utente. Basta un semplice controllo del tipo
if (t+h0>tfin)
   h0 = tfin-t;
end
% all'inizio del ciclo while, che provoca l'accorciamento dell'ultimo
% passo temporale
```