```
close all
clear all

% driver for numerical experiments in homework
%Need to specify h before running it.
%Change only parameter p when using different methods.
% Input parameters:
a = 0; % left side of integration interval
eta = [2.6726e-1; -5.3452e-1; 8.0178e-1]; % initial value (column
 vector)
p = 2; % parameter p of method. p=0: single step, p>0: multistep
h = 0.001025
A = [200 398 198; -500 -696 -296; 500 694 294];
% end of input

global count ;%counter for evaluations of function f(x,y)
count = 0;

m = max(size(eta)); % number of equations
y = zeros(m,p+1);
f=y;
for j=1:p                     %initialize matrices y and f.
  y(:,j+1) = exact_sol_linear_ivp(0,A,0,eta); %The (j+1)-th column of
 y contains y(a+j*h)
  f(:,j+1) = fun(a+j*h,y(:,j+1),A); % compute f(a+j*h,y(a+j*h))
end
y(:,1) = eta;
f(:,1) = fun(a,eta,A);

nstep = 1.0/h;
xn = a+p*h;
for k=1:5 %solve ode. Stop after nstep steps  to compute error
  for n=1:nstep
    [y,f] = method(xn,h,y,f); %compute solution at x+h
    xn = xn+h;
  end
  x(k) = xn-p*h; %x-value where error is computed.
  err(k) = norm(y(:,1) - exact_sol_linear_ivp(x(k),A,0,eta)); %compute
 error
  c(k) = count; %function evaluations needed so far
end
z=[x' err' c'];
disp('      x       error   evaluations of f(x,y)')
disp(z)
```

Published with MATLAB® R2016b