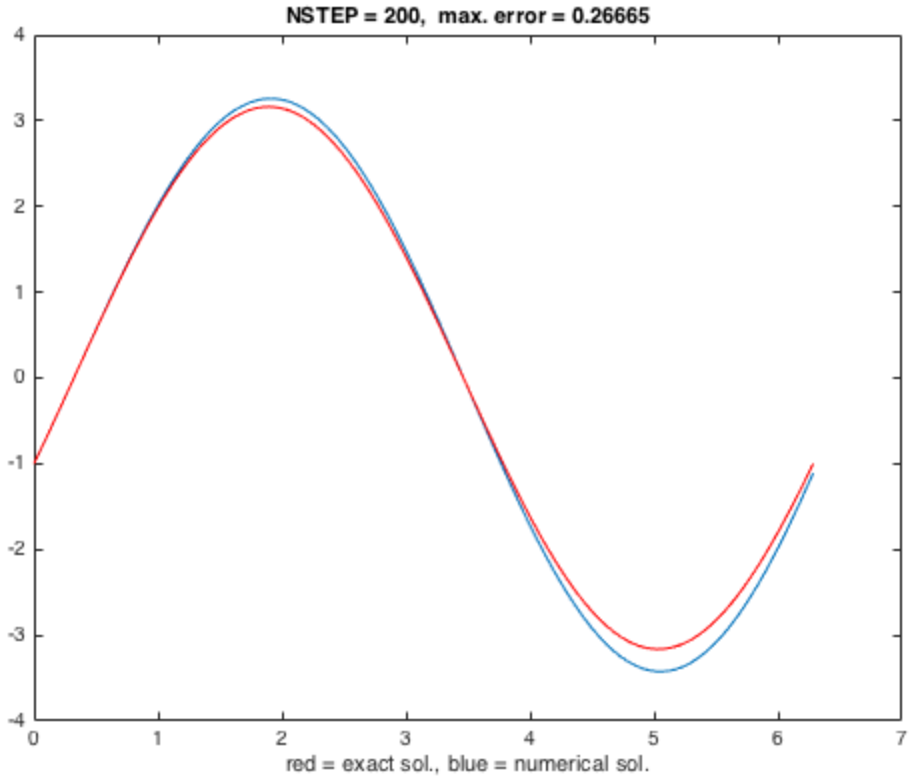
```
function [t,U] = eulerw17(odefun, TSPAN ,U0,NSTEP);
T0 = TSPAN(1); TFINAL = TSPAN(2);
dt = (TFINAL-T0)/NSTEP;
m = length(U0);
U = zeros(m,NSTEP+1);
U(:,1) = U0;
t = T0:dt:TFINAL;
for k = 1:NSTEP
    U(:,k+1) = eulerstep(odefun,t(k),U(:,k),dt);
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function U = eulerstep(odefun,t,U0,dt)
U = U0 + dt*feval(odefun,t,U0);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function yprime = myode1(t,y)
% Implements ODE y''= -y as a first order system
yprime = zeros(size(y));
yprime(1) = y(2);
yprime(2) = -y(1);
end
```

Published with MATLAB® R2015b

```
% Main routine (driver) for Euler method demonstration
odefun = 'myode1'; %Specify name of user supplied
                   % function M-file with rhs of ode
t0 = 0; tfinal = 2*pi; % Specify initial and final times
U0 = [-1;3]; % Specify column vector of initial values
NSTEP=200; % Specify number of steps.
           %Stepsize Delta_t = (tfinal-t0)/NSTEP.


TSPAN = [t0,tfinal];
[t,U] = eulerw17(odefun,TSPAN,U0,NSTEP);

% plot numerical solution;
figure;
hold off;
plot(t,U(1,:));

% If the IVP is the demonstration  IVP
% u'' = -u, u(t0) = U0(1), u'(t0) = U0(2)
% plot the exact solution and find the maximum error
if odefun == 'myode1'
    uexact = U0(1)*cos(t) + U0(2)*sin(t);
    % Compute maximum error
    maxerr = max(abs(uexact-U(1,:)))
    % Plot exact solution
    hold on;
    plot(t,uexact,'r');
    xlabel(['red = exact sol., blue = numerical sol.']);
    title(['NSTEP = ' num2str(NSTEP) ',  max. error = ' ...
            num2str(maxerr)])

end
```

```
maxerr =

    0.2667
```



Published with MATLAB® R2015b