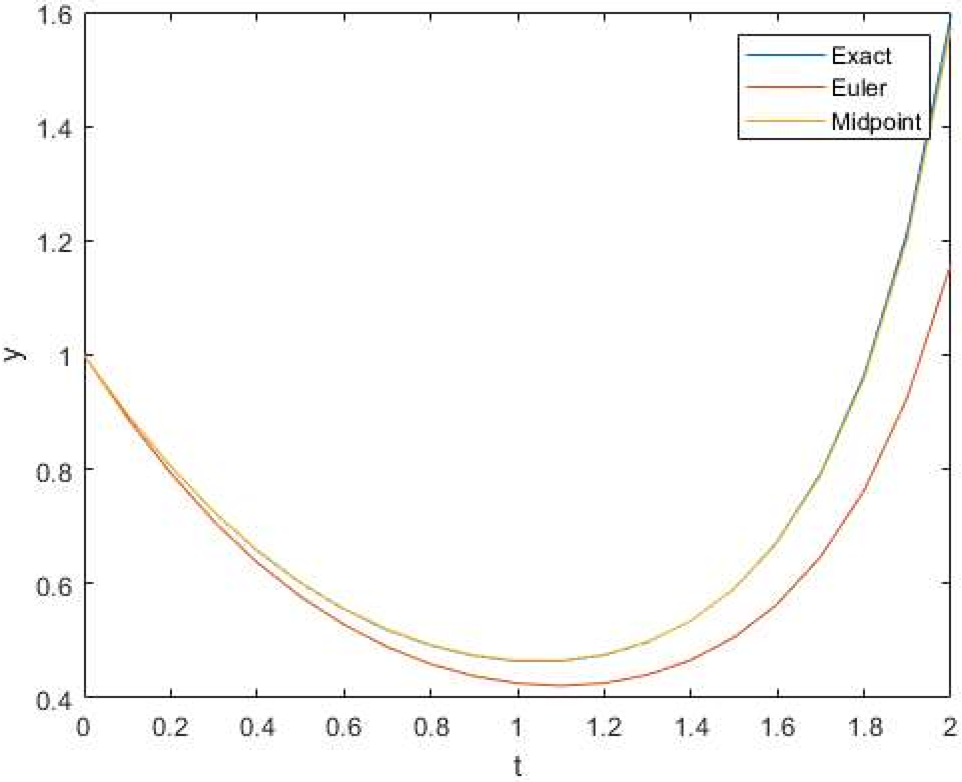
```
clc;
close all;
clear;
h=0.1; % step size
f=@(t,y) y*t^2-1.1*y; % define the right hand side of the dy/dt as an anonymous
                      % function
y0=1; % initial condition

t=0:h:2; % create a vector of t values
y_eu(1)=y0; % Euler solution
y_mid(1)=y0; % mid-point solution

for i=2:length(t) % loop going from 2 not 1 as y0 already stored in the solution vector
    yeu=y_eu(i-1)+h*f(t(i-1),y_eu(i-1)); % Euler solution
    y_eu(i)=yeu; % store into y_eu

    % mid-point solution
    k1=h*f(t(i-1),y_mid(i-1));
    k2=h*f(t(i-1)+h/2,y_mid(i-1)+k1/2);
    ymid=y_mid(i-1)+k2; % mid-point solution
    y_mid(i)=ymid;
end

% exact solution
exact=exp((t.^3-3.3*t)/3);

% now plot
plot(t,exact)
hold on
plot(t,y_eu)
plot(t,y_mid)
xlabel('t')
ylabel('y')
legend('Exact','Euler','Midpoint')
hold off
h=0.5;
y0=1; % initial condition
t0=0;
while t0<2  % t0 can't go upto 2 as at t=2 the solution will be for y(2.5)
    % mid-point solution
    k1=h*f(t0,y0);
    k2=h*f(t0+h/2,y0+k1/2);
    y=y0+k2; % mid-point solution at t=t0+h,solution at t=t0+h depends on t0
    y0=y; % update y0
    t0=t0+h; % update t0
end
disp('y(2)=')
disp(y)
```

```
y(2)=
   1.364266861934474
```



Published with MATLAB® R2021b