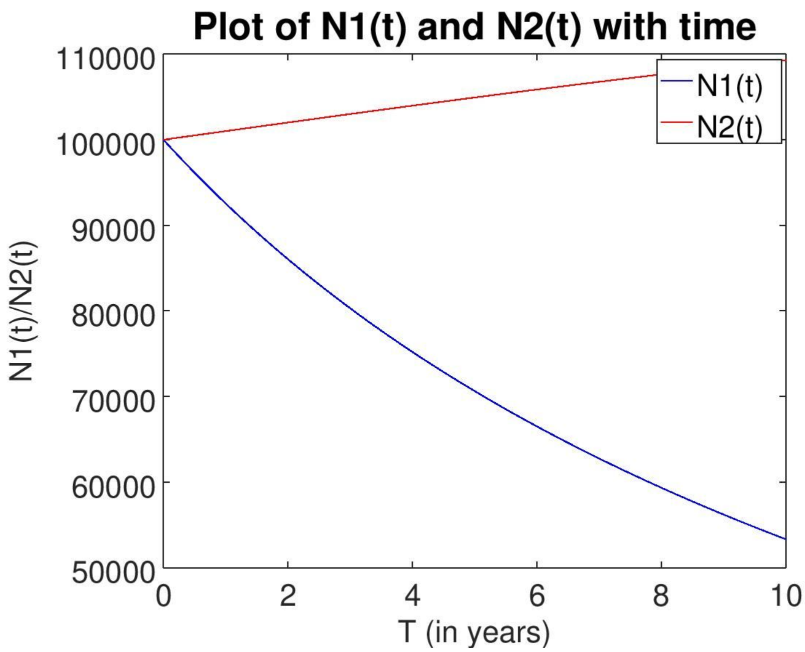
Question 1: Coupled System of equations using RK-4

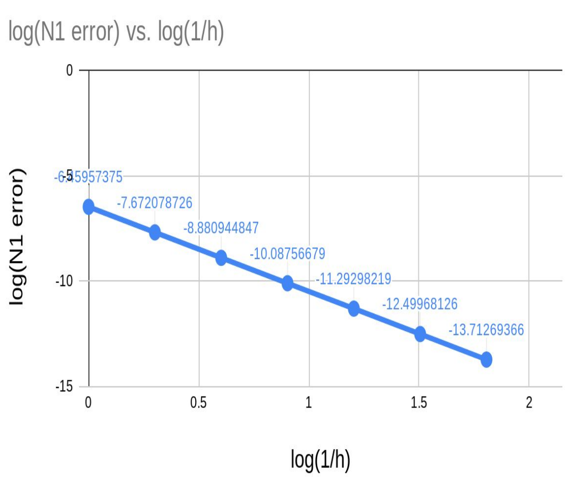
In this question we solve a coupled differential set of equations using the Runge-Kutta order 4 method. The different values of N1(t) and N2(t) have been calculated and plotted as below for time step size of 1/2^7.

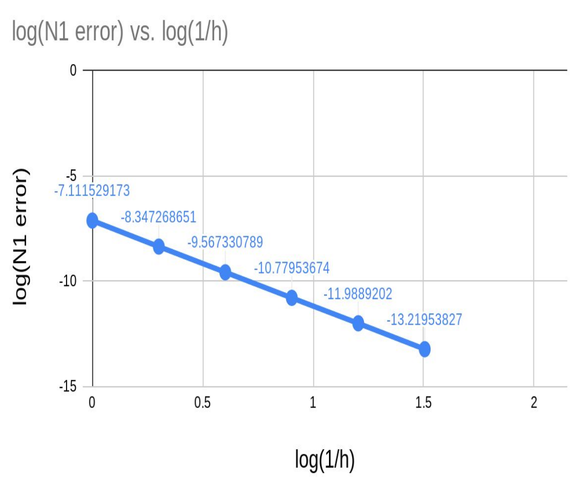


1. The experiment was run for different time step sizes and the value of N1(10) and N2(10) was calculated as shown in the table below. The value of the time step size of .00781(i.e. $1/2^7$) was taken as the one producing the correct result. The value of N1(t) seems to converge at 53317 and N2(t) converges at 109284.
2. In the table below, the value of the step size was varied from h=1 year to h= $1/2^7$ years by successively dividing the step size by 2.
3. By the 8th iteration, i.e. with step size $1/2^7$ the values of N1(10) and N2(10) seem to have converged within a tolerance of 10^{-9}. And this value is taken to be the exact value of the model. The differential equations in this question are supposed to model a population and decimal values have no significance. However, for the purposes of error analysis, the value of the tolerance has been fixed as low as 10^{-9}.
4. The graphs of log of relative errors with the $\log(h^{-1})$ are plotted below. From the graphs, it can be seen that log of relative error decreases with $\log(h^{-1})$ and the curve is a straight line with slope of -4. This is to be expected because for RK-4

methods, the local truncation error is of the order of h^4 and hence it is expected that the slope of the curve is -4.

	Different values of N1(10) and N2(10) with various step sizes			
h (in years)	**1.00000**	**0.50000**	**0.25000**	**0.12500**
N1(10)	53317.812122643	53317.794751717	53317.793687380	53317.793621604
Absolute error of N1	0.0000003470773	0.0000000212775	0.0000000013154	0.0000000000817
N2(10)	109284.019293308	109284.011331219	109284.010869581	109284.010841801
Absolute error of N2	0.0000000773519	0.0000000044950	0.0000000002708	0.0000000000166
h (in years)	**0.06250**	**0.03125**	**0.01563**	**0.00781(=1/2^7)**
N1(10)	53317.793617518	53317.793617263	53317.793617247	53317.793617246
Absolute error of N1	0.0000000000051	0.0000000000003	0.0000000000000	0.0000000000000
N2(10)	109284.010840097	109284.010839992	109284.010839985	109284.010839985
Absolute error of N2	0.0000000000010	0.0000000000001	0.0000000000000	0.0000000000000





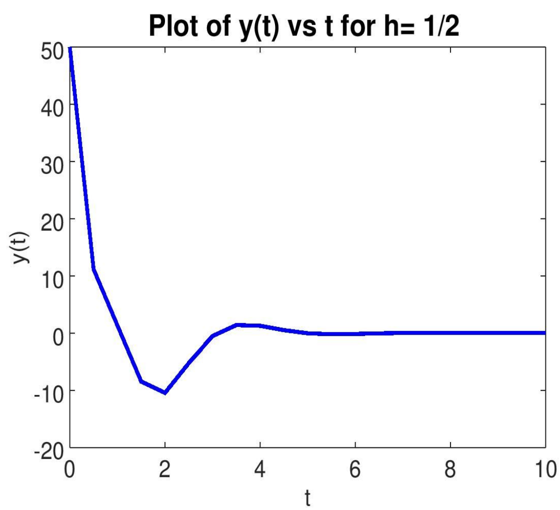
Question 2: Stability Analysis of various methods

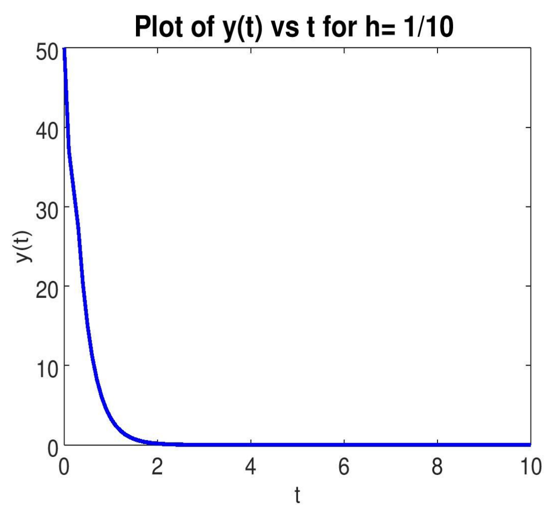
Stability of the various methods have been analyzed analytically for the maximum allowable step size and the results are as follows:

1. RK-2(Midpoint rule): h= 2/3
2. Adams-Bashforth 2-step method: h=1/3
3. Adams-Moulton 2-step method: h=2
4. Adam's 2 step P/C method: h=1/3

Hence from the above it seems that the ranking of the most stable to least stable method for this problem is: **Adam-Molton 2 step > RK-2(Midpoint rule) > Adams-Bashforth 2-step = Adam's 2 step P/C method.**

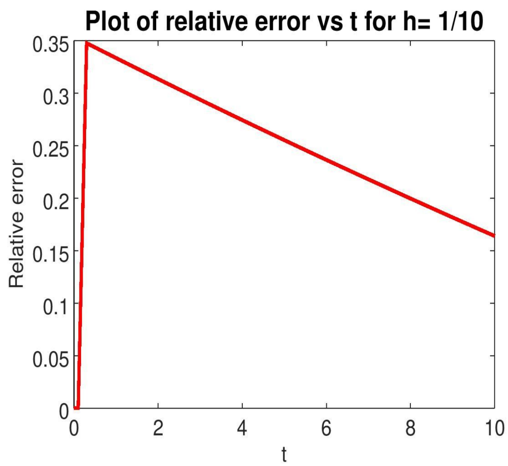
To verify the above analysis, Adams 2 step Predictor/Corrector algorithm was programmed and executed for the problem at hand i.e dy/dt = -3t between t =0 and t= 10. The plots of the function with different values of h are as follows:

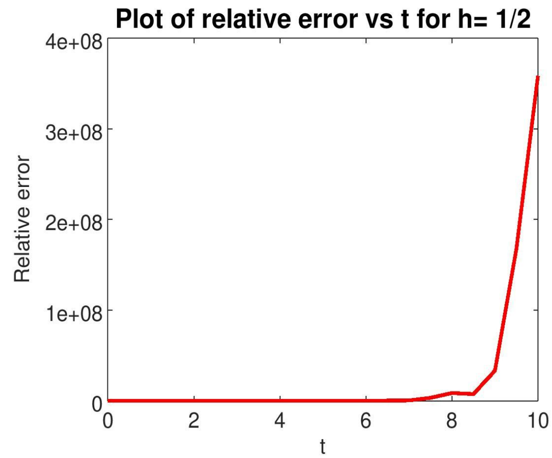


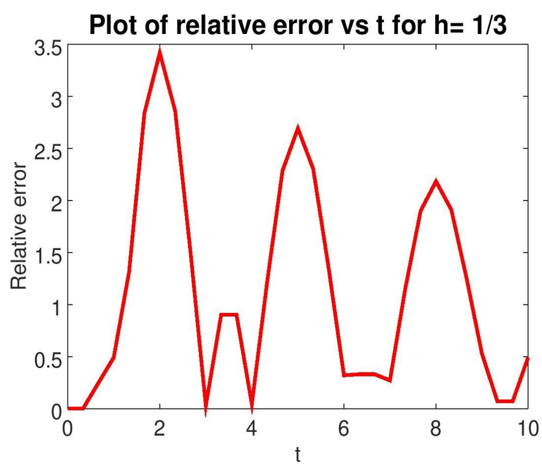


1. For the step size of h= ½ the plot of oscillates and finally converges to $-1.6764*10^{-3}$ which is quite different from the correct value i.e $4.6788*10^{-12}$.Hence the solution does oscillates but does not converge to the true solution.
2. For step sizes h = 1/10 the solution converges to $5.4461*10^{-12}$ which is very close to the exact solution of $4.6788*10^{-12}$.
3. This shows that for step size greater than ⅓ the solution oscillates but for step size less than ⅓ there is no oscillation and the solution converges rapidly.

Errors were also plotted for the above functions and the results are below:







All the codes and handwritten analysis of the different methods are attached as Annexure.

Stability analysis of the equation $\frac{dy}{dt} = -3y(t)$.

1

DS 288 HW6
Avishek Shaw

- Midpoint RK-2

$$w_{i+1} = w_i + h f\left(w_i + \frac{h}{2} f_i, t_i + \frac{h}{2}\right)$$

$$= w_i + h\left[-3\left(w_i + \frac{h}{2} f_i\right)\right]$$

$$= w_i\left[1 - 3h\right] - \frac{3h^2}{2} f_i$$

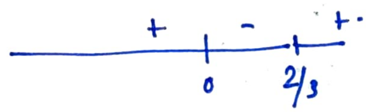
$$= w_i\left[1 - 3h\right] - \frac{3h^2}{2}(-3w_i)$$

$$= w_i\left[1 - 3h + \frac{9h^2}{2}\right]$$

We want $\frac{9h^2}{2} - 3h + 1 < 1$

or $\frac{9h^2}{2} - 3h < 0$

or $h\left(\frac{9h}{2} - 3\right) < 0$



$\therefore \cancel{h < 0 \text{ or } h > \frac{2}{3}}$

$\therefore h \in \left(0, \frac{2}{3}\right)$

Adam Bashforth 2 step method

$$w_{i+1} = w_i + \frac{h}{2}\left[3f_i - f_{i-1}\right] \qquad f(t) = -3y(t).$$

$$= w_i + \frac{h}{2}\left[3(-3w_i) - (-3w_{i-1})\right]$$

$$w_{i+1} = w_i\left(1 - \frac{9h}{2}\right) + w_{i-1}\left(\frac{3h}{2}\right)$$

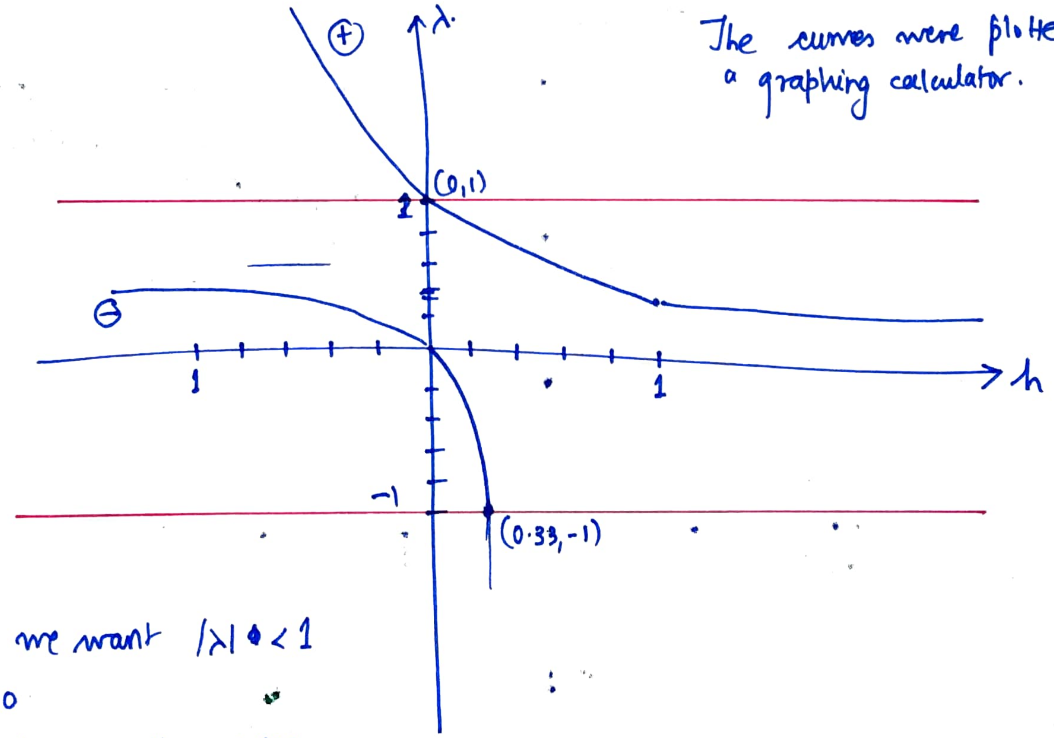
Assume $w_i = k\lambda^i$

$$\therefore \cancel{k}\lambda^{i+1} = \cancel{k}\lambda^i\left(1 - \frac{9h}{2}\right) + \cancel{k}\lambda^{i-1}\left(\frac{3h}{2}\right).$$

or $$2\lambda^2 - \lambda(2 - 9h) - 3h = 0.$$

$$\lambda = \frac{(2 - 9h) \pm \sqrt{81h^2 + 4 - 36h + 24h}}{4}$$

The curves were plotted with a graphing calculator.



- Basically we want $|\lambda| < 1$ and $h > 0$

$\therefore$ from the graph we see that the ~~correct~~ step size interval for which the method is stable is ~~(0, 1/3)~~ $(0, 1/3)$

Adam's Moulton 2 Step Method

$$y_{i+1} = y_i + \frac{h}{12}\left[5f_{i+1} + 8f_i - f_{i-1}\right]$$

$$w_{i+1} = w_i + \frac{h}{12}\left[5(-3w_{i+1}) + 8(-3w_i) - (-3w_{i-1})\right]$$

$$\therefore\ w_{i+1}\left(1 + \frac{\cancel{15}^{5}h}{\cancel{12}_{4}}\right) + w_i\left(\frac{\cancel{24}^{8}h}{\cancel{12}_{4}} - 1\right) - \frac{\cancel{3}h}{\cancel{12}_{4}} w_{i-1} = 0.$$

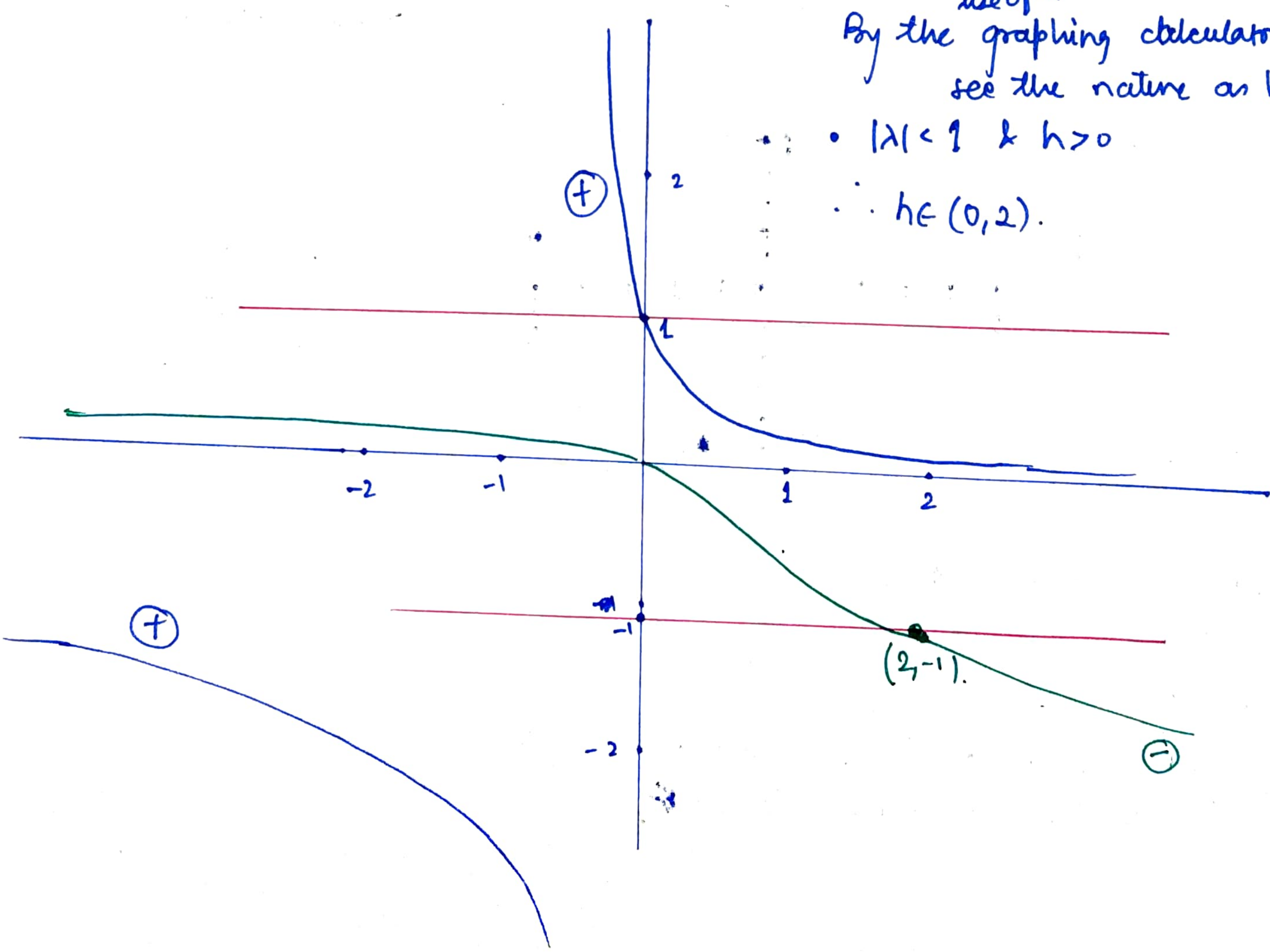
~~$w_{i+1}(12 + 15h) + w_i$~~

$$w_{i+1}(4 + 5h) + w_i(8h - 4) - h w_{i-1} = 0.$$

$$w_i = k\lambda^i$$

$$\lambda^2(4+5h) + \lambda(8h-4) - h = 0$$

$$\lambda = \frac{(4-8h) \pm \sqrt{64h^2 + 16 - 64h + 16h + 20h^2}}{2(4+5h)}$$



By the use of a graphing calculator, we see the nature as below.

- $|\lambda| < 1$ & $h > 0$

$\therefore\ h \in (0, 2)$.

Adam Bashforth Predictor Corrector Order 2

It is composed of 2 steps: for prediction & correction.
Prediction is done with Adams Bashforth. & correction is done with Adams-Molton.

The maximum allowable step sizes for the two are 1/3 & 2 respectively.
Hence for P/C method of order 2, the maximum allowable stepsize will be 1/3. The same can be corroborated by the code by varying the step sizes.

```
#Assgn 6, Q1
clear
output_precision(16)
# System Parameters
a1=0.1;a2=0.1;b1=8*power(10,-7);b2=8*power(10,-7);c1=power(10,-6);c2=power(10,-7);
f=@(n1,n2) [n1*(a1-b1*n1-c1*n2); n2*(a2-b2*n2 -c2*n1)];

#m=no. of equations; N= no. of time intervals; a,b =start and end points resp.
#init=array of initial conditions.
m=2;N=10*2^7;a=0;b=10;init=[power(10,5),power(10,5)];
n1(1)=init(1);n2(1)=init(2); #n1 and n2 are the arrays storing values at different t
#h=time step size,t=current instant of time, T is the iterator which stores the time values
h=(b-a)/N;t=a;T(1)=a;
for j=1:1:m
  w(j)=init(j);
endfor
for i=1:1:N
  for j=1:1:m
    k1(j)=h*f(w(1),w(2))(j);
  endfor
  for j=1:1:m
    k2(j)=h*f((w(1)+0.5*k1(1)),(w(2)+0.5*k1(2)))(j);
  endfor
  for j=1:1:m
    k3(j)=h*f((w(1)+0.5*k2(1)),w(2)+0.5*k2(2))(j);
  endfor
  for j=1:1:m
    k4(j)=h*f((w(1)+k3(1)),w(2)+k3(2))(j);
  endfor
  for j=1:1:m
    w(j)=w(j)+(k1(j)+2*k2(j)+2*k3(j)+k4(j))/6;
  endfor
  t=a+i*h;
  T(end+1)=t;
  n1(end+1)=w(1);
  n2(end+1)=w(2);
endfor

plot(T,n1,'-b',T,n2,'-r')
xlabel("T (in years)","Fontsize",20)
ylabel("N1(t)/N2(t)","Fontsize",20)
title("Plot of N1(t) and N2(t) with time","Fontsize",25)
set(gca,'fontsize',20)
legend('N1(t)','N2(t)')
n1(end)
n2(end)
```

```
# This code is to approximate the solution to the equation y'=3y by
# Adams fourth order predictor corrector methods

#a=start point, b=end point, N= no. of intervals, f is the solution to the\
# equation, h = step size, w0= starting value, w1=2nd value obtained from
# the analytic equation.
clear
clc
g=@(t) 50*exp(-3*t);
f=@(y) -3*y

a=0;b=10;N=100;h=(b-a)/N;
w0=50;w1=g(h);w(1)=w0;w(2)=w1;
T(1)=0;T(2)=h;
Exact(1)=g(0);Exact(2)=g(h);
for i=3:1:N
  t=a+i*h;
  T(i)=t;
  Exact(end+1)=g(t);
  W=w1+0.5*h*(3*f(w1)-f(w0));
  W=w1+h*(5*f(W)+8*f(w1)-f(w0))/12;
  w(end+1)=W;
  w0=w1;
  w1=W;
endfor
w(end)
Exact(end)
#plot of error
#plot(T,abs(1-w./Exact),'-r','linewidth',3)
plot(T,w,'-b','linewidth',3)
xlabel("t","Fontsize",20)
ylabel("y(t)","Fontsize",20)
title("Plot of y(t) vs t for h= 1/2 ","Fontsize",25)
set(gca,'fontsize',20)
```