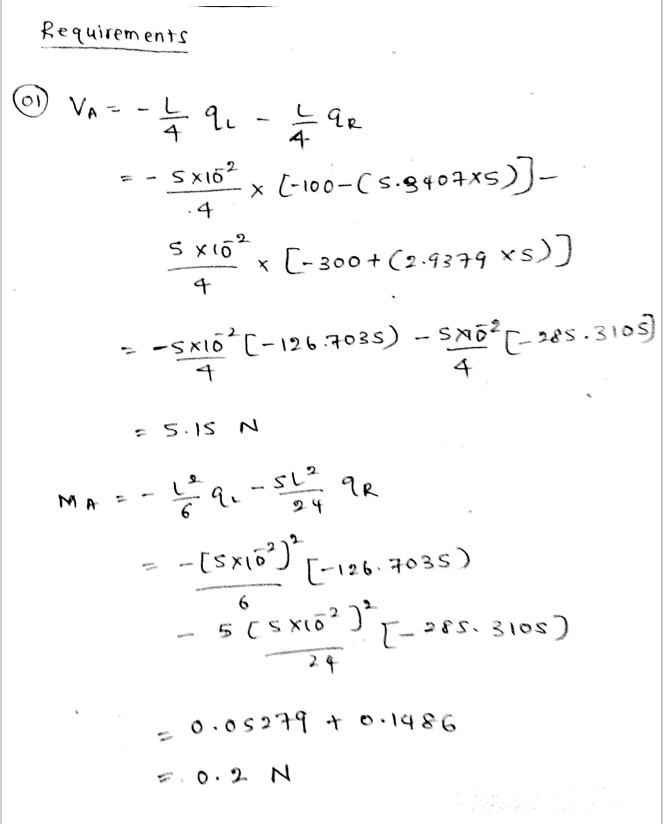
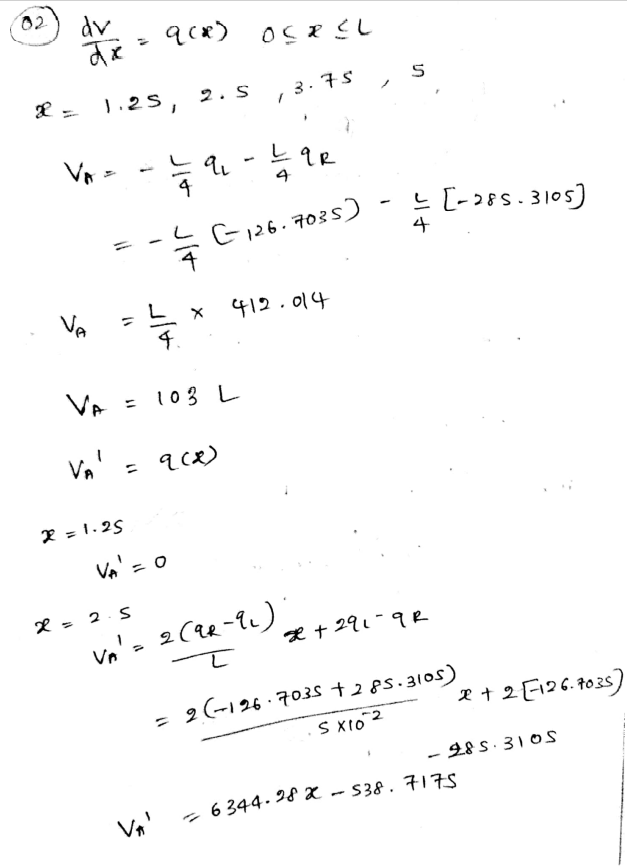
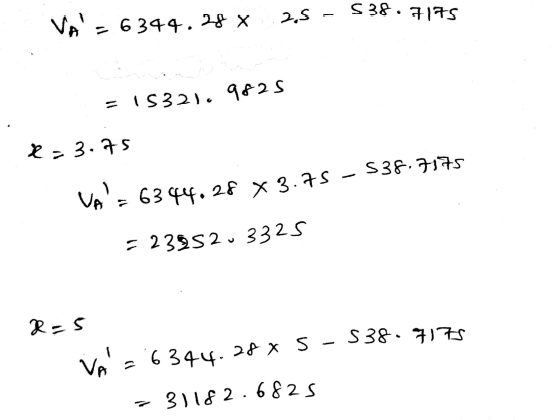
Requirements

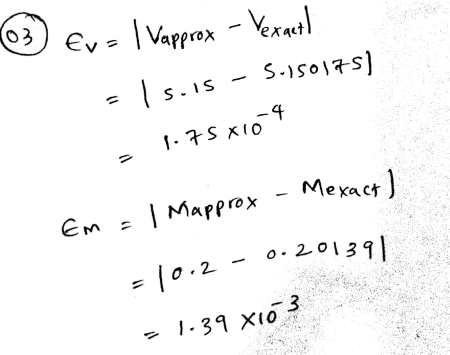
01



02



03



04.

Matlab code

h=1.25;

x=0:h:5;

y=zeros(size(x));

y(1)=1;

n=numel(y);

for i = 1:n-1

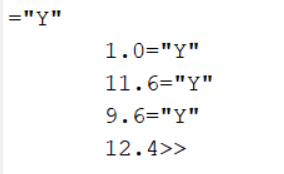
dydx= -2\*x(i).^3 +12\*x(i).^2 -20\*x(i)+8.5 ;

y(i+1) = y(i)+dydx\*h ;

fprintf('="Y"\n\t %0.01f',y(i));

end

output



05.

Matlab code

h=0.05;

x=0:h:5;

y=zeros(size(x));

y(1)=1;

n=numel(y);

for i = 1:n-1

dydx= -2\*x(i).^3 +12\*x(i).^2 -20\*x(i)+8.5 ;

y(i+1) = y(i)+dydx\*h ;

fprintf('="Y"\n\t %0.01f',y(i));

end

output

="Y"

1.0="Y"

1.4="Y"

1.8="Y"

2.1="Y"

2.4="Y"

2.7="Y"

2.9="Y"

3.1="Y"

3.2="Y"

3.3="Y"

3.4="Y"

3.5="Y"

3.5="Y"

3.5="Y"

3.5="Y"

3.5="Y"

3.5="Y"

3.4="Y"

3.4="Y"

3.3="Y"

3.3="Y"

3.2="Y"

3.1="Y"

3.0="Y"

2.9="Y"

2.8="Y"

2.8="Y"

2.7="Y"

2.6="Y"

2.5="Y"

2.5="Y"

2.4="Y"

2.3="Y"

2.3="Y"

2.3="Y"

2.2="Y"

2.2="Y"

2.2="Y"

2.2="Y"

2.2="Y"

2.2="Y"

2.2="Y"

2.3="Y"

2.3="Y"

2.4="Y"

2.4="Y"

2.5="Y"

2.6="Y"

2.7="Y"

2.8="Y"

2.9="Y"

3.0="Y"

3.1="Y"

3.2="Y"

3.4="Y"

3.5="Y"

3.6="Y"

3.8="Y"

3.9="Y"

4.0="Y"

4.2="Y"

4.3="Y"

4.4="Y"

4.5="Y"

4.6="Y"

4.7="Y"

4.8="Y"

4.8="Y"

4.9="Y"

4.9="Y"

4.9="Y"

4.9="Y"

4.9="Y"

4.8="Y"

4.7="Y"

4.6="Y"

4.4="Y"

4.2="Y"

4.0="Y"

3.7="Y"

3.4="Y"

3.0="Y"

2.6="Y"

2.1="Y"

1.6="Y"

1.0="Y"

0.3="Y"

-0.4="Y"

-1.2="Y"

-2.1="Y"

-3.1="Y"

-4.1="Y"

-5.2="Y"

-6.4="Y"

-7.7="Y"

-9.1="Y"

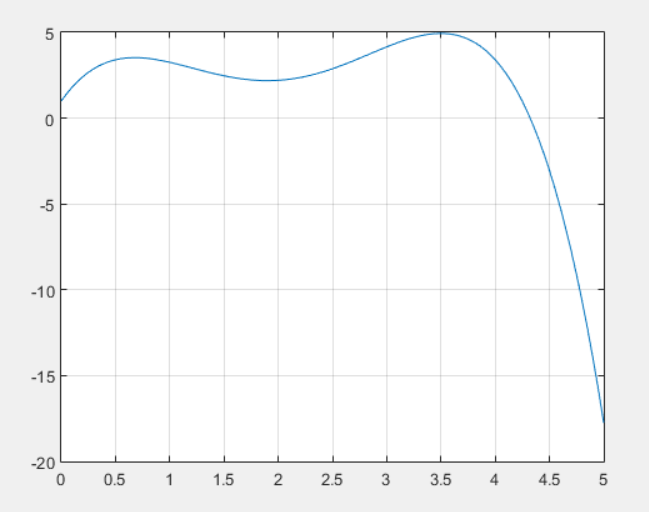
-10.6="Y"

-12.3="Y"

-14.0="Y"

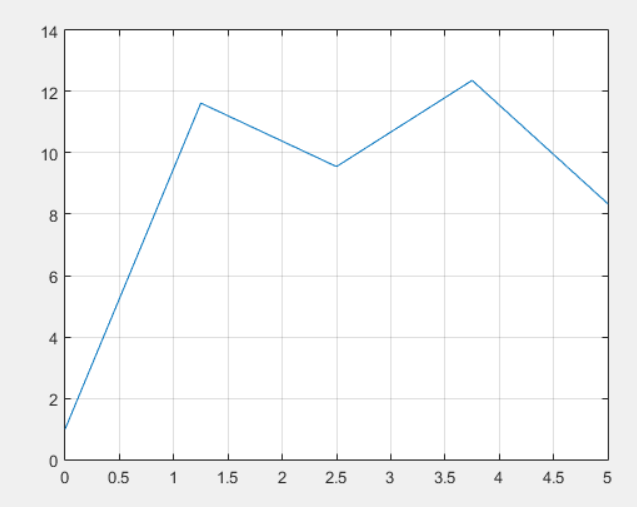
-15.8>>

06.



The output is varying not continuously.

07.



The output is varying continuously.

08.

Matlab code

% y = [R;L;E;V] = [y(1);y2;y(3);y(4)]

y0 = [200;0;0;4\*10^-7]

f = @(t,y) [0.272 - 0.00136\*y(1) - 0.00027\*y(1)\*y(4);

2.7e-05\*y(1)\*y(4) - 1.36e-3\*y(2) - 3.6e-2\*y(2);

0.243e-3\*y(1)\*y(4) + 3.6e-2\*y(2) - 0.33\*y(3);

100\*y(3) - 2\*y(4)];

t = 0;

ts = 120;

h = 1/18;

y = y0;

yout = y0;

while t < ts

y = y + h\*f(t,y);

t = t + h;

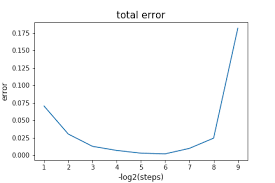
yout = [yout y];

end

v = (0:1/18:120)

f1 = 800 + yout(1,:) + yout(2,:) + yout(3,:)

Output



09.

Matlab code

odeFun = @(t,x) [ x(2);

-2\*x(1)-x(2)];

x0 = [5,1];

span = [0,10];

[t,state]=ode45(fun,span,x0);

% Grab last f' value to use in bvp4c

df\_end = state(end,2);

f0 = x0(1);

% Make boundary conditions function

bc = @(ya,yb) [ya(1)-f0;

yb(2)-df\_end];

% Solve

solinit = bvpinit(t,[0,1]);

sol = bvp4c(odeFun,bc,solinit);

y = deval(sol,t);

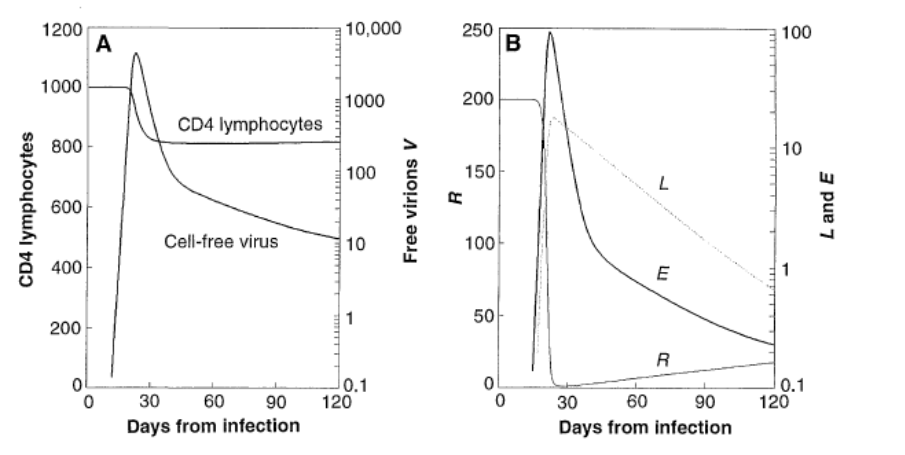
figure;

plot(t,state(:,1)); hold on;

plot(t,y(1,:),'--');

legend('ode45','bvp4c');

Output



10.

Matlab code

syms u(x)

Du = diff(u,x);

D2u = diff(u,x,2);

ode = diff(u,x,3) == u;

cond1 = u(0) == 1;

cond2 = Du(0) == -1;

cond3 = D2u(0) == pi;

conds = [cond1 cond2 cond3];

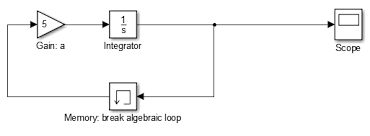
uSol(x) = dsolve(ode,conds)

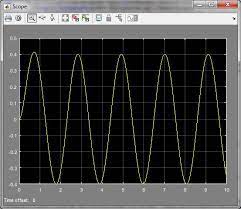
Output

uSol(x) =

(pi\*exp(x))/3 - exp(-x/2)\*cos((3^(1/2)\*x)/2)\*(pi/3 - 1) -...

(3^(1/2)\*exp(-x/2)\*sin((3^(1/2)\*x)/2)\*(pi + 1))/3





11

A sort of variable measurement scale that is quantitative in nature is the ratio scale. It enables any researcher to contrast the ranges or variations. The fourth level of measurement, the ratio scale, has a zero point or character of origin. Nominal, ordinal, interval, and ratio scales are the four types of scales that can be used to categorize data. Every level of measurement has certain significant characteristics that are vital to understand. For instance, the ratio scale is the only one with meaningful zeros.

12

Calculated values has proved by the matlab also. And also we can see graphically solutions in matlab. But using hand calculation we can’t see that.