Contents

[List of Figure 2](#_Toc56113963)

[Euler Method 3](#_Toc56113964)

[Task 01 3](#_Toc56113965)

[Semi implicit Euler 3](#_Toc56113966)

[Task 02 6](#_Toc56113967)

[Task 03 7](#_Toc56113968)

[Task04 9](#_Toc56113969)

[Appendix 11](#_Toc56113970)

[Task 01 11](#_Toc56113971)

[A 11](#_Toc56113972)

[C 11](#_Toc56113973)

[Task 02 12](#_Toc56113974)

[Task 03 12](#_Toc56113975)

[Task04 13](#_Toc56113976)

[b) 13](#_Toc56113977)

[c) 13](#_Toc56113978)

# List of Figure

[Figure 1:Euler method 3](#_Toc56113979)

[Figure 2: semi implicit Euler in function 4](#_Toc56113980)

[Figure 3: semi implicit Euler 4](#_Toc56113981)

[Figure 4: real roots 4](#_Toc56113982)

[Figure 5: matlab semi implicit Euler 5](#_Toc56113983)

[Figure 6: semi implicit Euler for N=100 ,1000 6](#_Toc56113984)

[Figure 7: discreate energy for N=100,1000 7](#_Toc56113985)

[Figure 8: discrete energy 7](#_Toc56113986)

[Figure 9: energy compare at N=100 8](#_Toc56113987)

[Figure 10: energy compare at N=1000 9](#_Toc56113988)

[Figure 11: beta=3 10](#_Toc56113989)

[Figure 12: beta=10 11](#_Toc56113990)

# Euler Method

Euler method is the famous method which are used for the first order different equation on the based of initial condition. Euler method basically the linear approximation in which we draw the tangent line over the short distance to approximate the solution by using the initial value.

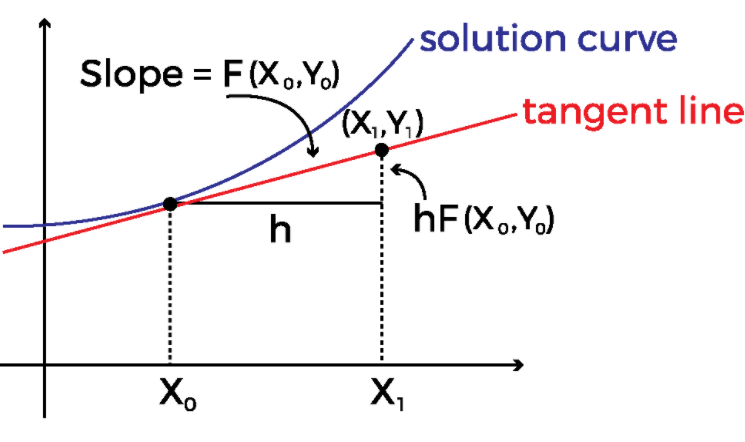
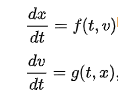


Figure :Euler method

# Task 01

## Semi implicit Euler

Semi implicit Euler method apply on pair of equation.



In these equation x and v maybe scaler. F and g are two give function which we wan to use in our method.



For the solution of the equation we need initial value as same we used in simple Euler method.



The equation of the Euler method for the discreate solution is



Figure : semi implicit Euler in function

Δ*t* is the time step and we can calculate the T=t0+n Δ*t* is the total time.

The main difference between Euler and semi Euler method is that in Euler method we used xn and in the semi Euler method we used xn+1



Figure : semi implicit Euler

In the part we can that xn+1 is explicit part and it used to derived from the implicit part vn+1.

for the real roots it must be s> -2/ Δ*t*

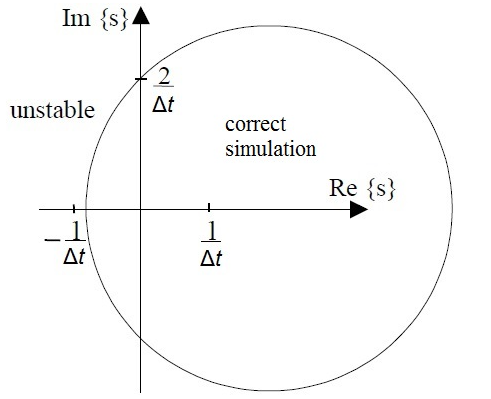


Figure : real roots

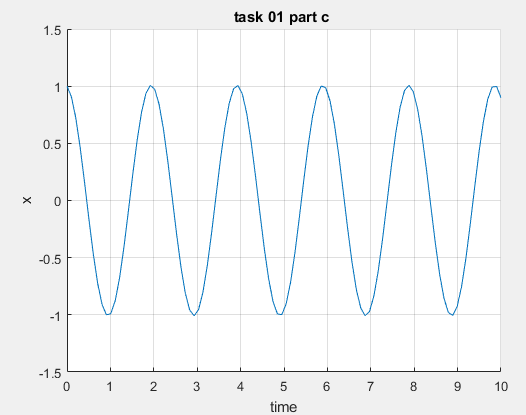


Figure : matlab semi implicit Euler

We can check both ideal case circle almost match with matlab code.

# Task 02

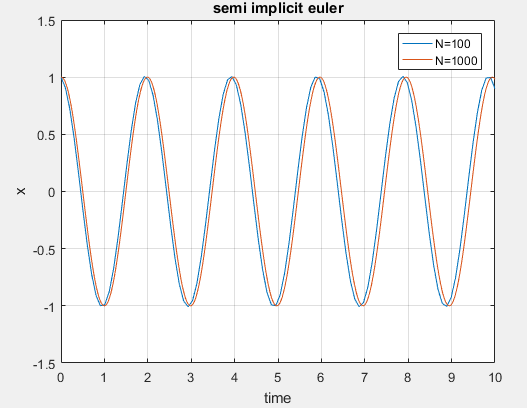


Figure : semi implicit Euler for N=100 ,1000

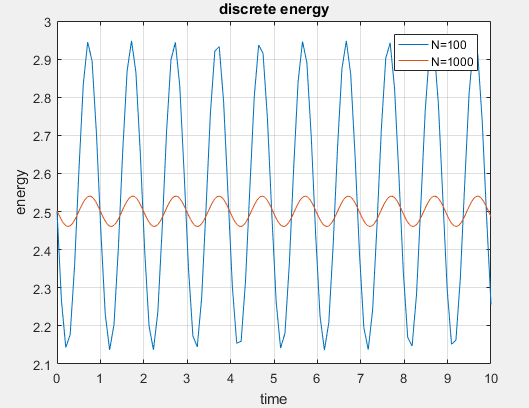


Figure : discreate energy for N=100,1000

The performance of the implicit and explicit are observe from plot. Implicit is more stable and convergence other than Euler. Second observation observed from the graph is discreate energy. The discrete energy calculates by using the formula and it only depend on the value of the semi-implicit Euler.



Figure : discrete energy

We can compare both discrete energies when we increase the N then energy is decrease. Length of the graph is not matter it shown that number of values. Energy become constant after some interval and repeat itself. Energy value is change as semi-implicit vectors value changes.

# Task 03

The modified energy equation is used in the semi-implicit Euler method.



We can prove this by using the K and m equal to 1.

Simple energy equation become when we take ½ common from Equation is

E=1/2 (V2 +x2)

Modify energy equation become when put m and k is 1

Mod\_E=1/2(v2 + x2 - Δ*txnvn* )

Δ*t* is too much small value and we can ignore this part and both equations become equal.

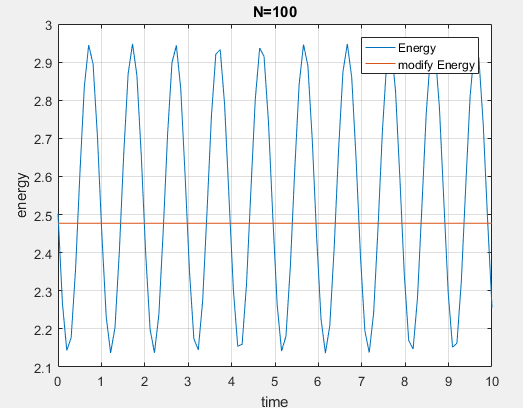


Figure : energy compare at N=100

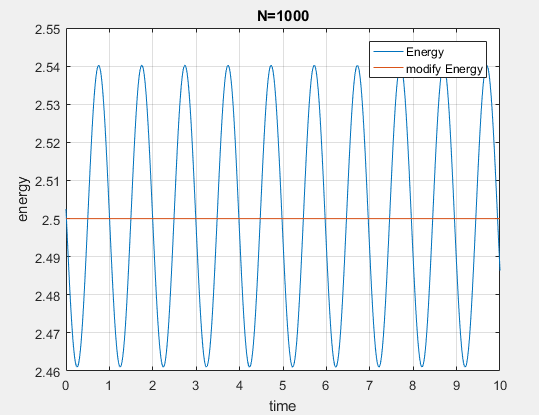
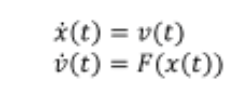


Figure : energy compare at N=1000

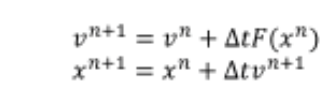
The modify energy become constant and do not change with N.

# Task04

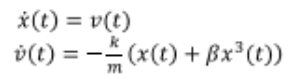
For the second order we have these equation are



simplistic Euler method



Non-linear equation for the spring mass system



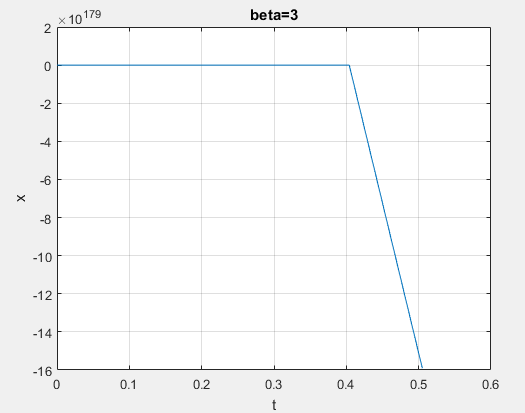


Figure 11: beta=3

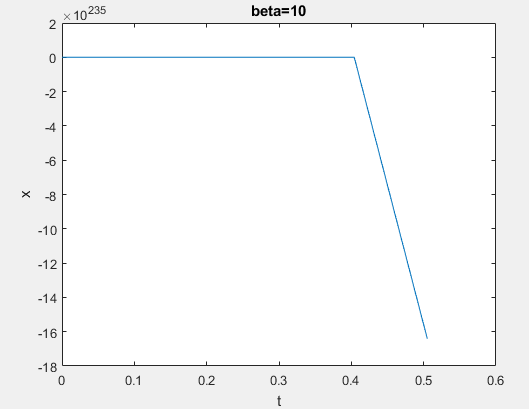


Figure 12: beta=10

By increase the beta value overall value of V become more and more small. We can obverse the value on y-axis as well in both graph.

# Appendix

## Task 01

### A

function [x,v]=Task01(x0,v0,k,m,T,N)

x=[x0];

v=[v0];

time=linspace(0,T,N);

t=time(2)-time(1);

for i=2:length(time)

v\_Temp=v(i-1)-t\*(k/m)\*x(i-1);

v=[v v\_Temp];

x=[x x(i-1)+t\*v\_Temp];

end

end

### C

clc

clear

close all

k=5;

m=0.5;

T=10;

N=100;

v0=.1;

x0=1;

time=linspace(0,T,N);

[x,v]=Task01(x0,v0,k,m,T,N);

hold on

plot(time,x)

grid

xlabel('time')

ylabel('x')

title('task 01 part c ')

## Task 02

clc

clear all

close all

k=5;

m=0.5;

T=10;

N=100;

v0=.1;

x0=1;

time=linspace(0,T,N);

[x\_n100,v\_n100]=Task01(x0,v0,k,m,T,N);

plot(time,x\_n100)

grid

xlabel('time')

ylabel('x')

title('semi implicit euler')

hold on

N=1000;

time2=linspace(0,T,N);

[x\_n1000,v\_n1000]=Task01(x0,v0,k,m,T,N);

plot(time2,x\_n1000)

legend('N=100','N=1000')

Engergy\_100=(1/2)\*m\*(v\_n100).^2 + (1/2)\*k\*(x\_n100).^2 ;

Engergy\_1000=(1/2)\*m\*(v\_n1000).^2 + (1/2)\*k\*(x\_n1000).^2 ;

figure

plot(time,Engergy\_100)

hold on

plot(time2,Engergy\_1000)

legend('N=100','N=1000')

xlabel('time')

ylabel('energy')

title('discrete energy')

grid on

## Task 03

clc

clear all

close all

k=5;

m=.5;

T=10;

N=[100 1000];

v0=.1;

x0=1;

for i=1:length(N)

[x,v]=Task01(x0,v0,k,m,T,N(i));

time=linspace(0,T,N(i));

t=time(2)-time(1);

Engergy=(1/2)\*m\*(v).^2 + (1/2)\*k\*(x).^2 ;

mod\_E=(1/2)\*(m\*v.^2 +k\*x.^2 - t\*k\*x.\*v);

time=linspace(0,T,N(i));

figure(i)

plot(time,Engergy)

hold on

plot(time,mod\_E)

grid

legend('Energy','modify Energy')

xlabel('time')

ylabel('energy')

title(['N=',num2str(N(i))])

end

## Task04

### b)

function [x,v]=Task04(x0,v0,k,m,T,N,beta)

x=[x0];

v=[v0];

time=linspace(0,T,N);

t=time(2)-time(1);

for i=2:length(time)

v\_Temp=v(i-1)-t\*(k/m)\*x(i-1);

v=[v v\_Temp];

x=[x (k/m)\*(v\_Temp+beta\*v\_Temp^3)];

end

end

### c)

clc

clear

close all

k=5;

m=0.5;

T=10;

N=100;

v0=.1;

x0=1;

time=linspace(0,T,N);

beta=3;

[x,v]=Task04(x0,v0,k,m,T,N,beta);

plot(time,x)

title('beta=3')

grid

xlabel('t')

ylabel('x')

figure

beta=10;

[x,v]=Task04(x0,v0,k,m,T,N,beta);

plot(time,x)

xlabel('t')

ylabel('x')

title('beta=10')