Assignment3: NMM

2) Explicit Euler:

Program Commands:

```
function [T,Y] = vEuler(t0,tf,y0,h)
% [T,Y] contains information about each variable at each time step
% The ceil function rounds to the highest integer.
n=length(y0); nsteps=ceil((tf-t0)/h);
P=zeros(nsteps+1,n+1);
t=t0;y=y0;
T(1,1)=t0; Y(1,:)=y0';
for i=1:nsteps
% Make sure we hit tf on last step.
if i==nsteps
h=tf-t;
end
y=y+h*f(y);
t=t+h;
T(i+1,1)=t; Y(i+1,:)=y';
end
%
function rhs=f(y)
rhs(1,1)=y(1,1)*(1-y(1,1))-y(1,1)*y(1,2);
rhs(1,2)=2*y(1,1)*y(1,2)-y(1,2);
```

Implicit Euler:

Program Commands:

```
function [T,Y] = vImpEuler(t0,tf,y0,h)
% [T,Y] contains information about each variable at each time step
% The ceil function rounds to the highest integer.
n=length(y0); nsteps=ceil((tf-t0)/h);
P=zeros(nsteps+1,n+1);
t=t0;y=y0;
T(1,1)=t0; Y(1,:)=y0';
for i=1:nsteps
% Make sure we hit tf on last step.
if i==nsteps
h=tf-t;
end
% Iterate Implicit Euler equation using Newton's method
itmax=100; tol=1e-10; yi=y;
for k=1:itmax
r=-(y-yi-h*f(y));
J=eye(n)-h*ssjac(y);
delta=J\r;
y=y+delta;
% Can use this to check for QC:
%disp(num2str(norm(delta)))
if norm(delta)<=tol
break
end
end
if k==itmax
disp(['Error. Newton method did not convegere at time step ',num2str(i)])
return
end
t=t+h;
T(i+1,1)=t; Y(i+1,:)=y;
end
%
```

```
function rhs=f(y)
n=length(y);
rhs(1,1)=y(1,1)*(1-y(1,1))-y(1,1)*y(2,1);
rhs(2,1)=2*y(1,1)*y(2,1)-y(2,1);
%
function J=ssjac(y)
% This function calculates tßhe Jacobian of the rhs function, f(y),
% namely df_i/dy_j
n=length(y);
J=zeros(n,n);
J(1,1)=1-2*y(1,1)-y(2,1);
J(1,2)=-y(1,1);
J(2,1)=2*y(2,1);
J(2,2)=2*y(1,1)-1;
```

RK4:

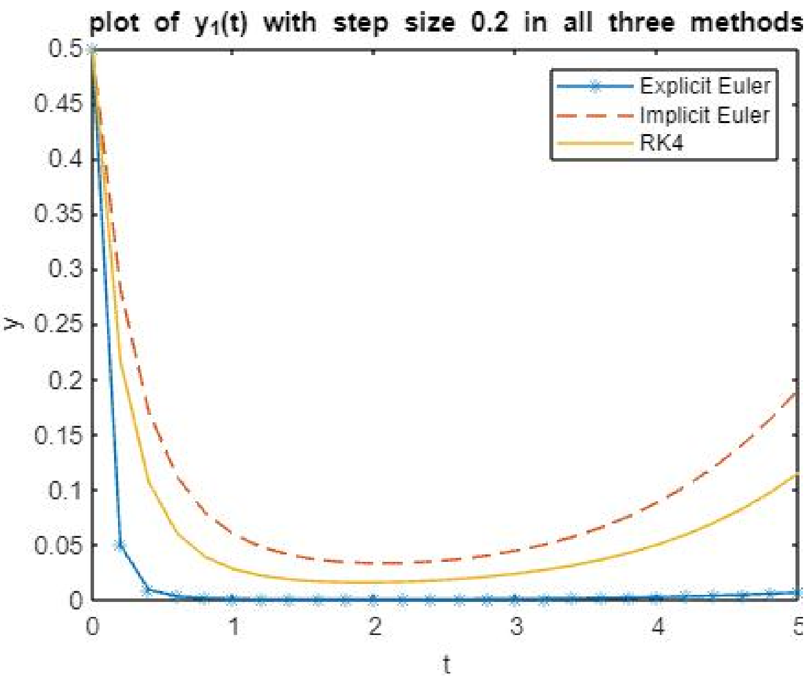
Program Commands:

```
function [T,Y] = vRK4(t0,tf,y0,h)
% [T,Y] contains information about each variable at each time step
% The ceil function rounds to the highest integer.
n=length(y0); nsteps=ceil((tf-t0)/h);
P=zeros(nsteps+1,n+1);
t=t0;y=y0;
T(1,1)=t0; Y(1,:)=y0';
for i=1:nsteps
% Make sure we hit tf on last step.
if i==nsteps
h=tf-t;
end
% Compute recursion functions, Ki.
K1=h*f(y);
K2=h*f(y+K1/2);
K3=h*f(y+K2/2);
K4=h*f(y+K3);
% Compute y and t at this step.
y=y+(K1+2*K2+2*K3+K4)/6;
t=t+h;
T(i+1,1)=t; Y(i+1,:)=y';
end
%
function rhs=f(y)
rhs(1,1)=y(1,1)*(1-y(1,1))-y(1,1)*y(1,2);
rhs(1,2)=2*y(1,1)*y(1,2)-y(1,2);
```

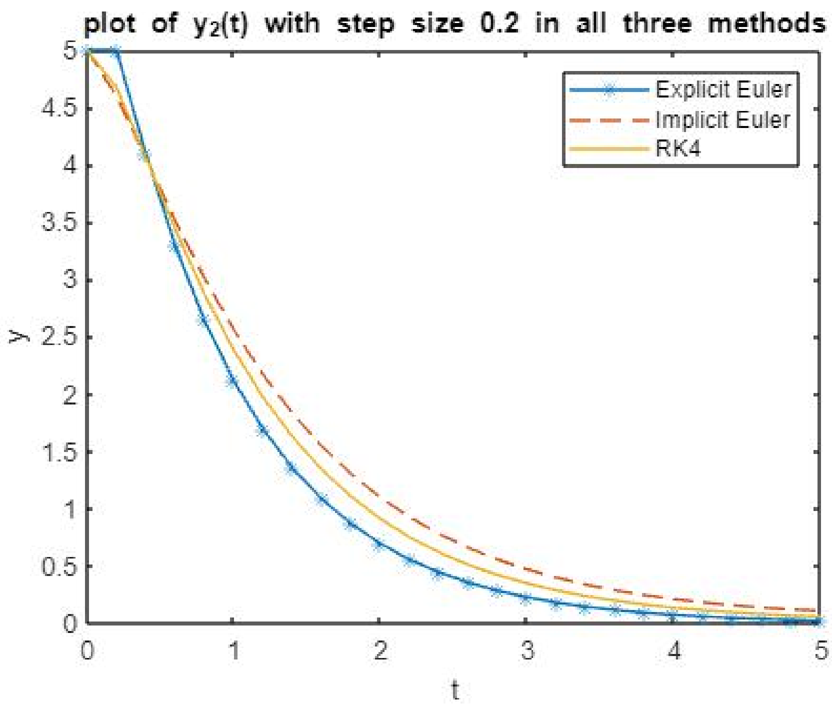
$y_1(t)$, y2(t) plots with step size 0.2 in all three methods:

Program Commands:

```
[T,Y]=vEuler(0,5,[0.5,5],0.2);
plot (T,Y(:,1),'-*');
hold on
[T,Y]=vImpEuler(0,5,[0.5;5],0.2);
plot (T,Y(:,1),'--');
[T,Y]=vRK4(0,5,[0.5,5],0.2);
plot (T,Y(:,1));
legend ('Explicit Euler','Implicit E
xlabel('t');
ylabel('y');
title('plot of y_1(t) with step size
figure(2)
[T,Y]=vEuler(0,5,[0.5,5],0.2);
plot (T,Y(:,2),'-*');
hold on
[T,Y]=vImpEuler(0,5,[0.5;5],0.2);
plot (T,Y(:,2),'--');
[T,Y]=vRK4(0,5,[0.5,5],0.2);
```



```
plot (T,Y(:,2));
legend ('Explicit Euler','Implicit Euler','RK4');
xlabel('t');
ylabel('y');
title('plot of y_2(t) with step size 0.2 in all three methods');
```



Note: For remaing step sizes**(0.6,1.08)** it is not possible to plot $y_1(t)$ (and $y_2(t)$) in one figure by using all three methos**(Explicit Euler, Implicit Euler, RK4)** beacuze **y values varying largely** like 10^15

Therefore I draw **individual figures with induvidual method** for 0.6 and 1.08 step sizes, see the below mentioned.

Program Commands: for remaing invidual plots

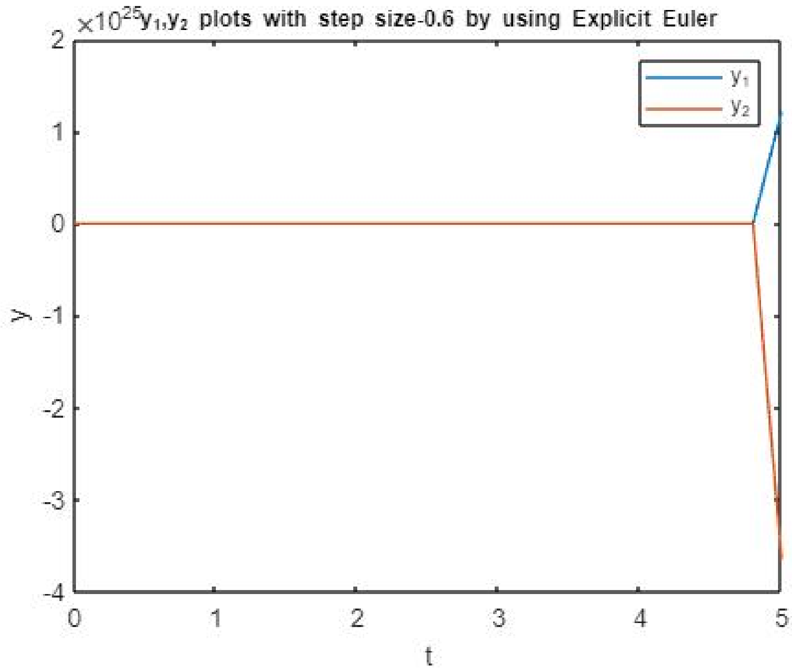
```
[T,Y]=vEuler(0,5,[0.5,5],0.6);
plot(T,Y);
xlabel('t')
ylabel('y')
title('y_1,y_2 plots with step size-0.6 by using Explicit Euler');
legend('y_1','y_2');
figure(2);
[T,Y]=vEuler(0,5,[0.5,5],1.08);
plot(T,Y,'.-');
xlabel('t')
ylabel('y')
title('y_1,y_2 plots with step size-1.08 by using Explicit Euler');
legend('y_1','y_2');
figure(3);
[T,Y]=vImpEuler(0,5,[0.5;5],0.6);
plot(T,Y,'-*')
xlabel('t')
ylabel('y')
title('y_1,y_2 plots with step size-0.6 by using Implicit Euler');
legend('y_1','y_2');
figure(4);
[T,Y]=vImpEuler(0,5,[0.5;5],1.08);
plot(T,Y,':gs')
xlabel('t')
ylabel('y')
title('y_1,y_2 plots with step size-1.08 by using Implicit Euler');
legend('y_1','y_2');
figure(5);
[T,Y]=vRK4(0,5,[0.5,5],0.6);
plot(T,Y,'--');
```

```
xlabel('t')
ylabel('y')
title('y_1,y_2 plots with step size-0.6 by using RK4');
legend('y_1','y_2');
figure(6);
[T,Y]=vRK4(0,5,[0.5,5],1.08);
plot(T,Y,'--');
xlabel('t')
ylabel('y')
title('y_1,y_2 plots with step size-1.8 by using RK4');
legend('y_1','y_2');
```

2)Explicit Euler: @step size 0.6

Command Window Execution Statement:

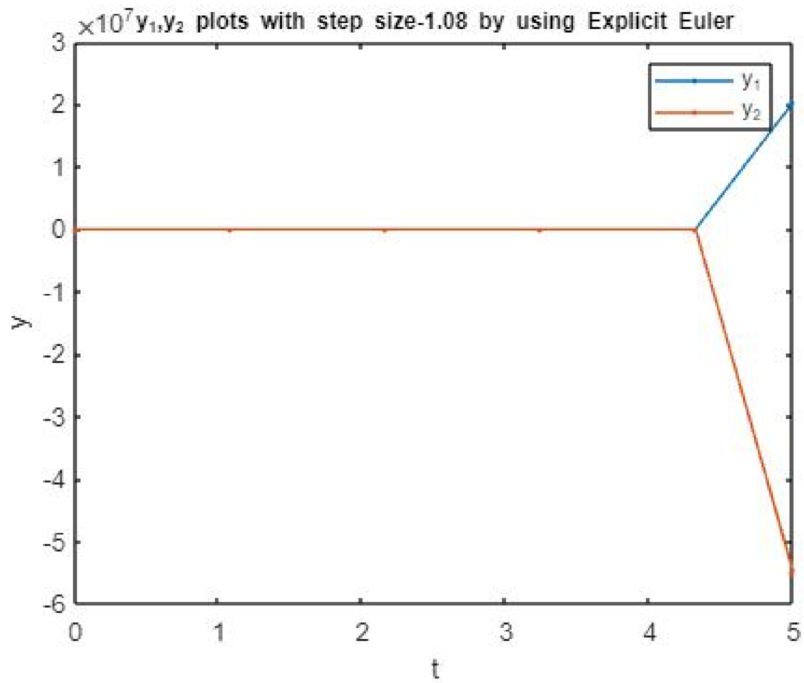
```
[T,Y]=vEuler(0,5,[0.5,5],0.6)
```



2) Explicit Euler: @step size 1.08

Command Window Execution Statement:

```
[T,Y]=vEuler(0,5,[0.5,5],1.08)
```



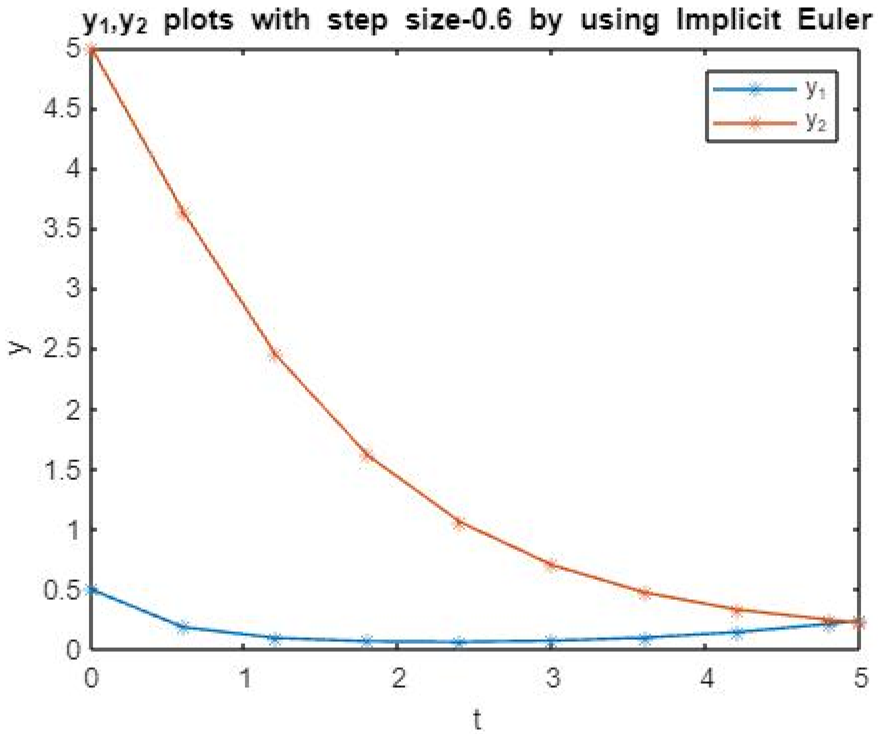
Note:

1) With step size h = 0.6,1.08 causes instability (solution is unbounded)

2)Implicit Euler: @step size 0.6

Command Window Execution Statement:

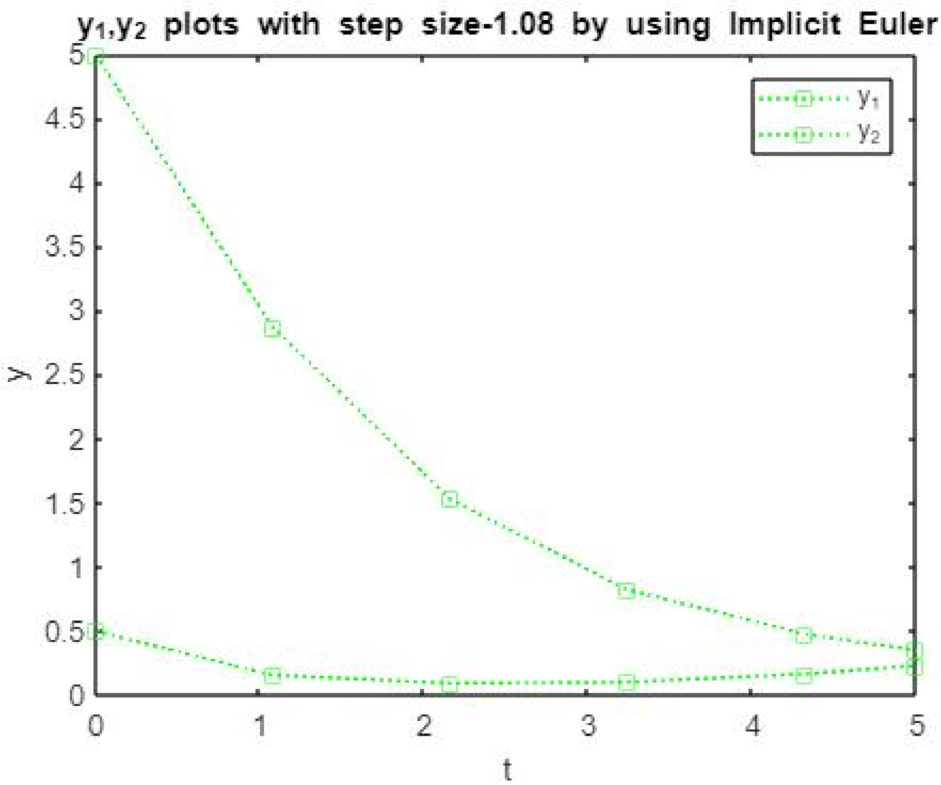
```
[T,Y]=vImpEuler(0,5,[0.5;5],0.6)
```



2)Implicit Euler: @step size 1.08

Command Window Execution Statement:

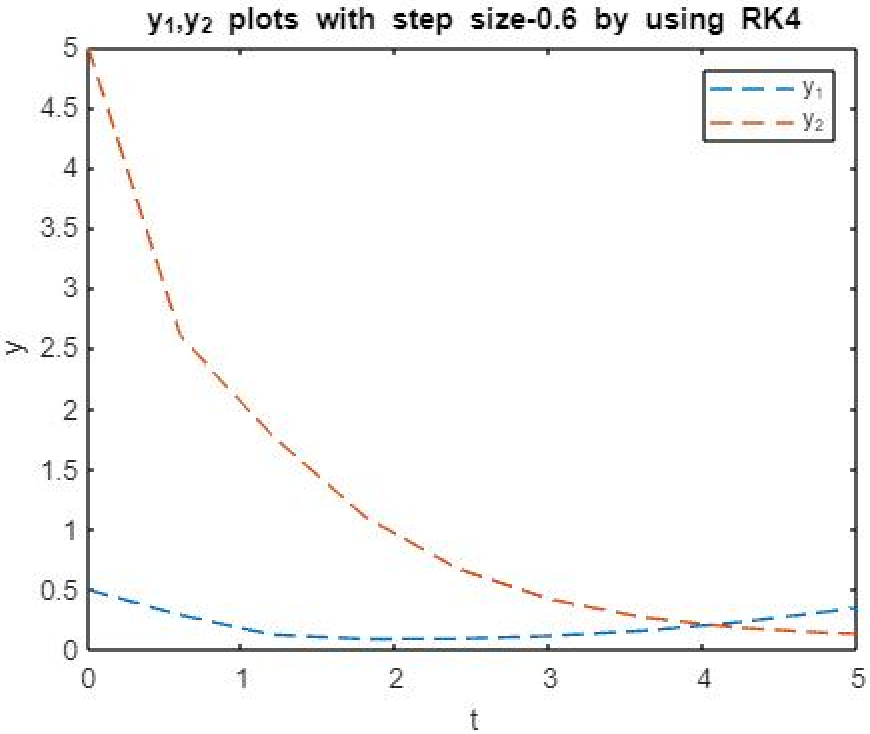
```
[T,Y]=vImpEuler(0,5,[0.5;5],1.08)
```



2)RK4: @step size 0.6

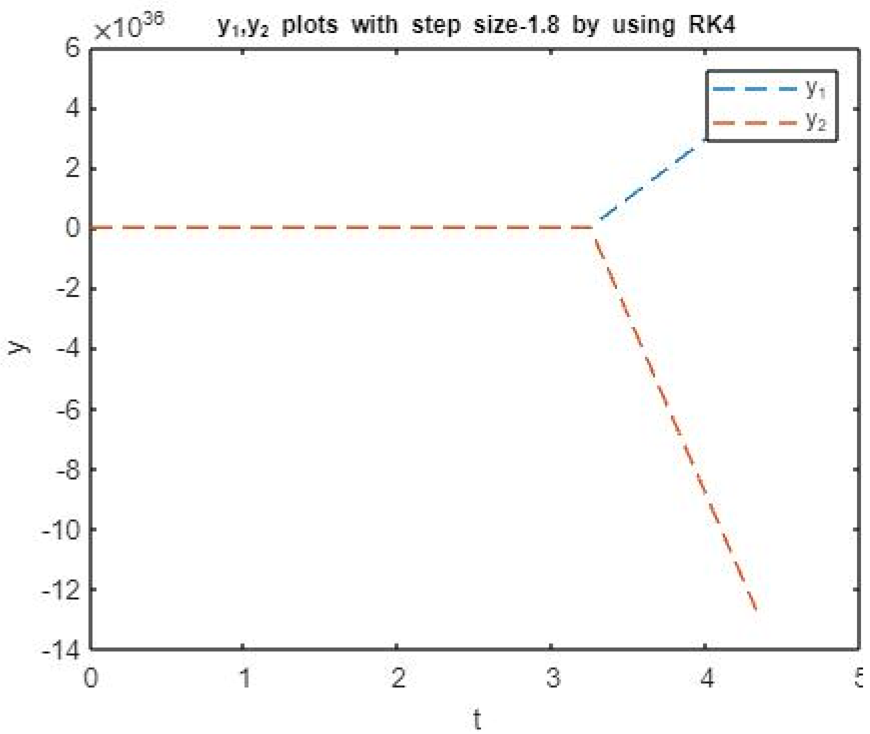
Command Window Execution Statement:

```
[T,Y]=vRK4(0,5,[0.5,5],0.6)
```



Command Window Execution Statement:

```
[T,Y]=vRK4(0,5,[0.5,5],1.08)
```



1) With step size h =1.08 causes instability (solution is unbounded)

Therefore step size 0.2 suitable for Explicit Euler and RK4 Method.

1) $\frac{dy_1}{dt} = y_1(1-y_1) - y_1 y_2$ | $y_1(0) = 0.5$

$\frac{dy_2}{dt} = 2y_1 y_2 - y_2$ | $y_2(0) = 5$

Explicit Euler method

$y_{i+1,1} = y_i + h f(t_i, y_{i,1})$ — Eq (1)

$y_{i+1,2} = y_i + h f(t_i, y_{i,2})$ — Eq (2)

$f(0, 0.5)$ (at $t=0$)
$= 0.5(1-0.5) - 0.5(5)$
$= 0.25 - 2.5 = -2.25$

i=0 :- Eq (1)

$y_{1,1}(t) = 0.5 + h(-2.25)$

Eq (2)

$y_{1,2}(t) = 5 + h f(t_i, y_{i,2})$

$f(0,5) = 2(0.5)(5) - 5 = 0$ (at $t=0$)

$y_{1,2}(t) = 5 + h(0)$

$\therefore y_{1,1}(t) = 0.5 - h\,2.25 \Rightarrow h \le \frac{5}{2.25}$

$h \le 2.22$

$y_{1,2}(t) = 5 - h(0)$

$\Rightarrow \frac{5}{0} \ge h$

$\Rightarrow \infty \ge h$

$\therefore$ at $t=0$ $h \le 2.22$ (or) $h \le \infty$

therefor one is changing rapidly, one is changing slowly.

Implicit Euler method :-

$y_{i+1,1} = y_{i,1} + h f(t_{i+1}, y_{i+1})$

$f(t_{i+1}, y_{i+1}) = y_{i+1,1}(1 - y_{i+1,1}) - (y_{i+1,1} \times y_{i+1,2})$

$\therefore y_{i+1,1} = y_{i,1} + h\{y_{i+1,1}(1 - y_{i+1,1}) - [y_{i+1,1} \times y_{i+1,2}]$

$y_{2,i+1} = y_{2,i} + h(2y_{i+1,1} \times y_{i+1,2} - y_{i+1,2})$

rearranging terms

$$\therefore y_{i,1} = y_{i+1,1}(1 - h + y_{i+1,2} h) + h y_{i+1,1} - y_{i+1,2}$$

$$y_{i,2} = y_{2,i+1} - h 2 y_{i+1,1} y_{i+1,2} - h y_{i+1,2}$$

$$y_{i,1} = y_{i+1,1}(1 - h + y_{i+1,1} \times h + x) - y_{i+1,2}$$

$$y_{i+1,1} = \frac{y_{i,1} + y_{i+1,2}}{1 + h y_{i+1,1}}$$

For this case regardless of the step size, $|y_i| \to 0$ as $i \to \infty$, Hence approach unconditionally stable. $\therefore$ any step size accepted

RK4 method

$$y_{i+1} = y_i + h\left(\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4\right)$$

$$k_{1,1} = f_1(0, 0.5, 5) \cancel{= 2.25} \ (-2.25)$$

$$k_{1,2} = f_2(0, 0.5, 5) = 0$$

$$y_1 + k_{1,1}\frac{h}{2} = 0.5 + \left(-2.25 \times \frac{h}{2}\right) = 0.5 - 1.125h$$

$$y_2 + k_{1,2}\frac{h}{2} = 5 + 0 = 5$$

$$k_{2,1} = f_1(t, 0.5 - 1.125h, 5)$$

$$= (0.5 - 1.125h)(1 - (0.5 - 1.125h)) - (0.5 - 1.125h)(5)$$

$$k_{2,2} = f_1(t, 0.5 - 1.125h, 5) = 2(0.5 - 1.125h)$$

$$y_1 + k_{2,1}\frac{h}{2} = 0.5 + \left[(0.5 - 1.125h)(1 - (0.5 - 1.125h)) - (0.5 - 1.125h)(1) \times \frac{h}{2}\right]$$

$$y_2 + k_{2,2}\frac{h}{2} = 5 + 2(0.5 - 1.125h)(h/2)$$

From the Calculations got the step size is **0.22**

Name: K. Sowjanya
Roll No: MM22M023