

EM314 – NUMERICAL METHODS
ASSIGNMENT - ODE

DE SILVA K.G.P.M.

E/15/065

SEMESTER 04

08/01/2019

(1)

a) Forward Euler method,

```
function a = Feuler(f,h,x0,xend,init)
    index=1; % set index value to 1
    a(index)=init; %find y when x=x0 (initial value)
    for i=x0:h:xend-h
        a(index+1) = a(index) + h*f(i,a(index)); %iterative part
        index=index+1;
    end
end
```

b) Improved Euler method,

```
function a =Imeuler(f,h,x0,xend,init)
    index=1; % set index value to 1
    a(index)=init; %find y when x=x0(actualy initial value)
    for i=x0:h:xend-h
        val = a(index) + h*f(i,a(index)); %find yn+1
        tempY=f(i+h,val); %find y'(n+1)
        avgY=(f(i,a(index))+tempY)/2.0; % find the average of y'(n)
        and y'(n+1)
        a(index+1)=a(index)+avgY*h; %find the next value for y
        according method
        index=index+1;
    end
end
```

C) 4th order Runge – Kutta method,

```
function a = RungeK(f,h,x0,xend,init)
    index=1; %set intital value to 1
    a(index)=init; % initial y value

    for i=x0:h:xend-h
        k0=h*f(i,a(index)); % find k0
        k1=h*f(i+h/2,a(index)+0.5*k0); % find k1
        k2=h*f(i+h/2,a(index)+0.5*k1); % find k2
        k3=h*f(i+h,a(index)+k2); % find k3
        a(index+1)=a(index)+(1/6)*(k0+2*k1+2*k2+k3); % find next
        y value
        index=index+1;
    end
end
```

(2)

```
A=1;
fi=10;
beta=8;
x0=0;
xend=1.2;
initial=0;
func= @(x,y) (fi-beta*sqrt(y))/A; %differential function

h=0.2;
y1=Feuler(func,h,x0,xend,initial); % call forward euler method
h=0.1;
y2=Feuler(func,h,x0,xend,initial); % call forward euler method
h=0.2;
y3=Imeuler(func,h,x0,xend,initial); % call Improved euler method
y4= RungeK(func,h,x0,xend,initial); % call Runge Kutta method

%display y values according to each method
disp(y1);
disp(y2);
disp(y3);
disp(y4);

x=x0:0.2:xend;
x1=x0:0.1:xend;
% plot variation of h(t) according to methods
hold on
plot(x,y1); %forward euler when h=0.2
plot(x1,y2,'r'); %forward euler when h=0.1
plot(x,y3,'g'); %Improved euler when h=0.2
plot(x,y4,'c'); %Runge Kutta when h=0.2
xlabel('t');ylabel('h(t)');title('Variation in h(t)');
legend('forward euler h=0.2','forward euler h=0.1','improved
euler','runge kutta');
```

Forward Euler method ,

h=0.2

| x | y _n |
|-----|----------------|
| 0 | 0 |
| 0.2 | 2.0000 |
| 0.4 | 1.7373 |
| 0.6 | 1.6284 |
| 0.8 | 1.5867 |
| 1.0 | 1.5713 |
| 1.2 | 1.5657 |

h=0.1

| x | y _n |
|-----|----------------|
| 0 | 0 |
| 0.1 | 1.0000 |
| 0.2 | 1.2000 |
| 0.3 | 1.3236 |
| 0.4 | 1.4032 |
| 0.5 | 1.4556 |
| 0.6 | 1.4904 |
| 0.7 | 1.5137 |
| 0.8 | 1.5295 |
| 0.9 | 1.5401 |
| 1.0 | 1.5473 |
| 1.1 | 1.5522 |
| 1.2 | 1.5555 |

Improved Euler method,

$h=0.2$

| x | y_n |
|-----|--------|
| 0 | 0 |
| 0.2 | 0.8686 |
| 0.4 | 1.1841 |
| 0.6 | 1.3526 |
| 0.8 | 1.4451 |
| 1.0 | 1.4965 |
| 1.2 | 1.5253 |

Runge Kutta method,

$h=0.2$

| x | y_n |
|-----|--------|
| 0 | 0 |
| 0.2 | 0.9259 |
| 0.4 | 1.2443 |
| 0.6 | 1.3987 |
| 0.8 | 1.4771 |
| 1.0 | 1.5177 |
| 1.2 | 1.5389 |

