AS3

In this assignment, we are allowed to solve a second order IVP question as follows

$$\ddot{Y} + 4Y = \cos 2t\,, \quad Y(0) = 1\,, \dot{Y}(0) = 0$$

Its exact solution is $Y = \cos 2t + 1/4\sin 2t$.

1.Principal

Firstly, we need to convert 2nd-order derivative question into 1st order derivative system by integrating a new variable.

$$Y = y_1\,, \quad \dot{Y} = y_2$$

$$\ddot{Y} = \dot{y}_2$$

$$\vec{Y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

$$\dot{\vec{Y}} = \begin{pmatrix} y_2 \\ \cos 2t - 4y_1 \end{pmatrix}$$

Basically, using Euler's explicit method, we can get a solution.

$$Y_{n+1} = \begin{pmatrix} y_{1n} \\ y_{2n} \end{pmatrix} + h\begin{pmatrix} y_{2n} \\ \cos 2t - 4y_{1n} \end{pmatrix}$$

After iteration, we approximately approach the exact value.

2.Subroutines

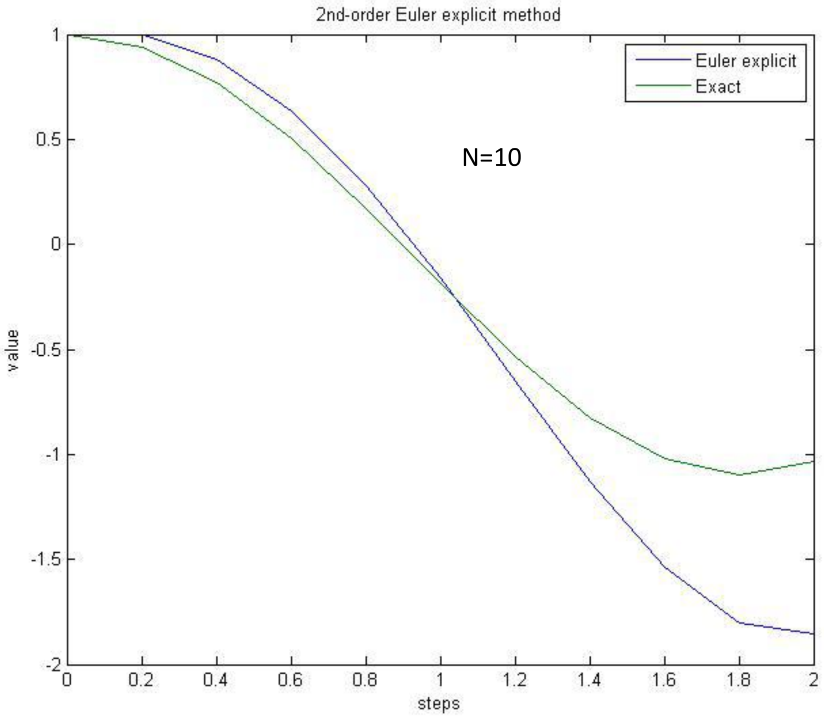
```
function out=secondorder(f,a,b,n)
y1=zeros(1,n+1);
y2=zeros(1,n+1);
y1(1)=1;
y2(1)=0;
Y=[y1',y2']';
h=(b-a)/n;
T=a:h:b;
Q=zeros(1,n+1);
Q(1)=1;
err=zeros(1,n+1);
for i=1:n
    for j=1:2
    Yt=[y2(i);feval(f,T(i),y1(i))];
    Y(j,i+1)=Y(j,i)+h*Yt(j);
    end
    y2(i+1)=Y(2,i+1);
```

```
    y1(i+1)=Y(1,i+1);
    Q(i+1)=cos(2*T(i+1))+1/4*T(i+1)*sin(2*T(i+1));
    err(i+1)=100*abs((Q(i+1)-Y(1,i+1))/Y(1,i+1));
end
out=[T;Y;Q;err]';
plot(T,y1,T,Q);
legend('Euler explicit','Exact');
xlabel('steps')
ylabel('value')
title('2nd-order Euler explicit method')
end
```

3.Outcome and accuracy

We call this function by inputting step h=0.2, and analyse it in the interval [0,2].

step	y_1	y_2	Exact value	Relative error(%)
0	1	0	1	0
0.2	1	-0.6	0.9405	5.9468
0.4	0.88	-1.2158	0.7684	12.6770
0.6	0.6368	-1.7804	0.5022	21.1479
0.8	0.2808	-2.2174	0.1707	39.1938
1.0	-0.1627	-2.4479	-0.1888	16.0295
1.2	-0.6523	-2.4009	-0.5308	18.0220
1.4	-1.1325	-2.0266	-0.8250	27.1545
1.6	-1.5378	-1.3090	-1.0216	33.5651
1.8	-1.7996	-0.2784	-1.0959	39.1040
2.0	-1.8553	0.9819	-1.0320	44.3730



4.Stability analysis

We can transform initial equation to a matrix through algebraic relation.

$$\dot{Y} = \begin{pmatrix} 0 & 1 \\ -4 & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} + \begin{pmatrix} 0 \\ \cos 2t \end{pmatrix}$$

$$Y_{n+1} = Y_n + h \begin{pmatrix} 0 & 1 \\ -4 & 0 \end{pmatrix} Y_n$$

Eventually, we get a normal formula $Y_{n+1} = AY_n$.

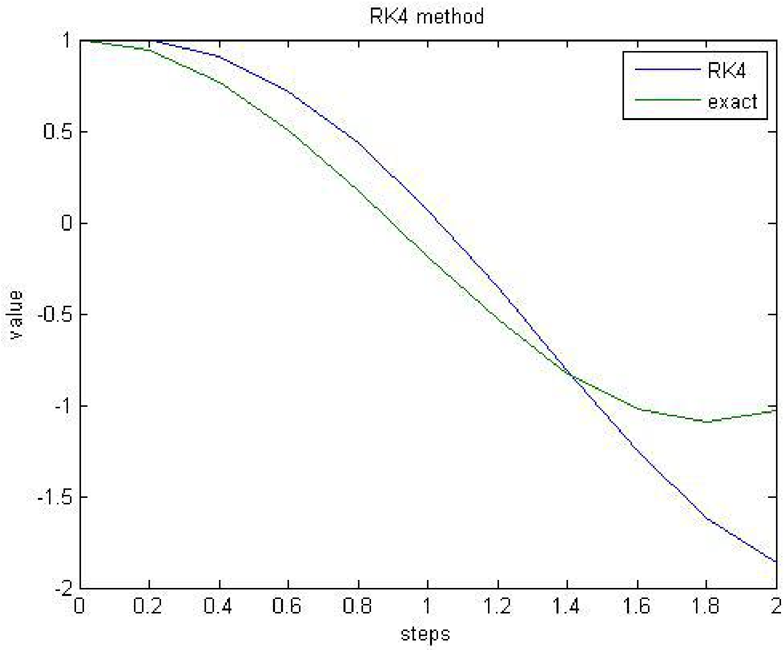
$$A = \begin{pmatrix} 1 & 0.2 \\ -0.8 & 1 \end{pmatrix}$$

And then solve out its eigenvalue by using eig() function in matlab. $\lambda_{1,2} = 1 \pm 0.4i$ .As we know, the solution of 1st order IVP question is $y(t) = c_1 e^{\lambda_1 t} \overset{\rightarrow(1)}{\eta} + c_2 e^{\lambda_2 t} \overset{\rightarrow(2)}{\eta}$ (η is the eigenvector of matrix A and can be calculated by eigenvalue). Once either of λ_1 and λ_2 is positive, our result will blow up and it could be unbounded. While in this question, the real part of the eigenvalues are both positive. We can draw a conclusion that this function is instable.

By the way, I have tried to solve this question by RK4 method, but it seems not as perfect as I thought. Here are subroutines.

```
function outcome=RK4sec(f,a,b,n)
h=(b-a)/n;
T=a:h:b;
y1=zeros(1,n+1);
y2=zeros(1,n+1);
y1(1)=1;
y2(1)=0;
Y=zeros(2,n+1);
Y=[y1;y2];
Q=zeros(1,n+1);
Q(1)=1;
err=zeros(1,n+1);
dy=zeros(1,n+1);
% since dy/dt=y2  dy2/dt=feval(f,t,y1)
    %we're going to use RK4 twice for diffefent deriative.
for i=1:n
    %k1-4 is for feval(f,t,y1)
    k1=h*feval(f,T(i),y1(i));
    k2=h*feval(f,T(i)+h/2,k1/2+y1(i));
    k3=h*feval(f,T(i)+h/2,k2/2+y1(i));
    k4=h*feval(f,T(i)+h,k3+y1(i));
    K=(k1+2*k2+2*k3+k4)/6;
    %l1-4 is for dy/dt
    l1=h*y2(i);
    l2=h*(l1/2+y2(i));
    l3=h*(l2/2+y2(i));
    l4=h*(l3+y2(i));
    L=(l1+2*l2+2*l3+l4)/6;
    %-----------------------
    P(1,i)=L;
    P(2,i)=K;
    for j=1:2
    Y(j,i+1)=Y(j,i)+P(j,i);
    y1(i+1)=Y(1,i+1);
    y2(i+1)=Y(2,i+1);
    end
    Q(i+1)=cos(2*T(i+1))+1/4*T(i+1)*sin(2*T(i+1));
    err(i+1)=(abs(Q(i+1)-Y(1,i+1)))/Y(1,i+1)*100;
    dy(i+1)=-2*sin(2*T(i+1))+1/4*sin(2*T(i+1))+1/2*T(i+1)*cos(2*T(i+1));
end
outcome=[ T' Y' Q' err'  dy'];
plot(T,y1,T,Q);
legend('RK4','exact');
```

```
end
```
This is the figure I drew



The gap between exact value and approximation seems bigger than Euler's explicit method.