**MATH 2271 - Ordinary Differential Equations**

**Semester II - 2017/2018**

**Lab #3**

# 1 Solving a general system of coupled 1st order ODEs using Euler's Method

## 1.1 Problem

1. Write a MATLAB function to find and make overlaid plots of the numerical solution of the system of first order differential equations:

$$\frac{dY}{dx} = F(x, Y), \qquad Y(a) = Y0,$$

over the interval $[a, b]$, using Euler's method, with step size $h$.

[Here, $Y$ is a column vector in which each row contains a different unknown variable to be solved for. If there are $n$ equations in the system then $Y$, $Y0$ and $F$ are all of size $n \times 1$.]

2. Consider the IVP

$$\frac{du}{dx} = v - x, \qquad u(0) = 2,$$

$$\frac{dv}{dx} = u, \qquad v(0) = 1,$$

$$\frac{dw}{dx} = u - v + x, \qquad w(0) = 1,$$

where $x \in [0, 2]$. Use Euler's method to obtain overlaid plots of the numerical solutions for $u$, $v$ and $w$ over the interval $[0, 2]$, using step size $h = 0.001$.

[Note: The exact solutions are: $u = e^x + 1$, $v = e^x + x$, $w = x + 1$.]

## 1.2 MATLAB Implementation

1. **Code for Euler's Method:**

```
function [x,Y] = Eulersys(a, b, h, Y0, F)
% MATLAB code for using Euler's Method to solve a general system of
%   1st order ODEs dY/dx=F(x,Y) on the interval [a,b]
%       with  initial conditions Y(a)=Y0

% Inputs:   a and b define the interval (a,b)
%           h is the step size
%           Y0 is a column vector that defines the initial conditions
%           F is a function that gives the LHS of the system
%               dY/dx=F(x,Y), and is defined separately

% Outputs:  x stores the discretised values of x in the interval [a,b]
%           Y is a vector that stores the corresponding approximate values


x=a:h:b;  % discretizes the interval for x
N=length(x);    % determines the number of points
M=length(Y0);   % determines the number of dependent variables
Y=zeros(M,N);   % initializes Y as the required matrix of zeros
Y(:,1)=Y0;    % stores the value of Y(a) as the first column of Y
for j=1:N-1   % using a for loop to compute the remaining Y values
    for i=1:M   % Applying the Euler algorithm for each unknown variable
        Y(i,j+1)=Y(i,j)+h*feval(F,i,x(j),Y(:,j));
    end
end
for i=1:M   % plotting the solution for each unknown variable
    plot(x,Y(i,:))
    hold on
end
hold off
end
```

```
function [x,Y] = Eulersys(a, b, h, Y0, F)
x=a:h:b;
N=length(x);
M=length(Y0);
Y=zeros(M,N);
Y(:,1)=Y0;
for i=1:N-1
for i=1:M
Y(i,j+1)=Y(i,j)+h*feval(F,i,x(j),Y(:,j));
end
end
for i=1:M
plot(x,Y(i,:))
hold on
end
hold off
end
```

2. **Code for the function F:**

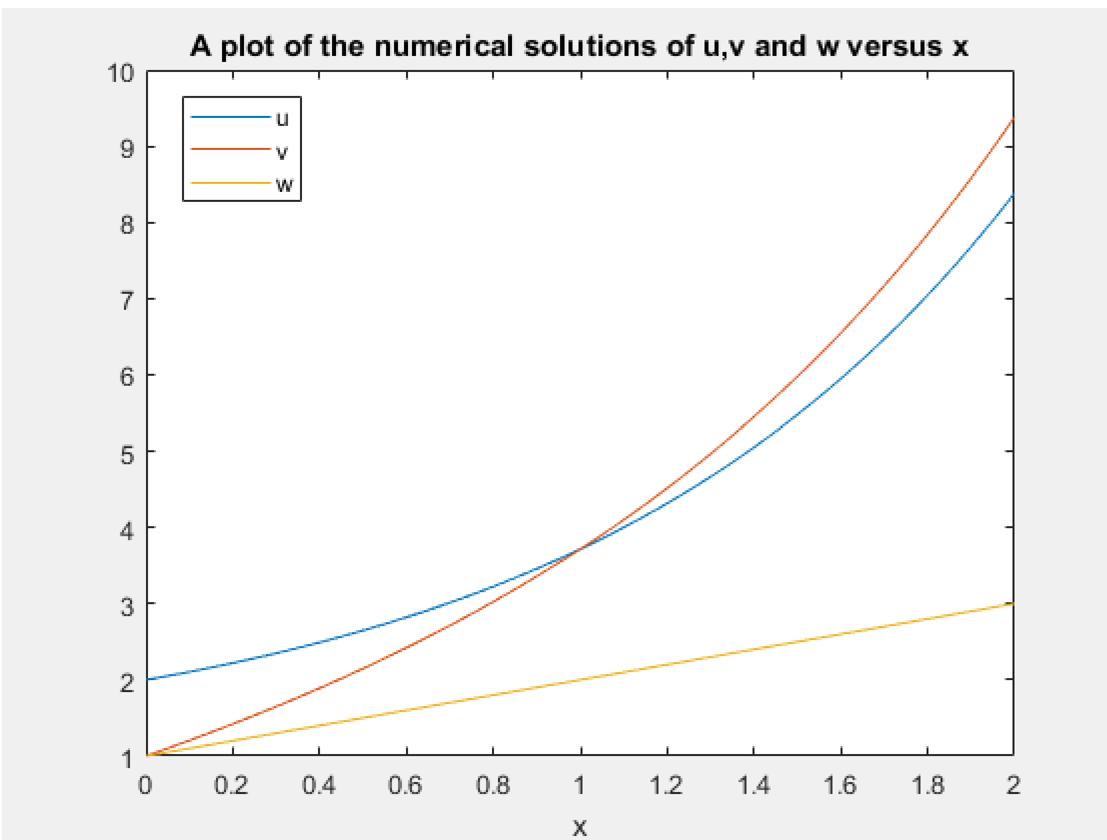
Note that the functions $f$ , $g$ and $h$ are defined here as $f(x,u,v,w) = v - x$, $g(x,u,v,w) = u$ and $h(x,u,v,w) = u - v + x$.

```
function [ t ] = F( i,x,Y )
u=Y(1); v=Y(2); w=Y(3);
if i==1
t=v-x;
elseif i==2
t=u;
else
t=u-v+x;
end
end
```

**Command Window:**

```
>> [x,Y] = Eulersys(0, 2, 0.001, [2;1;1], 'F');
>>xlabel('x')
>> legend('u','v','w')
```

**Plot:**



## 2 Solving 1st order ODEs using MATLABS built-in solvers

Consider again the general first order IVP $\frac{dy}{dx} = f(x, y)$, where $x \in [a, b]$, subject to the initial condition $y(a) = y0$.

MATLAB has several built in solvers for problems of this type, the most popular being **ode23** and **ode45**.

The syntax for using these solvers is as follows:

$>>$ **[x,y]=ode23('fun_name', [a b], y0)**

$>>$ **[x,y]=ode45('fun_name', [a b], y0)**

where **fun_name** is the function that defines $f(x, y)$.

Note that **ode23** is a faster solver, but less accurate than **ode45.**

**Example:** Consider the IVP

$$\frac{dy}{dx} = x - y^2$$

where $x \in [0, 2]$, subject to the initial condition $y(0) = 1$.

**Solution:**

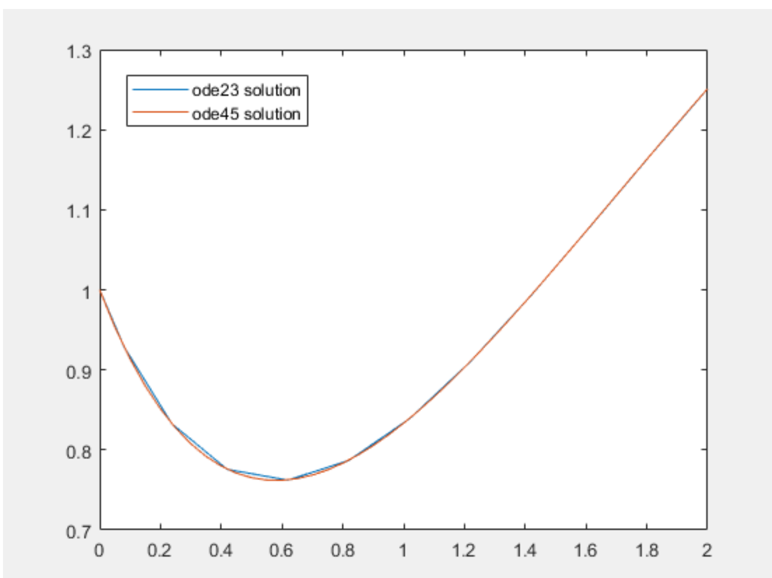
**Function f:**

```
function t = f(x,y)
t=x-y.^2;
end
```

**Command Window:**

```
>> [x1,y1]=ode23('f',[0 2],1);
>> [x2,y2]=ode45('f',[0 2],1);
>> plot(x1,y1,x2,y2)
>> legend('ode23 solution', 'ode45 solution')
```

**Plot:**



## 3 PPLANE

The software is available at: http://math.rice.edu/∼dfield/dfpp.html

Click on the link to download the file **pplane.jar** at the bottom of the page.

Using pplane, obtain phase portraits for the following systems:

**CASE 1: Distinct, Real Eigenvalues**

1.

$$\mathbf{x}' = \begin{pmatrix} -1 & 4 \\ -2 & 5 \end{pmatrix} \mathbf{x}$$

[The eigenvalues $\lambda = 1, 3$ are both real and positive, so the solution is said to be **unstable**.]

2.

$$\mathbf{x}' = \begin{pmatrix} -3 & 0 \\ 3 & -2 \end{pmatrix} \mathbf{x}$$

[The eigenvalues $\lambda = -3, -2$ are both real and negative, so the solution is said to be **asymptotically stable**.]

3.

$$\mathbf{x}' = \begin{pmatrix} 4 & 0 \\ 2 & -1 \end{pmatrix} \mathbf{x}$$

[The eigenvalues $\lambda = 4, -1$ are both rea, but have different signs, so the solution is said to have a **saddle point**. Solutions with saddle points are always unstable.]

**CASE 2: Repeated, Real Eigenvalues**

4.

$$\mathbf{x}' = \begin{pmatrix} 2 & -3 \\ \frac{1}{3} & 4 \end{pmatrix} \mathbf{x}$$

[The eigenvalues $\lambda = 3, 3$ (repeated) are both real and positive, so the solution is **unstable**.]

5.

$$\mathbf{x}' = \begin{pmatrix} -7 & 1 \\ -4 & -3 \end{pmatrix} \mathbf{x}$$

[The eigenvalues $\lambda = -5, -5$ (repeated) are both real and negative, so the solution is **asymptotically stable**.]

**CASE 3: Complex Eigenvalues**

6.

$$\mathbf{x}' = \begin{pmatrix} -2 & 3 \\ -3 & -2 \end{pmatrix} \mathbf{x}$$

[The eigenvalues $\lambda = -2 \pm 3i$ are both complex. The real part of the eigenvalues is negative, so the solution is said to have a **spiral point**, and the solution is **asymptotically stable**.]

7.

$$\mathbf{x}' = \begin{pmatrix} 2 & 3 \\ -3 & 2 \end{pmatrix} \mathbf{x}$$

[The eigenvalues $\lambda = 2 \pm 3i$ are both complex. The real part of the eigenvalues is positive, so the solution is said to have a **spiral point**, and the solution is **unstable**.]

8.

$$\mathbf{x}' = \begin{pmatrix} 0 & 1 \\ -5 & 0 \end{pmatrix} \mathbf{x}$$

[The eigenvalues $\lambda = \pm 5i$ are both complex. The real part of the eigenvalues is zero, so the solution is said to have a **center**, and the solution is said to be **neutrally stable**.]