Advanced Programming in Engineering
Using the matlab ode45 routine

Question: numerically solve $y(x)$ from

$$\frac{dy}{dx} = f(y) = 1 - \tfrac{1}{2}y$$

over the interval $0 \leq x \leq 10$, given that $y(0) = 0$.

Answer: create a function m-file that, for a given x and y, returns the derivative of y.

```
function dydx = MyODE(x,y)   % invocation must accept two arguments
dydx = 1 - 0.5 * y;          % calculate and return the derivative
```

Initialize variables and call `ode45`,

```
range = [0 10];                     % range of independent coordinate
y0 = 0;                             % start value
[T,Yt] = ode45(@MyODE,range,y0);    % obligatory order of arguments
plot(T,Yt)                          % plot the result
```

Note that the function `MyODE` is passed to `ode45` as an argument.

Question: numerically solve the coupled differential equations

$$\begin{cases} dr(t)/dt = -r(t) - 0.1s(t) \\ ds(t)/dt = -0.2r(t) - s(t) \end{cases}$$

with the starting point

$$\begin{cases} r(0) = 1 \\ s(0) = 2 \end{cases}$$

over the interval $0 \leq t \leq 10$.

Answer: combine the two coupled equations into one vector equation

$$\frac{d\mathbf{z}}{dt} = \mathbf{A}\mathbf{z} \text{ with } \mathbf{z} = \begin{pmatrix} r \\ s \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} -1 & -0.1 \\ -0.2 & -1 \end{pmatrix} \text{ and } \mathbf{z}(0) = \begin{pmatrix} 1 \\ 2 \end{pmatrix}.$$

Write a function m-file that, for given $\mathbf{z}$ and t, returns the derivative of $\mathbf{y}$,

```
function dzdt = MyODE(t,z)  % invocation must accept two arguments
global A                    % share matrix A
dzdt = A * z;               % calculate and return the derivative
```

Initialize variables and call `ode45`,

```
global A                          % share matrix A
A = [ -1 -0.1 ; -0.2 -1]          % fill matrix A
range = [0 10];                   % range of independent coordinate
z0 = [1 ; 2];                     % starting point
[T,Zt] = ode45(@MyODE,range,z0);  % obligatory order of arguments
plot(T,Zt)                        % plot the result
```

Note that `Zt(10,2)` contains the value of s at time `T(10)`.

Question: numerically solve the above differential equation till $s = \frac{1}{2}$.

Answer: create a function m-file that reaches the value zero when the integration is to be halted,

```
function [value,isterminal,direction] = MyCriterium(p,q)
                     % again, to be invoked with two arguments
value = q(2) - 0.5;  % crossing zero is the "event"
direction = 0;       %  0 : every crossing is an event
                     % +1 : only increasing through zero is an event
                     % -1 : only decreasing through zero is an event
isterminal = 1;      % stop (1) or don’t stop (0) ode at event
```

Initialize variables in the ‘old’ way, and instruct `ode45` to detect events

```
options = odeset(’Events’,@MyCriterium);
[T,Zt] = ode45(@MyODE,interval,z0,options);
plot(T,Zt)
```

Advanced Programming in Engineering
my own ODE solver

Question: write a function to numerically solve $y(x)$ from

$$\frac{dy}{dx} = f(y) = 1 - \tfrac{1}{2}y.$$

Numerically solve $y(x)$ over the interval $0 \leq x \leq 10$, given that $y(0) = 0$.

Answer: integration using simplest scheme,

```
function [X,Y] = MyODEsolver( xmin, xmax, dx, y0 )
y    = y0;
Y(1) = y0;
X    = xmin:dx:xmax;
for x = X(1:end-1)
    y = y + ( 1 - 0.5 * y ) * dx;
    Y = [ Y y ];
end % x
end
```

To run my ODE solver,

```
[X,Y] = MyODEsolver( 0., 10., 0.1, 0. );
plot(X,Y);
```

Question: write a function to numerically solve $y(x)$ from

$$\frac{dy}{dx} = f(x, y)$$

for any function f.

Answer: integration using simplest scheme,

```
function [X,Y] = MyODEsolver( f, xmin, xmax, dx, y0 )
y    = y0;
Y(1) = y0;
X    = xmin:dx:xmax;
for x = X(1:end-1)
    y = y + f(x,y) * dx;
    Y = [ Y y ];
end % x
end
```

To run my ODE solver, create an m-file containting your function

```
function dydx = MyODE( x, y )
dydx = 1 - 0.5*y;
end
```

and pass it as an argument to your ODE solver

```
[X,Y] = MyODEsolver( @MyODE, 0, 10, 0.1, 0 );
plot(X,Y);
```