`ode45` Differential Equation Solver

This routine uses a variable step Runge-Kutta Method to solve differential equations numerically. The syntax for `ode45` for first order differential equations and that for second order differential equations are basically the same. However, the `.m` files are quite different.

I. First Order Equations $\begin{cases} y' = f(t,y) \\ y(t_0) = y_0 \end{cases}$

A. Create a `.m` file for $f(t,y)$ (see the tutorial on numerical methods and `.m` files on how to do this). Save file as, for example, `yp.m`.

B. *Basic syntax for* `ode45`. At a MATLAB prompt type:

```
[t,y]=ode45('yp',[t0,tf],y0);
```

(your version of `ode45` may not require brackets around `t0,tf`)

$$\begin{cases} \texttt{yp} = \text{the } \texttt{.m} \text{ file of the function } f(t,y) \text{ saved as } \texttt{yp.m} \\ \texttt{t0,tf} = \text{initial and terminal values of } t \\ \texttt{y0} = \text{initial value of } y \text{ at } \texttt{t0} \end{cases}$$

C. For example, to numerically solve $\begin{cases} t^2 y' = y + 3t \\ y(1) = -2 \end{cases}$ over $1 \le t \le 4$:

- ∗ Create and save the file `yp.m` for the function $\frac{1}{t^2}(y+3t)$.
- ∗ At a MATLAB prompt type:

  ```
  [t,y]=ode45('yp',[1,4],-2);
  ```

  (your version of `ode45` may not require brackets around `1,4`)
- ∗ To print results type: `[t,y]`
- ∗ To plot results type: `plot(t,y)`
- ∗ To plot results type with a '+' symbol: `plot(t,y,'+')`

II. Second Order Equations $\begin{cases} y'' + p(t)y' + q(t)y = g(t) \\ y(t_0) = y_0,\ y'(t_0) = y_1 \end{cases}$

A. First convert 2^{nd} order equation to an equivalent system of 1^{st} order equations. Let $x_1 = y$, $x_2 = y'$:

$$\begin{cases} x_1' = x_2, \\ x_2' = -q(t)x_1 - p(t)x_2 + g(t) \\ x_1(t_0) = y_0,\ x_2(t_0) = y_1 \end{cases}$$

B. Create and save a `.m` file which will return a *vector-valued* function. This is a little tricky so here is a specific example.

* Suppose the system is as below and $0 \le t \le 4$

$$\begin{cases} x_1' = x_2, \\ x_2' = -t\,x_1 - e^t x_2 + 3\sin 2t \\ x_1(0) = 2,\ x_2(0) = 8 \end{cases}$$

* Create the following function file and save it as `F.m`:

```
function xp=F(t,x)
xp=zeros(2,1); % since output must be a column vector
xp(1)=x(2);
xp(2)=-t*x(1)+exp(t)*x(2)+3*sin(2*t);
```

(Don't forget the ";" after each line.)

C. *Basic syntax for* `ode45`. At a MATLAB prompt, type:

```
[t,x]=ode45('F',[t0,tf],[x10,x20]);
```

$$\begin{cases} \texttt{F} = \text{the } \texttt{.m} \text{ file of the vector-function saved as above} \\ \texttt{t0,tf} = \text{initial and terminal values of } t \\ \texttt{x10} = \text{initial value of } x_1 \text{ at } \texttt{t0}\text{: } x10 = x_1(t_0) \\ \texttt{x20} = \text{initial value of } x_2 \text{ at } \texttt{t0}\text{: } x20 = x_2(t_0) \end{cases}$$

(The example above becomes: `[t,x]=ode45('F',[0,4],[2,8]);`)

* Since $x_1(t) = y$, to print out the values of the solution y for $t_0 \le t \le t_f$, at a MATLAB prompt type: `[t,x(:,1)]`
* To plot the solution on a graph t vs y, type: `plot(t,x(:,1))` (since the vector $\mathbf{x}$ has 1st component $x_1 = y$ and 2nd component $x_2 = y'$.)
* To plot x_1 vs x_2 (phase plane) type: `plot(x(:,1),x(:,2))`