Numerical Methods & .m Files

In order to use MATLAB routines for the Euler, Improved Euler or Runge-Kutta Methods, you will need the files **eul.m, rk2.m** or **rk4.m**, respectively. These files are already present on all ITaP machines as standard software. (If using your own copy of MATLAB you may need to download these files from **http://math.rice.edu/~dfield**) You may also access these files from Matlab 7 via the *Citrix Client* :
https://goremote.ics.purdue.edu/Citrix/MetaFrame/site/default.aspx

- You must first create a function file in the same directory (or folder) as your MATLAB. Here is one way. After MATLAB has been opened, pull down the **File** menu and select "**New M-File**". A window will pop up for you to create your function file. For example, to create a function file for the function $f(x,y) = 6x^3 - e^{2y} + \frac{\sqrt{x}}{y}$, type :

```
function W=fcn1(x,y)
W=6*x^3-exp(2*y)+sqrt(x)/y;
```

  (Don't forget the "**;**" at the end.)

- Save this file as a **.m file** with the SAME name as your function. The above example would be saved as "**fcn1.m**". You can check if your function has been saved by typing something like the following at a MATLAB prompt: `fcn1(0,3)`

  You should get the value of $f(0,3)$.

- Your initial value problem should have the form : $\begin{cases} y' = f(x,y) \\ y(x_0) = y_0 \end{cases}$.

  Assuming $f(x,y)$ was saved as the file **fcn1.m**, the syntax for **eul** (as well as **rk2** and **rk4**, just replace **eul**) will be: `eul('fcn1',[x0,xf],y0,h)`

  where `x0` and `xf` denote the initial and final values of x, respectively, `y0` is the initial value of y, and `h` is the step size.

  (Your version of MATLAB may not utilize brackets. Type **help eul** to find out.) To approximate the actual solution to the IVP at `xf`, with given `h`, using **eul**, just type the following at a MATLAB prompt:

```
>>   [x,y]=eul('fcn1',[x0,xf],y0,h);
```

  The approximations $y_0, y_1, y_2, \ldots, y_n$ are stored in the matrix **y**.
  To print them out, type: `[x,y]`
  To plot them, type: `plot(x,y)`