MATH.2360 Engineering Differential Equations
Euler's Method, the Runge-Kutta Method, and *ode45*

1. You will need Professor White's m-files euler.m and rk4.m. To download these files, go to our course web page. Under **Course Materials** click on *Class handouts etc.*. Scroll down to the MATLAB Handouts section and look for the first Numerical Methods entry. Right-click "euler.m," click on "Save Target As" (or "Save Link As" depending on which browser you use) and save the file. Right-click "rk4.m," click on "Save Target As" and save the file. Take note of where you saved the files.

2. Start MATLAB and change directories to the directory where you saved euler.m and rk4.m. To change directories, click on the toolbar icon that looks like a file folder with a green arrow coming out of it.

3. Before you use euler and rk4, you will need to create a file defining the right-hand side of the d.e. you want to solve. Click on New Script on the toolbar. An Editor/Debugger box will open on the screen. For the homework problem from section 2.4, type in the following three lines (including the last line with the word end):

   ```
   function z = f(x, y)
   z=2*cos(x)+y;
   end
   ```

   (Don't forget the semicolon at the end of the second line.)

   Save the file using file name f, then return to the MATLAB command window.

4. To use Euler's Method on the homework problem $dy/dx - y = 2\cos(x),\ y(0) = -2,$ for $0 \le x \le 2$ with $n = 10$ subintervals, type the commands

   ```
   a = 0;
   b = 2;
   yi = -2;
   n = 10;
   [x, y] = euler('f', a, b, yi, n)
   ```

   The vector y contains the computed y values, and the vector x contains the x values at which the solution was computed.

   If you wanted to graph the solution, **which you do not want to do for this problem**, you would type the command

   ```
   plot(x, y)
   ```

OVER

5. Change the number of intervals to 20 and run Euler's Method again by typing the commands

```
n = 20;
[x, y] = euler('f', a, b, yi, n)
```

6. To use the Runge-Kutta Method on the problem
$dy/dx - y = 2\cos(x),\ y(0) = -2,$ for $0 \le x \le 2$ with 2 subintervals, type the commands

```
a = 0;
b = 2;
yi = -2;
n = 2;
[x, y] = rk4('f', a, b, yi, n)
```

Note that you didn't have to create a new m-file specifying the differential equation because this is the same problem you just solved with Euler's method, so you already had the m-file.

7. Change the number of subintervals to 4 and run the Runge-Kutta method again by typing the commands

```
n = 4;
[x, y] = rk4('f', a, b, yi, n)
```

8. (For future use.) To use the built-in MATLAB routine *ode45* to generate an approximate solution to the problem $dy/dx - y = 2\cos(x),\ y(0) = -2,$ for $0 \le x \le 2$, type the commands

```
a = 0;
b = 2;
yi = -2;
[x, y] = ode45('f', [a, b], yi)
```

Note that you didn't have to create a new m-file specifying the differential equation because this is the same problem you just solved with Euler's method and the Runge-Kutta method, so you already had the m-file. For a different differential equation you would have to create a new m-file. Note also that you do not have to specify the step size with *ode45*.

If you want to solve the same d.e. with two different initial conditions, say $y(0) = -2$ and $y(0) = -1$, you can use the commands

```
a = 0;
b = 2;
yi = [-2, -1];
[x, y] = ode45('f', [a, b], yi)
```

Use the command

plot(x, y)

if you want to see a graph of the computed solutions.