University of Pennsylvania

EAS 105: Fall 2011

MATLAB PROJECT: Initial Value Problems

Instructor: Donna Dietz

For this project, you will write a variety routines which calculate solutions to Initial Value Problems (IVPs). Your code will be tested for correctness against a sizeable bank of solutions, contained in a .mat file which will be provided. You will also be provided with a file called *TestFunctionList.m* It is called using the command c=TestFunctionList(); which results in a cellarray, c, being created. This has all your test problems in it. The cellarray c will have size 12, each cell having *four* elements. The first is an anoymous function of two variables, which we will call f. The second is the vector of t-values, the third is the initial value, the fourth is an anonymous function (to be used for comparision purposes) is the actual solution to the IVP. (This is the really-truly actual value, the one and only!)

We will discuss Matlab's version of anonymous functions in class. You may also read about them online.

The *Tester* will not work without these supplied files. Additonally, you will be creating plots with legends for each problem set, with each solution type having a different color.

If you don't know what an IVP is, check your class notes, or ask for help. Any integration problem can be posed as an IVP, but the class of IVPs is larger than just integration. IVPs are a subset of differential equations. All the routines you will write for this project will be numerical solvers for problems which can be posed as:

$$y'(t) = f(t, y) \quad y(t_0) = y_0$$

For each of the five IVP-solving functions (Euler, Midpoint, Heun, ModifiedEuler, and RK4), the inputs will be f, the t-vector, and the initial value. The size of the output, w, is the same as the size of t, so we do not have to pass back a different t-vector as we did the Numerical Differentiation project.

For this project, create these six functions:

function [w] = Euler(f, t, init) Let h be the common difference in the t-values. Let $w(1) = init$, and thereafter,

$$w(i) = w(i-1) + hf\left(t(i-1)\ ,\ w(i-1)\right);$$

Remember that even though f takes in t and y, we do not know y, which is why are doing all this work in the first place. So, we use our estimate, w, in place of y. You will end up using a loop for this, unlike what you did in the Numerical Differentiation project, because you need an estimate to $y(i)$ before calculating the estimate to $y(i+1)$.

function [w] = Midpoint(f, t, init) Let h be the common difference in the t-values. Let $w(1) = init$, and thereafter,

$$w(i) = w(i-1) + hf\left(t(i-1) + \frac{h}{2}\ ,\ w(i-1) + \frac{h}{2} f\left(t(i-1), w(i-1)\right)\right);$$

function [w] = Heun(f, t, init) Let h be the common difference in the t-values. Let $w(1) = init$, and thereafter,

$$w(i) = w(i-1) + \frac{h}{4}\left(f(t(i-1)\ ,\ w(i-1)) + 3f(t(i-1) + \frac{2}{3}h\ ,\ w(i-1) + \frac{2}{3}hf(t(i-1), w(i-1)))\right);$$

function [w] = ModifiedEuler(f, t, init) Let h be the common difference in the t-values. Let $w(1) = init$, and thereafter,

$$w(i) = w(i-1) + \frac{h}{2}\left(f(t(i-1)\ ,\ w(i-1)) + f(t(i)\ ,\ w(i-1) + hf(t(i-1), w(i-1)))\right);$$

function [w] = RK4(f, t, init) Let h be the common difference in the t-values. Let $w(1) = init$, and thereafter,

$$\begin{aligned} k_1 &= hf(t(i-1)\ ,\ w(i-1)); \\ k_2 &= hf(t(i-1) + \frac{h}{2}\ ,\ w(i-1) + \frac{1}{2}k_1); \\ k_3 &= hf(t(i-1) + \frac{h}{2}\ ,\ w(i-1) + \frac{1}{2}k_2); \\ k_4 &= hf(t(i)\ ,\ w(i-1) + k_3); \end{aligned}$$

$$w(i) = w(i-1) + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4);$$

function PlotTests(c) See the specifications for PlotDiffTests in the Numerical Differentiation project. This the the same thing.

For this project, you will want to run the *Tester* early and often, to make sure you have everything just right!