Recitation 9: Wednesday, 11 April / Friday, 13 April
MATLAB Exercises_Recitation 9 due: Friday, 13 April 2012 at 5 PM by upload to Stellar

Format for upload: Students should upload to the course Stellar website a folder

YOURNAME_MatlabExercises_Rec9

which contains the completed scripts and functions for the assigned MATLAB Exercises_Recitation 9: all the scripts should be in a single file, with each script preceded by a comment line which indicates the exercise number; each function .m file should contain a comment line which indicates the exercise number.

Note early upload given the Patriot’s Day holiday.

Please review Sections 6.1–6.4.

New reading: Section 6.5 (on function handles).

Optional reading (not required): Section 6.6 (on anonymous functions).

Consider the following scalar ODE IVP

$$\frac{du}{dt} = -(u^4), \quad 0 < t \leq 1; \quad u(t = 0) = 1.$$  

This equation is a simple model for thermal “dunking” with radiative (rather than convective) heat transfer from the body to the environment. The exact solution to this (nonlinear) equation is

$$u(t) = \frac{1}{(1 + 3t)^{1/3}},$$

which note is no longer exponential.

We denote the Euler Forward approximation to $u(t)$ as $\tilde{u}(t^j)$, $0 \leq j \leq J$, where $t^j = j \Delta t$, $1 \leq j \leq J$, and $J \Delta t = 1$. Explicit techniques can be convenient for nonlinear ODEs and also quite efficient unless the stability restrictions are severe. (In practice, RK4 would be a better choice than EF for an explicit scheme — but we leave RK4 for Problem Set 4.)

We provide you with the two MATLAB functions: EF_general

```matlab
function [ u_tfinal, uvec, tvec ] = EF_general( g, deltat, J, u0 )
tvec = deltat*[0:J];
uvec = zeros(1,J+1);
uvec(1,1) = u0;

for j = 2:J+1
    uvec(1,j) = uvec(1,j-1) + g( uvec(1,j-1),tvec(j-1) ) * deltat;
end
u_tfinal = uvec(1,end);
```
return
end

and g_rad

function [ gval ] = g_rad( u, t )
gval = - ( u^4 );
return
end

which you may copy-paste and use in this exercise.

Your deliverable: write a one-line script which computes and displays |u(t = 1) - \tilde{u}(t_J = 1)|
for \Delta t = 0.01 and J = 100. Your one line of code should involve both EF_general and g_rad.