CS 257: Numerical Methods
Spring 2006
Homework, Set 11 **Due May 2nd (officially you can hand in until May 4th, at the end of the class) , 2006**

---

1. Euler Consider a IVP with two solutions:

$$y' = y^{1/3}$$
$$y(0) = 0$$

The two solutions are $y_1(t) = 0$ and $y_2(t) = (\frac{2}{3}t)^{3/2}$. What happens when Euler's Method is applied to the IVP?

2. Euler Chapter 12 NMM: #2, #3

3. Euler/Midpoint/RK4 Consider the IVP

$$\frac{dy}{dt} = t - 2ty$$
$$y(0) = 1$$

The exact solution is

$$y(t) = \frac{1}{2}\left[1 + e^{-t^2}\right]$$

Use Euler's Method, Midpoint Method and RK4 with a sequence of decreasing h. Submit a log-log plot of h versus cpu-time and a plot of h versus maximum error with the three methods on each plot. Assume x-axis versus y-axis format. Do these plots confirm the theoretical accuracy of each of the methods?

4. ode45,ode23s Consider a flame problem (from the Mathworks). If we light a match, the ball of flame grows rapidly until it reaches a critical size (or steady state). It remains at this size due to the level of oxygen in the combustion process; there is a balance between the interior fuel and the surface fuel. A simple IVP models this process quite well:

$$y' = y^2 - y^3 \qquad 0 \le t \le 2/\delta$$
$$y(0) = \delta$$

Here $y(t)$ is the radius of the flame (y^2 and y^3 result from the surface and volume contributions). The initial radius δ is the important parameter.

```
delta = 0.01;
F=inline('y^2 - y^3','t','y');
opts=odeset('RelTol',1.0e-4);
ode45(F,[0 2/delta],delta,opts);
```

With no output arguments, the adaptive solver is displayed at each iteration. Use output arguments of `[t,y]` if you want to save this data. The length of the output arguments will tell you how many iterations `ode45` needed. Try decreasing δ to 0.0001. This is much harder.

- hand in a plot of the approximation for $\delta = 0.0001$

- zoom in around the point where the radius first reaches its maximum (the upper "step") and hand in a plot
- Discuss the behavior you see. Why is ODE45 performing like this?
- Run "`ode23s`" and discuss in 1-2 sentences what you see.
- Compare the time and number of steps needed for `ode45` and for `ode23`