**Chemical Engineering 310**
**Tips for solving nonlinear ODEs**

Nonlinear ordinary differential equations (ODEs) can be recalcitrant. Here are a couple tips to help you use the routines needed for the efflux from a tank experiment.

**(1) The routine sysodeNR.m requires certain input values sysodeNR(m,n,xo,xf,yo,dydxo)**

- "m" is the ODE-solver method. Set m=2 for Classical Fourth-Order Runge-Kutta.
- "n" is the number of Runge-Kutta steps. Choose a number of steps so that each step size is 1 second for long runs and 0.5 seconds for short runs. A good value of "n" depends on the choice of "xf".
- "xo" is the starting time value. It will always be zero.
- "xf" is the final time value. This value depends on how long you expect the tank to drain. Try small values. If the tank hasn't drained to one inch in that time, try larger values until it has. Choosing a xf that is too long may result in a program crash since the ODE may not have solutions for negative tank heights.
- "yo" is the initial tank height. Make sure you enter the correct value.
- "dydxo" is the initial guess for the tank velocity, needed because not only is the ODE nonlinear (requiring Runge-Kutta) but the velocity appears in an algebraically nonlinear manner (requiring Newton-Raphson at each Runge-Kutta evaluation). A good initial guess for dydxo is the average tank velocity (the total tank height divided by the total emptying time.) This will change for each run. Use the experimental average velocity first. If that causes the program to crash. Try the theoretical average velocity obtained from the analytical expression for Case One.

You should not have to alter any lines in sysodeNR.m.

**(2) The file sysodeNRinput.m contains the physical parameters and the ODE.**

- "L" is the pipe length. It changes for each run.
- "dp" is the pipe diameter. It changes for each run.
- "g" is gravity. "mu" is viscosity. "rho" is density. "dta" is tank diameter. "hp" is h prime. These values remain constant for all runs.
- the last line of **sysodeNRinput.m** defines the variable f. f is the function (the mechanical energy balance) of all the parameters, height, time, and tank velocity, which, when the correct tank velocity is determined, equals zero. At every time, this routine iteratively chooses new values of tank velocity until it converges on a value that satisfies the mechanical energy balance. Since system is not at steady state, a different value of the velocity is required at different times.

These routines work. I have tested them on experimental data. With the correct values of dydxo and the rest of the parameters, every run can be numerically simulated. You may have to play around with your initial guesses for a while before the code yields sensible results. Don’t be afraid to open the sysodeNR.m file and see how the thing is working.