AMS 27L **LAB #9** **Winter 2009**

Solving ODE's in Matlab

Objectives:

1. To solve ODE using ode45 and ode15s
2. To solve higher-order ODEs

1 Matlab's ODE Suite

In the previous lab we were introduced to Matlab's suite of ODE solvers including: `ode23, ode45, ode15s, ode23s, ode23t, ode23tb, ode113` and `ode15i`. The types of differential equation solvers include nonstiff and stiff problems, and fully implicit ODEs. If solutions to differential equations have components which vary at different rates, by a couple of orders of magnitude, the equations are called stiff. The Matlab suite of ODE solvers includes routines designed to solve stiff equations, including `ode15s, ode23s, ode23t` and `ode23tb`. `ode15s` is the first of these to try.

Rule of Thumb: Try to solve using `ode45`. If that fails, or is way slow, try `ode15s`.

Problem #1. The system

$$\epsilon \frac{dx}{dt} = x(1-x) - \frac{(x-q)}{(q+x)} fz \tag{1}$$

$$\frac{dz}{dt} = x - z \tag{2}$$

models an oscillating chemical reaction called an *oregonator* – (the cool name was given by the authors who, at that time, worked at the University of Oregon). Suppose that $\epsilon = 10^{-2}$ and $q = 9 \times 10^{-5}$. Put the system into normal form and write and ODE function M-file for the system that passes f as a parameter. The idea is to vary the parameter f and note its affect on the solution of the oregonator model. We will use the initial conditions $x(0) = 0.2$ and $z(0) = 0.2$ and the solution interval $[0, 50]$.

(a) Use `ode45` to solve the system with $f = 1/4$. This should provide no difficulties.

(b) Use `ode45` to solve the system with $f = 1$. This should set off all kinds of warning messages.

(c) Try to improve the accuract with `options = odeset('RelTol',1e-6)` and using `options` as the options parameter in `ode45`. You should fine that this slows computation to a crawl as the system is very stiff.

(d) The secret is to use the `ode15s` instead of `ode45`. Then you can note the oscillations in the reaction.

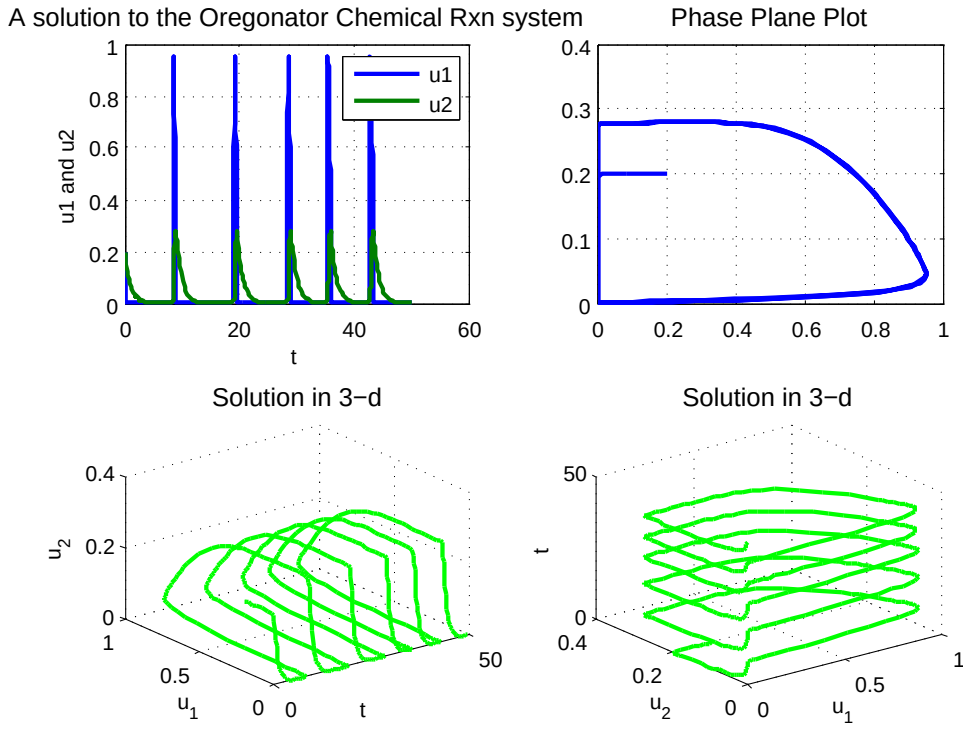

Figure 1: Oscillating Oregonator system with $f = 1$.

(e) Play around with making 3-d plots of the solution to the Oregonator chemical reaction ODE system by using the 3-d plotting command used in the previous lab. Evaluate the solution for different values of f. The phase plot and solution are graphed in Figure 1 with $f = 1$.

2 Second Order Differential Equations

In the previous lab we solved single and systems of first order equations. To solve a single second order differential equation it is necessary to replace it with the equivalent first order system. For example,

$$y'' = f(t,\, y,\, y') \tag{3}$$

we set $x_1 = y$, and $x_2 = y'$. Then $\mathbf{x}$= $[x_1, x_2]^T$ is a solution to the first order system

$$\begin{aligned} x_1' &= x_2, \\ x_2' &= f(t,\, x_1\,, x_2). \end{aligned}$$

Since $\mathbf{x} = [x_1, x_2]^T$ is a solution of the system, we then set $y = x_1$. Then we have $y' = x_1' = x_2$, and $y'' = x_2' = f(t, y, y')$. Thus, y is a solution of the equation in (3).

Example 1. Plot the solution of the initial value problem

$$y'' + yy' + y = 0, \quad y(0) = 0,\ y'(0) = 1 \tag{4}$$

on the interval $[0, 10]$.

First, solve equation (4) for y'', so that

$$y'' = -yy' - y$$

Introduce new variables for y and y', then

$$x_1 = y \quad and \quad x_2 = y'$$

Then we have that

$$\begin{aligned} x_1' &= y' = x_2 \\ x_2' &= y'' = -yy' - y = -x_1x_2 - x_1 \end{aligned}$$

We can vectorize the ODE problem to use Matlab's ODE solvers by

$$F(t, [x_1, x_2]^T) = \begin{bmatrix} x_2 \\ -x_1x_2 - x_1 \end{bmatrix} \tag{5}$$

Let's again use a function M-file to solve the second-order ODE using `ode45`.

Problem #2. I've written a function that solves the ODE in **lab9ex1.m**. Download the M-file from our course webpage at `http://www.soe.ucsc.edu/classes/ams027l/Winter09/` and give it a go by typing `lab9ex1` at the Matlab prompt.

The code in `lab9ex1.m` contains a subfunction, `dfile`, that implements the function F defined in equation (5). It's important to remember that the initial ODE asked for the solution of $y'' + yy' + y = 0$. Since $y = x_1$, the first plot command in `lab9ex1.m` will plot y versus t, as shown in Figure 2. The second plot command displays a phase-plane diagram as shown in Figure 2.

Problem #3. The system

$$\begin{aligned} m_1x'' &= -k_1x + k_2(y - x) \\ m_2y'' &= -k_2(y - x) \end{aligned}$$

model a *coupled* oscillator. Imagine a spring (with spring constant k_1), attached to a hook in the ceiling. Mass m_1 is attached to the spring, and a second spring (with spring constant k_2), is attached to the bottom of mass m_1. If a second mass, m_2, is attached to the second spring, you have a coupled oscillator, where x and y represent the displacements of masses m_1 and m_2 from their

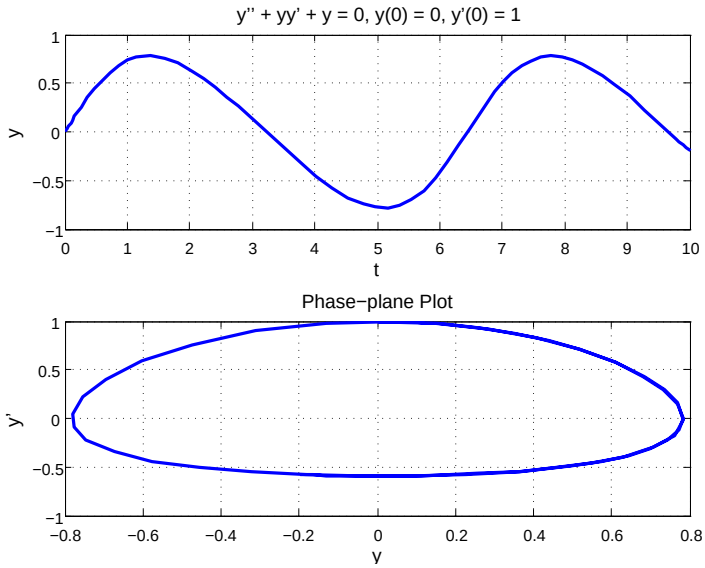

Figure 2: Solution and phase plane plot of $y'' + yy' + y = 0$.

respective equilibrium positions.

If you set $x_1 = x$, $x_2 = x'$, $x_3 = y$, and $x_4 = y'$, you can show that:

$$\begin{aligned} x_1' &= x_2 \\ x_2' &= -\frac{k_1}{m_1}x_1 + \frac{k_2}{m_1}(x_3 - x_1) \\ x_3' &= x_4 \\ x_4' &= -\frac{k_2}{m_2}(x_3 - x_1) \end{aligned}$$

(a) Draw a diagram of the coupled oscillator and label the diagram denoting the variables m_1, m_2, x and y.

(b) Assume $k_1 = k_2 = 2$ and $m_1 = m_2 = 1$. Create an ODE file to solve the system of equations. Suppose that the first mass is displaced upward two units, the second downward two units, and both masses are released from rest. Plot the position of each mass versus time.

Quit MATLAB by clicking on the `File` menu in the upper left corner and choosing `Exit`. Please remember to **Log Off** (from the "Start" menu in the lower left of the screen).