Review 1, Mathematical Modeling

In this part of the modeling course, we have looked at building models by theory- That is, we build differential equations (or difference equations) with specific assumptions about the phenomenon being modeled.

Part of the modeling course is building the model, but just as importantly, we need to be able to analyze our model in order to see if it properly captures what we think it ought to capture, and to see if it reflects that part of nature that we wish to capture in the model.

This exam will be designed to be approximately 50 minutes, but you'll be given 90 minutes to take it, plus 15 minutes to scan/upload your work. Generally speaking, the exam will cover topics from Chapters 7,8,9 from the Boyce and Diprima text, with some Octave content as well.

Topics from Chapter 7

Here the main idea is to be able to solve a linear first order system of differential equations. In particular, we looked at the details when A is 2×2:

$$\mathbf{x}' = A\mathbf{x} \quad \Rightarrow \quad \begin{bmatrix} x_1'(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

(Typically, we supress the dependence on time). To solve this differential equation, we began with the ansatz: $\mathbf{x}(t) = \mathrm{e}^{\lambda t}\mathbf{v}$, and we showed that λ had to be an eigenvalue of A, and $\mathbf{v}$ a corresponding eigenvector. Finding the eigenvalues means solving the characteristic equation:

$$|A - \lambda I| = 0 \quad \Rightarrow \quad \lambda^2 - (a+d)\lambda + (ad - bc) = 0 \quad \Rightarrow \quad \lambda^2 - \mathrm{Tr(A)}\lambda + \det(A) = 0$$

We have three cases, dependent on the discriminant $\Delta = \mathrm{Tr}(A)^2 - 4\det(A)$.

Solutions to $\mathbf{x}' = A\mathbf{x}$

- If $\Delta > 0$, then we have two real, distinct eigenvalues. The general solution to the DE is:

$$\mathbf{x}(t) = C_1\mathrm{e}^{\lambda_1 t}\mathbf{v}_1 + C_2\mathrm{e}^{\lambda_2 t}\mathbf{v}_2$$

- If $\Delta < 0$, then we have complex conjugate eigenvalues. Using just $\lambda = \alpha + \beta i$, and the corresponding eigenvector, the general solution to the DE is:

$$\mathbf{x}(t) = C_1\mathrm{Real}(\mathrm{e}^{\lambda t}\mathbf{v}) + C_2\mathrm{Imag}(\mathrm{e}^{\lambda t}\mathbf{v})$$

- If $\Delta = 0$ AND the dimension of the eigenspace is 2, then go to the first case.

  If $\Delta = 0$ and the initial condition is (x_0, y_0), then we compute a vector $\mathbf{w}$:

$$\begin{aligned} (a-\lambda)x_0 + by_0 &= w_1 \\ cx_0 + (d-\lambda)y_0 &= w_2 \end{aligned}$$

  And the solution to the DE is given by:

$$\mathbf{x}(t) = \mathrm{e}^{\lambda t}\left(\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + t\mathbf{w}\right)$$

  For a discussion of this version versus the one in the textbook, go to the last page of the review. You can use either method, but this one gives you the solution in terms of x_0, y_0 rather than having to solve for two constants C_1, C_2.

Three dimensional systems

You may be given a system of three equations to work with as well. In that case, you'll be able to use Octave to help find eigenvalues and eigenvectors.

Classify the Equilibrium

Be able to use the Poincare Diagram to classify equilibria (for two dimensional systems).

Other notes about systems

- We should be able to convert any n^{th} order differential equation into an equivalent system of n first order equations, and convert 2×2 systems of first order into an equivalent differential equation of second order.
- We can also solve a system of first order differential equations using the techniques we learned to solve a second order differential equation ("Chapter 3 techniques" outlined in the notes).
- We can solve a system of first order equations by first creating dy/dx, then solving using our usual first order methods (first order linear equation with an integrating factor, separable differential equation).

Topics from Chapter 8, Numerical Solvers

- Be able to describe how Euler's method (the forward version) works. By hand, be able to compute a step or two forward using the method. For this method, we should know that the local truncation error is proportional to h^2, and the global truncation error is proportional to h.
- For backwards Euler, understand what an implicit method is, and the approximation we use to write our own Euler's backward method.
- Similarly, be able to describe the reasoning behind Euler's improved method (the implicit version), and the approximation that allows us to write our own version in Octave.
- Understand how the improved Euler's method takes us naturally to consider other weighted averages, leading us to the Runge-Kutta method of order 4.
- Be able to numerically solve a system of differential equations using one of the methods we've discussed in Chapter 8, or use a method built-in to Octave (like `ode23`).

Topics from Chapter 9, Nonlinear Differential Equations

We might be able to solve a given system of nonlinear differential equations directly. For example, given dx/dt and dy/dt, we might compute dy/dx and solve it that way. There is no general analytic method to solve systems of nonlinear differential equations, but we can do graphical analysis and analysis by local linearization.

Analysis by Local Linearization

First, we defined what we meant by local linearization. Generally, the idea is to replace the nonlinear function f by a linear function, based at some point (our point will be an equilibrium solution).

In the most general case, the linearization is given below at a point $\mathbf{x} = \mathbf{a}$:

$$\mathbf{y} = \vec{F}(\mathbf{x}) \quad \Rightarrow \quad L(\mathbf{x}) = \vec{F}(\mathbf{a}) + D\vec{F}(\mathbf{a})(\mathbf{x} - \mathbf{a})$$

Written out using the coordinate functions:

$$\vec{F}(\mathbf{x})\begin{bmatrix} f_1(x_1,\cdots,x_n) \\ f_2(x_1,\cdots,x_n) \\ \vdots \\ f_m(x_1,\cdots,x_n) \end{bmatrix} \quad \Rightarrow \quad L(\mathbf{x}) = \begin{bmatrix} f_1(a_1,\cdots,a_n) \\ f_2(a_1,\cdots,a_n) \\ \vdots \\ f_m(a_1,\cdots,a_n) \end{bmatrix} + \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \begin{bmatrix} x_1 - a_1 \\ x_2 - a_2 \\ \vdots \\ x_n - a_n \end{bmatrix}$$

The matrix is a matrix of first partial derivatives (called the Jacobian matrix), and here it has been evaluated at the point $\mathbf{x} = \mathbf{a}$, so this is a matrix of constants, not formulas!

Local Linear Analysis

Given a nonlinear first order system of differential equations:

1. Find all the equilbrium solutions: $\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_k$.
2. About each equilibrium:
   - Linearize the system
   - Use the Poincaré Diagram to classify the equilbrium.
3. Try to get a global picture as to what is happening. Sometimes it is helpful to get a direction field (or slope field) to clarify.

Models we have considered

- **The SIR model.** Given $s(t), i(t), r(t)$ as the susceptible, infected and recovered proportions of the population at time t, we constructed the model of sickness as:

  $$\begin{aligned} s' &= -\beta si \\ i' &= \beta si - \gamma i \\ r' &= \gamma i \end{aligned}$$

  You should be able to discuss the expressions you see in the DE in terms of the physical situation (for example, what is the significance of having an "si" term in the first equation? What happens when you add the three equations together- What is being assumed there?

- **Tank Mixing:** Recall that the rate of change of the quantity is modeled as "Rate In − Rate Out", and be sure you're making the units match up.

  We get a system of first order if we have more than one tank.

  - This exercise is also where we learned how to convert a differential equation of the form:

    $$\mathbf{x}' = A\mathbf{x} + \mathbf{b}$$

    into a differential equation of the form: $\mathbf{u}' = A\mathbf{u}$. You'll notice that this is the DE we solve in Chapter 7.

- **Population Models:** Population models are excellent template models to know. From Math 244, we have the *exponential growth model*, $y' = ky$, and the *logistic growth model*, $y' = ky(b - y)$, which incorporates an environmental threshold. We reviewed how to analyze these models using the (y, y') graph to find and classify the equilibrium solutions.

- **Competition Between Species**

- **Predator-Prey Models**

Introduction to Octave

Be able to:

1. Be able to write a vector or matrix using Octave. For example, if A is a matrix, how would we denote the 5^{th} row of A? Or the 3^{rd} column of A?
2. Be able to find eigenvalues and eigenvectors of a matrix (command: `eig`)
3. Be able to modify given code- You won't have to write much if at all from scratch.
4. Solve a system of first order DEs in Octave.
5. Be able to plot the output from a DE solver (plot solution curves in two and three dimensions).

Textbook versus Notes on the Third Case

This has to do with solving $\mathbf{x}' = A\mathbf{x}$ when the eigenvalue is doubled up and you get only one eigenvector.

- Our class notes said:

  If $\Delta = 0$ and the initial condition is (x_0, y_0), then we compute a vector $\mathbf{w}$:

  $$\begin{aligned} (a-\lambda)x_0 + by_0 &= w_1 \\ cx_0 + (d-\lambda)y_0 &= w_2 \end{aligned}$$

  And the solution to the DE is given by:

  $$\mathbf{x}(t) = \mathrm{e}^{\lambda t}\left(\begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + t\mathbf{w})\right)$$

- Textbook version said:

  If $\Delta = 0$ AND the dimension of the eigenspace is 1 (so the matrix is defective), then we compute a second vector known as a generalized eigenvector $\mathbf{w}$ that satisfies the equation:

  $$(A - \lambda I)\mathbf{w} = \mathbf{v}$$

  Then the general solution to the DE is given by:

  $$\mathbf{x}(t) = \mathrm{e}^{\lambda t}\left(C_1\mathbf{v} + C_2(t\mathbf{v} + \mathbf{w})\right)$$

Which one should you use? You may use either, although I think the first one is easier to use, especially if you're solving a DE with initial conditions (since C_1, C_2 are solved for you in the first case).