M647 Spring 2011 Practice Problems for Final Exam

The final for M647 will be Wednesday May 11, 10:30 a.m. - 12:30 p.m. in Blocker 122 (the usual classroom). The exam will consist of two parts: Part 1 will not require MATLAB, while Part 2 will require MATLAB. Students will have to turn in Part 1 before starting Part 2, but for Part 2 students will have access to all M-files we've used this semester, from both lecture and homework. Students will be expected to access data files from the course web site.

The final will cover material from the second half of the semester, beginning with population dynamics and including classical mechanics (Newtonian, Lagrangian, Hamilitonian), solving ODE in MATLAB (including parameter estimation and confidence intervals), analysis of ODE models (exact solutions of linear constant-coefficient equations, equilibrium points, stability), modeling with PDE, and solving PDE in MATLAB (including parameter estimation and confidence intervals).

My office hours during finals will be 1:00-2:30 Wednesday, Friday, Monday and Tuesday.

Part One Problems

1. Consider a community of E. Coli which through mutation has divided itself into three general populations:

C: (*Colicinogenic* cells) Produce (and release) a toxin, colicin, which destroys E. Coli cells, and also produce a colicin-specific immunity protein, which renders the cell immune to the colicin.

R: (*Resistant* cells) Produce the colicin-specific immunity protein, which renders them immune to the colicin, but do not produce the colicin.

S: (*Sensitive* cells) Produce neither the immunity protein nor the colicin.

The production of either the toxin colicin or the colicin specific immunity protein requires a certain amount of effort, and the growth rate of S cells is greater than the growth rate of R cells, which in turn is greater than the growth rate of C cells. Consequently, S can displace R by outproducing it, R can displace C by outproducing it, and C can displace S by killing it with the toxin. The situation has been likened to the game rock–paper–scissors, for which rock crushes scissors, scissors cut paper, and paper covers rock. Write down a system of ODE that models the populations C, R, and S. Your model will involve a number of parameters that you would generally estimate from data. In lieu of this, discuss the relative values you would expect. For example, if you have growth constants r_1, r_2, and r_3, which would you expect to be largest?

2. Consider a simple pendulum as depicted in Figure 1 moving under the influence of gravity and air resistance. Explain why Lagrangian and Hamiltonian mechanics provide inappropriate frameworks for modeling this situation, and use Newtonian mechanics to write down an ODE describing the motion.

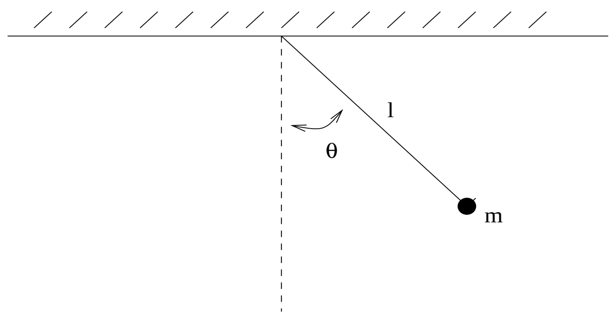

Figure 1: Figure for Problem 2.

3. Consider a pendulum of length l attached to a pivot that rotates with angular velocity ω along an upright circular frame with radius r (see Figure 2). Write down the Lagrangian for this system and also the Euler-Lagrange equations. Ignore friction and air resistance.

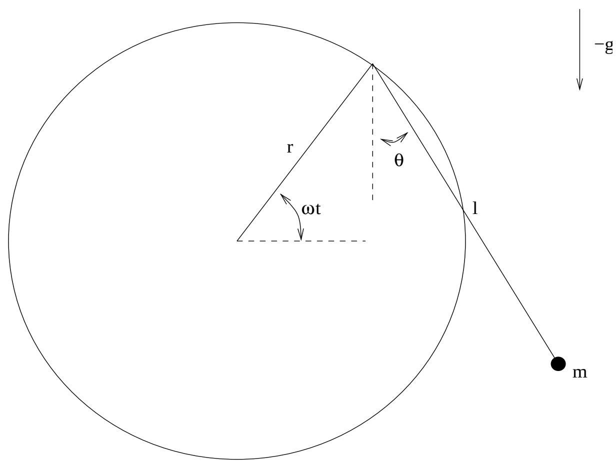

Figure 2: Figure for Problem 3.

4. Consider the mass-spring system depicted in Figure 3. Ignoring friction, write down the Hamiltonian for this system, and the associated Hamilton ODE system.

6. Non-dimensionalize the mutualism model

$$\frac{dy_1}{dt} = r_1 y_1 (1 - \frac{y_1}{K_1}) + b_1 y_1 y_2$$
$$\frac{dy_2}{dt} = r_2 y_2 (1 - \frac{y_2}{K_2}) + b_2 y_1 y_2.$$

7. Solve the ODE system

$$\frac{dy_1}{dt} = 4y_1 + y_2 + 6y_3; \quad y_1(0) = 1$$
$$\frac{dy_2}{dt} = -4y_1 - 7y_3; \quad y_2(0) = 1$$
$$\frac{dy_3}{dt} = -3y_3; \quad y_3(0) = 1.$$

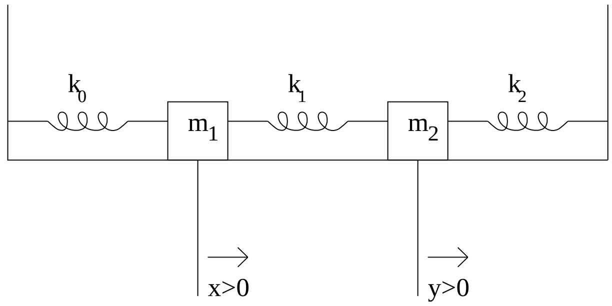

Figure 3: Figure for Problem 4.

8. Find all equilbrium points for the non-dimensionalized Lotka-Volterra competition model

$$\begin{aligned}\frac{dy_1}{dt} &= y_1(1 - y_1 - ay_2)\\ \frac{dy_2}{dt} &= by_2(1 - cy_1 - y_2),\end{aligned}$$

and classify each as unstable, stable, or asymptotically stable (to the extent that it can be decided by the Poincare-Perron Theorem). Take the parameters a, b, and c to be positive constants, and you need only consider equilibrium points for which neither population is negative.

9. A string of length L with constant density ρ is hanging from one end, under the influence of gravity (see Figure 4). Let the x-direction be downward from the string's top end, positioned at $x = 0$, and let $u(x, t)$ denote horizontal displacement from the vertical. Assuming small oscillations, find a PDE for u. Suggest appropriate initial and boundary conditions.

Note. Assume that the force of gravity at each point of the string equals the weight of the part of the string below that point, and is directed tangentially.

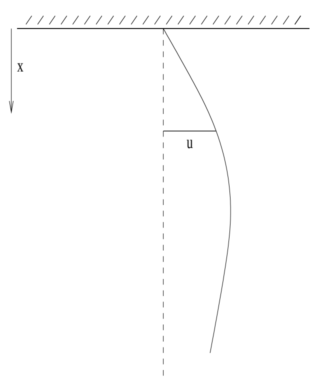

Figure 4: Vertical string.

10. Consider a tightly stretched membrane in two dimensions (such as a drumhead) or an elastic solid in three dimensions, and suppose motion is approximately confined to a single direction. (For example, in our derivation of the wave equation in class, the string could

in principle have moved both up-and-down and back-and-forth, but we assumed forces in the horizontal direction canceled, so that its direction was purely vertical.) Show that if u denotes distance in the preferred direction and the force at position $\vec{x}$ is $\vec{F} = \kappa D_x u$, where κ is a constant, then u solves the wave equation.

Note. Let Ω denote the region filled by the membrane or elastic solid, let ω denote any subset of Ω, and let $\rho(\vec{x})$ denote material density either in mass per unit area (membrane) or mass per unit volume (solid).

Part Two Problems

1. The Lotka-Volterra model with prey carrying capacity is

$$
\begin{aligned}
\frac{dy_1}{dt} &= ay_1(1 - \frac{y_1}{K}) - by_1y_2 \\
\frac{dy_2}{dt} &= -ry_2 + cy_1y_2.
\end{aligned}
$$

In this problem, we'll find values for a, b, c, r, and K using the data in *lvdata.mat* (available on the course web site).

1a. Use the method of derivative approximation to obtain initial approximations for a, b, c, r, and K. Plot solutions using these parameter estimates along with the data. What does this result suggest about including K in this model?

1b. Part (a) suggests that we cannot obtain a reasonable value for K by derivative approximation, but we can view the maximum observed prey population as a reasonable guess. Repeat Part (a) with this value of K to get corresponding first approximations for a, b, c, and r. Again, plot solutions using these parameters along with the data.

1c. Use nonlinear regression to refine your estimates from Part (b), and plot solutions to your refined model along with the data. When we carried out this fit in class without K, we found $s = 4.5519$. Discuss whether or not including K is justified.

1d. Find 95% confidence intervals for you values from Part (c). Again, what does your calculation suggest about including K in this model.

2. Consider a heat-conducting cylinder of length 1 that is insulated on both ends, but for which heat transfer occurs along the sides. If we assume Newton's law of cooling along the sides, the governing PDE is

$$
\begin{aligned}
u_t &= \mu u_{xx} - a(u - T_0) \\
u_x(0,t) &= 0; \quad t \geq 0 \\
u_x(1,t) &= 0; \quad t \geq 0 \\
u(x,0) &= 50e^{-5(x-.5)^2},
\end{aligned}
$$

where a, T_0, and μ are constants, and the initial condition states that the temperature of the bar is initially concentrated in the center of the bar. In this problem, we'll find values for a, T_0, and μ using data in *heatdata.mat* (available on the course web site).

2a. Use the method of derivative approximation to obtain initial approximations for a, T_0, and μ.

2b. Use nonlinear regression to refine your parameter estimates from Part (a).

2c. Find 95% confidence intervals for you values from Part (b).

Part One Solutions

1. The key to this problem is to recognize that the dynamic is primarily competitive. Set

$$\begin{aligned} C(t) &= \text{number of colicinogenic cells} \\ R(t) &= \text{number of resistant cells} \\ S(t) &= \text{number of sensitive cells.} \end{aligned}$$

Then

$$\begin{aligned} \frac{dC}{dt} &= r_1 C(1 - \frac{C + s_1 R + s_2 S}{K_1}) \\ \frac{dR}{dt} &= r_2 R(1 - \frac{s_3 C + R + s_4 S}{K_2}) \\ \frac{dS}{dt} &= r_3 S(1 - \frac{s_5 C + s_6 R + S}{K_3}). \end{aligned}$$

Here $r_3 > r_2 > r_1$, and we observe that the predation term we expect for the S equation has been subsumed into the expression

$$-r_3 S \frac{s_5 C}{K_3},$$

altering only the value of the parameter s_5. Since the C-cells produce both the toxin and the immunity protein we might expect them to use more environment than the R and S-cells (effectively, they are converting nutrients into toxins and proteins). I.e., $s_5 > s_6$, $s_3 > 1$, and $s_1 < 1$. Since the R-cells produce the immunity protein, and the S-cells do not, we might expect the R-cells to use more environment than the S-cells. I.e., $s_6 > 1$, $s_4 < 1$, and $s_1 > s_2$.

Also, note that it might not be unreasonable to assume the environment is the same for each cell, so that $K_1 = K_2 = K_3 = K$. (This assumes, for example, that the C-cells don't require a nutrient in addition to the nutrients required by the R and S-cells.) In this case, the numerator for each carrying capacity would be the same, in principle at least, because each cell would use up the same proportion of the shared carrying capacity.

2. Lagrangian and Hamiltonian mechanics are inappropriate because energy is lost to the surrounding air.

The force due to gravity on m acts vertically downward, and must be decomposed into a force $-T$, which is exactly balanced by the rod, and a force F, directed tangentially to the arc of motion (see Figure 5). Observing the right triangle, with hypotenuse of length $-mg$, we have

$$\cos\theta = \frac{T}{mg} \Rightarrow T = mg\cos\theta,$$
$$\sin\theta = -\frac{F}{mg} \Rightarrow F = -mg\sin\theta.$$

We know from our discussion of dimensional analysis that the force due to air resistance is proportional to velocity squared. Measuring distance as arclength, $d = l\theta$, we see that the pendulum's velocity along its arc is $v = l\frac{d\theta}{dt}$, so

$$F_{air} = -k\rho S l^2 \frac{d\theta}{dt}|\frac{d\theta}{dt}|,$$

where the absolute values are taken to get the direction correct. Newton's second law of motion ($F = ma$) becomes (noting $a = l\frac{d^2\theta}{dt^2}$)

$$ml\frac{d^2\theta}{dt^2} = -\,mg\sin\theta - k\rho S l^2 \frac{d\theta}{dt}|\frac{d\theta}{dt}|,$$

which can be written as

$$\frac{d^2\theta}{dt^2} = -\frac{g}{l}\sin\theta - \frac{k\rho S l}{m}\frac{d\theta}{dt}|\frac{d\theta}{dt}|.$$

It's customary to write

$$b = \frac{k\rho S l}{m}.$$

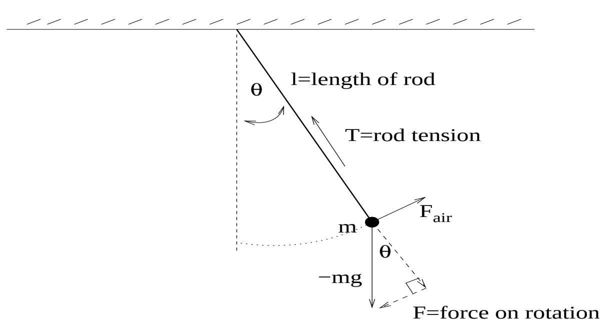

Figure 5: Pendulum motion under the influence of gravity alone.

3. In cartesian coordinates, the position of the mass is (measured from the center of the circle)

$$x = r\cos\omega t + l\sin\theta$$
$$y = -\,l\cos\theta + r\sin\omega t,$$

and so the kinetic energy of the mass is

$$
\begin{aligned}
K &= \frac{1}{2}m(\frac{dx}{dt}^2 + \frac{dy}{dt}^2) = \frac{1}{2}m\Big((-r\omega\sin\omega t + l\theta'\cos\theta)^2 + (l\theta'\sin\theta + r\omega\cos\omega t)^2\Big) \\
&= \frac{1}{2}m\Big(r^2\omega^2 + l^2\theta'^2 + 2lr\theta'\omega\sin(\theta - \omega t)\Big).
\end{aligned}
$$

If we take the bottom of the circle to be our reference (baseline) height for potential energy, we have

$$P = mg(r + y) = mg(r - l\cos\theta + r\sin\omega t).$$

In this case, there is only one generalized coordinate $q = \theta$. In terms of q, the Lagrangian is

$$L(q, q') = \frac{1}{2}m\Big(r^2\omega^2 + l^2q'^2 + 2lrq'\omega\sin(q - \omega t)\Big) - mg(r - l\cos q + r\sin\omega t).$$

The Euler-Lagrange equation in this case is

$$\frac{d}{dt}\frac{\partial L}{\partial q'} - \frac{\partial L}{\partial q} = 0,$$

and we have

$$
\begin{aligned}
\frac{\partial L}{\partial q} &= mlrq'\omega\cos(q - \omega t) - mgl\sin q \\
\frac{\partial L}{\partial q'} &= ml^2q' + mlr\omega\sin(q - \omega t).
\end{aligned}
$$

The Euler-Lagrange equation becomes

$$ml^2q'' + mlr\omega\cos(q - \omega t)(q' - \omega) = mlrq'\omega\cos(q - \omega t) - mgl\sin q,$$

which simplifies to

$$q'' - \frac{r\omega^2}{l}\cos(q - \omega t) + \frac{g}{l}\sin q = 0.$$

Clearly, if either r or ω is 0 we obtain the usual equation for an undamped pendulum.

4. In this case the kinetic energy is particularly easy,

$$K = \frac{1}{2}m_1x'^2 + \frac{1}{2}m_2y'^2,$$

and the potential energy is

$$P = \frac{1}{2}k_0x^2 + \frac{1}{2}k_1(y - x)^2 + \frac{1}{2}k_2y^2,$$

so the Lagrangian is

$$L = \frac{1}{2}m_1x'^2 + \frac{1}{2}m_2y'^2 - \frac{1}{2}k_0x^2 - \frac{1}{2}k_1(y - x)^2 - \frac{1}{2}k_2y^2.$$

In order to put this in the context of our notation from class, we'll use the generalized coordinates $(q_1, q_2) = (x, y)$, though this certainly isn't necessary. We have

$$L = \frac{1}{2}m_1 q_1'^2 + \frac{1}{2}m_2 q_2'^2 - \frac{1}{2}k_0 q_1^2 - \frac{1}{2}k_1(q_2 - q_1)^2 - \frac{1}{2}k_2 q_2^2.$$

The generalized momentum is

$$\vec{p} = D_{q'}L = (m_1 q_1', m_2 q_2')$$

(i.e., classical momentum in this case), and we easily solve for q_1' and q_2' in terms of p_1 and p_2 as

$$\begin{aligned} q_1' &= \frac{p_1}{m_1} \\ q_2' &= \frac{p_2}{m_2}. \end{aligned}$$

Note particularly that this provides us with equations for q_1 and q_2, so we only need to find equations for p_1 and p_2. For this, we write our Hamiltonian as

$$\begin{aligned} H &= \vec{p} \cdot \vec{q}' - L \\ &= \frac{p_1^2}{m_1} + \frac{p_2^2}{m_2} - \frac{1}{2}m_1\frac{p_1^2}{m_1^2} - \frac{1}{2}m_2\frac{p_2^2}{m_2^2} + \frac{1}{2}k_0 q_1^2 + \frac{1}{2}k_1(q_2 - q_1)^2 + \frac{1}{2}k_2 q_2^2 \\ &= \frac{1}{2}\frac{p_1^2}{m_1} + \frac{1}{2}\frac{p_2^2}{m_2} + \frac{1}{2}k_0 q_1^2 + \frac{1}{2}k_1(q_2 - q_1)^2 + \frac{1}{2}k_2 q_2^2. \end{aligned}$$

(Notice that we could have written this directly as total energy for the system, but it would have missed the point of the material we're reviewing.) Finally, $\vec{p}' = -D_q H$,

and

$$D_q H = (k_0 q_1 - k_1(q_2 - q_1), k_1(q_2 - q_1) + k_2 q_2).$$

This completes our system as

$$\begin{aligned} \frac{dq_1}{dt} &= \frac{p_1}{m_1} \\ \frac{dq_2}{dt} &= \frac{p_2}{m_2} \\ \frac{dp_1}{dt} &= -k_0 q_1 + k_1(q_2 - q_1) \\ \frac{dp_2}{dt} &= -k_1(q_2 - q_1) - k_2 q_2. \end{aligned}$$

6. Set

$$\tau = \frac{t}{A}; \quad Y_1(\tau) = \frac{y_1(t)}{C}; \quad Y_2(\tau) = \frac{y_2(t)}{D},$$

where A denotes a constant with dimension T, while C and D denote constants with dimension B (biomass). Our system becomes

$$
\begin{aligned}
\frac{dY_1}{d\tau} &= r_1 A Y_1 (1 - \frac{CY_1}{K_1}) + b_1 A D Y_1 Y_2 \\
\frac{dY_2}{d\tau} &= r_2 A Y_2 (1 - \frac{DY_2}{K_2}) + b_2 A C Y_1 Y_2.
\end{aligned}
$$

One natural choice is $C = K_1$, $D = K_2$, and $A = 1/r_1$. This gives

$$
\begin{aligned}
\frac{dY_1}{d\tau} &= Y_1(1 - Y_1) + \frac{b_1 K_2}{r_1} Y_1 Y_2 \\
\frac{dY_2}{d\tau} &= \frac{r_2}{r_1} Y_2 (1 - Y_2) + \frac{b_2 K_1}{r_1} Y_1 Y_2.
\end{aligned}
$$

Generally, this would be written as a three parameter system, for example with $\alpha = \frac{b_1 K_2}{r_1}$, $\beta = \frac{r_2}{r_1}$, and $\gamma = \frac{b_2 K_1}{r_1}$.

7. The matrix for this equation is

$$
A = \begin{pmatrix} 4 & 1 & 6 \\ -4 & 0 & -7 \\ 0 & 0 & -3 \end{pmatrix},
$$

which has eigenvalues $\lambda_1 = -3$ and $\lambda_2 = 2$, repeated. Setting $(A + 3I)\vec{v}_1 = 0$, we find that a choice of eigenvector for λ_1 is

$$
\vec{v}_1 = \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix}.
$$

Since $\lambda_2 = 2$ is repeated, we solve $(A - 2I)^2 \vec{v}_2 = 0$, for two generalized eigenvectors

$$
\vec{v}_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}; \quad \vec{v}_3 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.
$$

We can now write the initial vector $\vec{y}_0 = (1, 1, 1)^{tr}$ as

$$
\vec{y}_0 = c_1 \vec{v}_1 + c_2 \vec{v}_2 + c_3 \vec{v}_3,
$$

with

$$
\vec{c} = P^{-1} \vec{y}_0 = \begin{pmatrix} -1 \\ 2 \\ 0 \end{pmatrix}.
$$

(Recall $P = (\vec{v}_1, \vec{v}_2, \vec{v}_3)$.) Setting $\vec{x}_1 = c_1 \vec{v}_1$ and $\vec{x}_2 = c_2 \vec{v}_2 + c_3 \vec{v}_3$, we find

$$
\vec{x}_1 = \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix}; \quad \vec{x}_2 = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix}.
$$

The solution is

$$
\begin{aligned}
\vec{y}(t) &= e^{-3t} \begin{pmatrix} -1 \\ 1 \\ 1 \end{pmatrix} + e^{2t}[I + (A - 2I)t] \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} -e^{-3t} \\ e^{-3t} \\ e^{-3t} \end{pmatrix} + e^{2t} \begin{pmatrix} 1+2t & t & 6t \\ -4t & 1-2t & -7t \\ 0 & 0 & 1-5t \end{pmatrix} \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} \\
&= \begin{pmatrix} -e^{-3t} + 2(1+2t)e^{2t} \\ e^{-3t} - 8te^{2t} \\ e^{-3t} \end{pmatrix}.
\end{aligned}
$$

8. First, the equilibrium points are solutions of the system

$$
\begin{aligned}
\hat{y}_1(1 - \hat{y}_1 - a\hat{y}_2) &= 0 \\
b\hat{y}_2(1 - c\hat{y}_1 - \hat{y}_2) &= 0,
\end{aligned}
$$

and we find four

$$
\begin{pmatrix} 0 \\ 0 \end{pmatrix}; \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix}; \quad \begin{pmatrix} 1 \\ 0 \end{pmatrix}; \quad \frac{1}{1-ac} \begin{pmatrix} 1-a \\ 1-c \end{pmatrix}.
$$

Four the last of these, we are only interested in the case for which both populations are positive. In order to analyze stability, we write

$$
\vec{f}(\vec{y}) = \begin{pmatrix} y_1 - y_1^2 - ay_1y_2 \\ by_2 - cby_1y_2 - by_2^2 \end{pmatrix} \Rightarrow D\vec{f}(\vec{y}) = \begin{pmatrix} 1 - 2y_1 - ay_2 & -ay_1 \\ -cby_2 & b - cby_1 - 2by_2 \end{pmatrix}.
$$

For $(0,0)$,

$$
D\vec{f}(0,0) = \begin{pmatrix} 1 & 0 \\ 0 & b \end{pmatrix},
$$

with eigenvalues 1 and b. This equilibrium point is always unstable.

For $(0,1)$,

$$
D\vec{f}(0,1) = \begin{pmatrix} 1-a & 0 \\ -cb & -b \end{pmatrix},
$$

with eigenvalues $1-a$ and $-b$. This equilibrium point is unstable for $a < 1$ and asymptotically stable for $a > 1$. (The case $a = 1$ is undetermined by Poincare-Perron.)

For $(1,0)$,

$$
D\vec{f}(1,0) = \begin{pmatrix} -1 & -a \\ 0 & b(1-c) \end{pmatrix},
$$

with eigenvalues -1 and $b(1-c)$. This equilibrium point is unstable for $c < 1$ and asymptotically stable for $c > 1$. (The case $c = 1$ is undetermined by Poincare-Perron.)

For $(1-a, 1-c)/(1-ac)$,

$$
D\vec{f}\Big(\frac{1-a}{1-ac}, \frac{1-c}{1-ac}\Big) = \begin{pmatrix} -\hat{y}_1 & -a\hat{y}_1 \\ -cb\hat{y}_1 & -b\hat{y}_2 \end{pmatrix},
$$

where for notational brevity we're writing this equilibrium point as $(\hat{y}_1, \hat{y}_2)$. (Here, we have simplified by recalling that $1-\hat{y}_1-a\hat{y}_2=0$ and $1-c\hat{y}_1-\hat{y}_2=0$.) The eigenvalues for this matrix are

$$\lambda = \frac{-(\hat{y}_1+b\hat{y}_2) \pm \sqrt{(\hat{y}_1+b\hat{y}_2)^2 - 4b(1-ac)\hat{y}_1\hat{y}_2}}{2}.$$

If $1-ac<0$ the radical will dominate the numerator, and there will be a positive eigenvalue, giving instability. On the other hand, if $1-ac>0$ the numerator will certainly have negative real part and the equilibrium point is asymptotically stable.(The case $1-ac=1$ is undetermined by Poincare-Perron.)

9. As in our derivation of the 1-d wave equation in class, we'll apply Newton's second law (this time in the horizontal direction) to a strip of string between x and $x+\Delta x$. First,

$$ma \approx \rho \Delta x u_{tt}.$$

The force at position $x+\Delta x$ in the horizontal direction is

$$F_{x+\Delta x} \approx \rho(L-(x+\Delta x))g \sin\theta(x+\Delta x, t),$$

where we are assuming that for small displacement the string is almost straight (so we can ignore an arclength integral). Here, θ is the angle between the vertical and the tangent. At position x, the force downward is

$$F_x \approx \rho(L-x)g\sin\theta(x,t),$$

but this is interior to the part of the string we're working with, and we need to consider instead the opposite force caused by the string's being attached at the top (i.e., the force that keeps the string from falling). This is just the negative of F_x. In total,

$$F = F_{x+\Delta x} - F_x = \rho(L-(x+\Delta x))g\sin\theta(x+\Delta x,t) - \rho(L-x)g\sin\theta(x,t).$$

Newton's second law becomes

$$\rho\Delta x u_{tt} \approx \rho(L-(x+\Delta x))g\sin\theta(x+\Delta x,t) - \rho(L-x)g\sin\theta(x,t).$$

Upon dividing by Δx and taking $\Delta x \to 0$, we obtain

$$u_{tt} = g\Big((L-x)\sin\theta\Big)_x.$$

Finally, using the approximation from class,

$$\sin\theta \approx \tan\theta = u_x,$$

we obtain

$$u_{tt} = g\Big((L-x)u_x\Big)_x.$$

In this case, we require two initial conditions and two boundary conditions. For example, the initial conditions would generally by $u(x,0)=u_0(x)$ and $u_t(x,0)=w_0(x)$, and one of the boundary conditions would be $u(0,t)=0$ for $t \geq 0$, since the top location of the string is fixed. The second boundary condition is less clear, but the most natural one is probably $u_x(L,t) = 0$ for $t \geq 0$. This says that the bottom of the string be pointing straight downward.

10. Aside from some standard notation conventions, the calculation is the same for dimensions 2 and 3, so I'll write it for dimension 3. First, we can write

$$ma = \int_\omega \rho(\vec{x}) u_{tt}(\vec{x},t) dV.$$

Now the force acting on the solid's boundary is

$$F = \int_{\partial\omega} \kappa D_x u \cdot \hat{n} dS = \kappa \int_\omega \nabla \cdot D_x u dV = \kappa \int_\omega \Delta u dV.$$

Precisely as in our derivation of the continuity equation for $\mathbb{R}^3$, we conclude that since ω is arbitrary we must have

$$\rho(\vec{x}) u_{tt} = \kappa \Delta u.$$

For $\rho(\vec{x})$ constant, we have the wave equation with $c^2 = \kappa/\rho$.

Part Two Solutions

1. First, we express this equation in the linear form

$$\begin{aligned}
\frac{1}{y_1}\frac{dy_1}{dt} &= a - \frac{a}{K} y_1 - b y_2 \\
\frac{1}{y_2}\frac{dy_2}{dt} &= -r + c y_1.
\end{aligned}$$

For the first equation, we must fit $Z := \frac{1}{y_1}\frac{dy_1}{dt}$ as a function of two variables, $X = y_1$ and $Y = y_2$. (The first can be analyzed precisely as in class.) We use the MATLAB M-file *lvklinearfit1.m*.

```
%LVKLINEARFIT1: MATLAB script M-file to carry out linear parameter
%estimation for the Lotka-Volterra model with prey carrying capacity
%for the Hudson Bay data
%
lvdata;
%Prey equation
Z = (H(3:21)-H(1:19))./(2*H(2:20)); %Central difference derivative approximation
X = H(2:20);
Y = L(2:20);
M = [ones(size(Z))' X' Y'];
p = M\Z';
a = p(1)
K = -a/p(2)
```

```
b=-p(3)
%Predator equation
Y2 = (L(3:21)-L(1:19))./(2*L(2:20));
p2 = polyfit(X,Y2,1);
c = p2(1)
r = -p2(2)
%
%Plot the result
lvkrhs = @(t,y) [a*y(1)*(1-y(1)/K)-b*y(1)*y(2);-r*y(2)+c*y(1)*y(2)];
[t,y]=ode45(lvkrhs,[0,20],[30.0; 4.0]);
subplot(2,1,1)
plot(t,y(:,1),years,H,'o')
title('Prey population, model and data','FontSize',14)
subplot(2,1,2)
plot(t,y(:,2),years,L,'o')
title('Predator population, model and data','FontSize',14)
```

We find the following values:

$$\begin{aligned} a &= .4007 \\ b &= .0241 \\ c &= .0234 \\ r &= .7647 \\ K &= -184.8479. \end{aligned}$$

Clearly, the value for K is unphyical, though a plot of this model suggests our calculations were correct. (See Figure 6.)

For Part (b), we set $K := 77.40$. In this case, we use the linear form

$$\begin{aligned} \frac{1}{y_1}\frac{dy_1}{dt} &= a(1 - \frac{y_1}{K}) - by_2 \\ \frac{1}{y_2}\frac{dy_2}{dt} &= -r + cy_1. \end{aligned}$$

In this case, we consider $Z = \frac{1}{y_1}\frac{dy_1}{dt}$ to be a function of $X = (1 - \frac{y_1}{K})$ and $Y = y_2$. We carry out this fit with *lvklinearfit2.m*.

```
%LVKLINEARFIT2: MATLAB script M-file to carry out linear parameter
%estimation for the Lotka-Volterra model with prey carrying capacity
%for the Hudson Bay data. In this case, we set K to be the
%maximum observed prey population.
%
lvdata;
K = max(H)
```

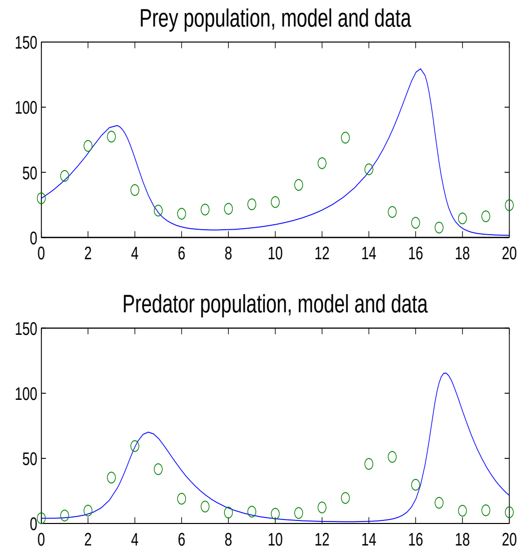

Figure 6: First linear fit.

```
%Prey equation
Z = (H(3:21)-H(1:19))./(2*H(2:20)); %Central difference derivative approximation
X = 1-H(2:20)/K;
Y = L(2:20);
M = [X' Y'];
p = M\Z';
a = p(1)
b = -p(2)
%Predator equation
Y2 = (L(3:21)-L(1:19))./(2*L(2:20));
X2 = H(2:20);
p2 = polyfit(X2,Y2,1);
c = p2(1)
r = -p2(2)
%
%Plot the result
lvkrhs = @(t,y) [a*y(1)*(1-y(1)/K)-b*y(1)*y(2);-r*y(2)+c*y(1)*y(2)];
[t,y]=ode45(lvkrhs,[0,20],[30.0; 4.0]);
subplot(2,1,1)
plot(t,y(:,1),years,H,'o')
title('Prey population, model and data','FontSize',14)
subplot(2,1,2)
plot(t,y(:,2),years,L,'o')
title('Predator population, model and data','FontSize',14)
```

In this case, we find

$$\begin{aligned}
a &= .4302 \\
b &= .0169 \\
c &= .0234 \\
r &= .7646 \\
K &= 77.4.
\end{aligned}$$

The solution using these values is plotted against the data in Figure 7.

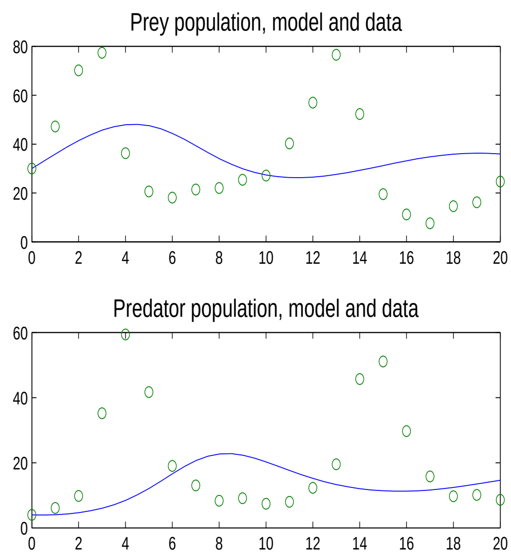

Figure 7: Second linear fit.

For Part (c), the nonlinear fit is carried out in *lvknonlinearfit.m*.

```
function lvknonlinearfit
%LVKNONLINEARFIT: MATLAB function M-file that takes an initial
%approximation of parameter values and carries out nonlinear
%regression to obtain best-fit parameter values for the Lotka-Volterra
%system with prey carrying capacity for the Hudson Bay data.
global years L H;
lvdata
guess = [.4302; .0169; .0234; .7646; max(H)]; %Order: a, b, c, r, K
options = optimset('MaxFunEvals',1e4);
[p,error]=fminsearch(@lverr, guess, options);
a = p(1)
b = p(2)
c = p(3)
r = p(4)
```

```
K = p(5)
s = sqrt(error/(2*(length(H)-1)-length(p)))
%
[t,y]=ode45(@lvrhs,[0,20],[H(1); L(1)],[],p);
subplot(2,1,1)
plot(t,y(:,1),years,H,'o')
title('Prey population, model and data','FontSize',14)
subplot(2,1,2)
plot(t,y(:,2),years,L,'o')
title('Predator population, model and data','FontSize',14)
%
function error = lverr(p)
%LVERR: Function defining error function for
%example with Lotka-Volterra equations.
global years L H;
[t,y] = ode45(@lvrhs,years,[H(1);L(1)],[],p); %Notice that we pass
%a parameter vector
error = norm(y(:,1)-H')^2+norm(y(:,2)-L')^2;
%
function value = lvrhs(t,y,p)
%LVRHS: ODE for example Lotka-Volterra paramter
%estimation example. p(1)=a, p(2) = b, p(3) = c, p(4) = r, p(5)=K
value=[p(1)*y(1)*(1-y(1)/p(5))-p(2)*y(1)*y(2);-p(4)*y(2)+p(3)*y(1)*y(2)];
```

We find values

$$\begin{aligned} a &= .6210 \\ b &= .0288 \\ c &= .0245 \\ r &= .7650 \\ K &= 724.0122, \end{aligned}$$

which give a reasonable fit to the data (see Figure 8), but we find $s = 4.8249$, which is worse than the value we obtained without a carrying capacity. This still suggests we are not justified in including a carrying capacity with this model.

For Part (d) we proceed in our usual way to find confidence intervals with *lvkconf.m*.

```
function lvkconf
%LVKCONF: MATLAB function M-file for computing 95% confidence
%intervals for parameter estimates of the Lotka-Volterra
%model fit to the Hudson Bay data.
%Parameter values
lvdata;
y0 = [30.0;4.0];
```

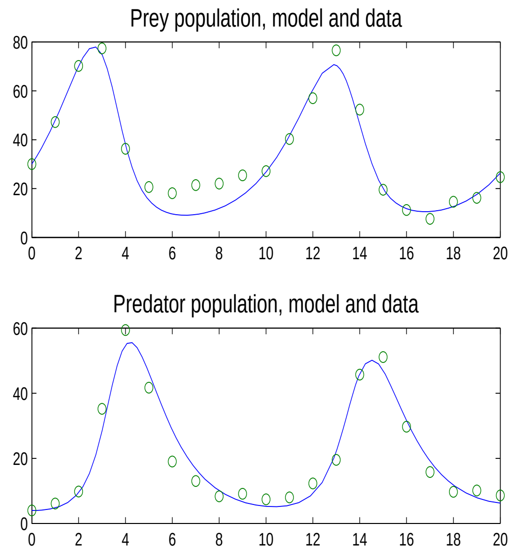

Figure 8: Nonlinear fit.

```
pbar = [.6210 .0288 .0245 .7650 724.0122];
[t ybar]=ode45(@lvrhs,years,y0,[],pbar);
dely = [H'; L'] - [ybar(:,1); ybar(:,2)];
%Set increment for derivative approximations
h = 1e-6;
[t ybar1]=ode45(@lvrhs,years,y0,[],[pbar(1)+h pbar(2) pbar(3) pbar(4) pbar(5)]);
[t ybar2]=ode45(@lvrhs,years,y0,[],[pbar(1) pbar(2)+h pbar(3) pbar(4) pbar(5)]);
[t ybar3]=ode45(@lvrhs,years,y0,[],[pbar(1) pbar(2) pbar(3)+h pbar(4) pbar(5)]);
[t ybar4]=ode45(@lvrhs,years,y0,[],[pbar(1) pbar(2) pbar(3) pbar(4)+h pbar(5)]);
[t ybar5]=ode45(@lvrhs,years,y0,[],[pbar(1) pbar(2) pbar(3) pbar(4) pbar(5)+h]);
F = [ybar1(:,1)-ybar(:,1) ybar2(:,1)-ybar(:,1) ybar3(:,1)-ybar(:,1) ybar4(:,1)-
ybar(:,1) ybar5(:,1)-ybar(:,1); ...
ybar1(:,2)-ybar(:,2) ybar2(:,2)-ybar(:,2) ybar3(:,2)-ybar(:,2) ybar4(:,2)-ybar(:,2)
ybar5(:,2)-ybar(:,2)]/h;
delp = F\dely
pnew = pbar'+delp
%For 95% confidence interval for q=2(N-1)-5=35
l = 2.0301;
V = inv(F'*F);
ssq = (norm(dely-F*delp)^2)/35;
s = sqrt(ssq)
error = l*sqrt(ssq*diag(V))
%
function yprime = lvrhs(t,y,p)
a = p(1); b = p(2); c = p(3); r = p(4); K=p(5);
yprime = [a*y(1)*(1-y(1)/K)-b*y(1)*y(2); -r*y(2)+c*y(1)*y(2)];
```

In this case, let's look at a diary file of the output:

```
>>lvkconf
delp =
-0.0482
-0.0001
0.0011
0.0418
474.5652
pnew =
1.0e+03 *
0.0006
0.0000
0.0000
0.0008
1.1986
s =
4.6523
error =
0.0889
0.0034
0.0029
0.0980
605.6802
```

We notice that the last component of Δp is not small, and this suggests these are not reasonable parameter values. (I.e., the linearization is not justified.) In particular, looking at *pnew*, we see that the suggested value for K is

$$K = 1198.6 \pm 605.6902.$$

Notice that the larger K is, the closer this becomes to the case in which carrying capacity is omitted. Again, we conclude that the model with carrying capacity is not justified for this data.

2. First, we obtain a linear form for this PDE by expressing it as

$$u_t = aT_0 - au + \mu u_{xx}.$$

The linear approximation, using both the *leapfrog* method and the Crank-Nicolson method, is carried out with *heatlinear1.m*. Notice that this file is more efficient than the file *fisher-linear1.m* from class, though problably not quite as clear.

```
%HEATLINEAR1: Data and linear fit for the
%Fisher-type PDE parameter estimation.
clear X Y Z;
```

```
load heatdata;
h=.5; k=.2;
dex=1;
%The first row of u corresponds with t=0, the second with t=.2 etc. It's
%consistent with MATLAB's pdepe syntax to take the first index to be
%associated with t.
X = reshape(udata(2:10,2:5),[],1);
Y = reshape((udata(2:10,3:6)-2*udata(2:10,2:5)+udata(2:10,1:4))/k^2,[],1);
Z = reshape((udata(3:11,2:5)-udata(1:9,2:5))/(2*h),[],1);
M = [ones(size(Z)) X Y];
p = M\Z;
a = -p(2)
T0 = p(1)/a
mu = p(3)
%
%Revised values consistent with the Crank-Nicolson finite difference scheme
X = reshape((udata(2:10,2:5)+udata(3:11,2:5))/2,[],1);
Y = reshape(((udata(2:10,3:6)-2*udata(2:10,2:5)+udata(2:10,1:4))/k^2+(udata(3:11,3:6)-
2*udata(3:11,2:5)+udata(3:11,1:4))/k^2)/2,[],1);
Z = reshape((udata(3:11,2:5)-udata(2:10,2:5))/(h),[],1);
M = [ones(size(Z)) X Y];
p = M\Z;
a = -p(2)
T0 = p(1)/a
mu = p(3)
```

Using the Crank-Nicolson values, we obtain

$$
\begin{aligned}
a &= .4232 \\
T_0 &= 34.9975 \\
\mu &= .0101.
\end{aligned}
$$

We now carry out the nonlinear fit with *heatnonlinear1.m*. (This runs in about 30 seconds.)

```
function heatnonlinear1
%HEATNONLINEAR1: MATLAB function M-file that
%estimates parameter values for a PDE version of
%the logistic population model based on
%numerically generated data
%
guess = [.4232;34.9975;.0101]; %order: a, T0, mu
[p,E]=fminsearch(@error, guess)
s = sqrt(E/117) %For this problem, there are 117
%effective degrees of freedom.
%
```

```
function err = error(p)
load heatdata;
m=0; %Recall that m is part of the PDE specification in MATLAB
tvals=0:.5:10; %These correspond with data; MATLAB uses an internal
%discretization
xvals=0:.01:1; %MATLAB uses this mesh, so we must evaluate u at data
point
%after the calculation
%Now evolve with in time
u = pdepe(m,@eqn,@initial,@bc,xvals,tvals,[],p);
err = norm(reshape(u(:,1:20:101)-udata,[],1))^2;
%
function [c,b,s] = eqn(x,t,u,DuDx,p)
%Define the PDE
a = p(1); T0=p(2); mu=p(3);
c=1; b=mu*DuDx; s=-a*(u-T0);
%
function [pl,ql,pr,qr]=bc(xl,ul,xr,ur,t,p)
%Define boundary data
pl=0; ql=1; pr=0; qr=1;
%
function value = initial(x,p)
value = 50*exp(-5*(x-.5).^2);
```

We find

$$\begin{aligned} a &= .0577 \\ T_0 &= 34.9064 \\ \mu &= .0186, \end{aligned}$$

with approximate standard deviation $s = .0704$. We plot this approximation along with the data with *heatcomp1.m*.

```
function heatcomp1
%HEATCOMP1: MATLAB function M-file that solves a
%heat equation with Newton-type source
%and compares the results with data.
m=0;
tvals=0:.5:10;
xvals=0:.01:1;
%Now evolve with in time
load heatdata;
u = pdepe(m,@eqn,@initial,@bc,xvals,tvals);
fig1=plot(xvals,u(1,:),'erase','xor');
title(['Temperature u as a function of x, Time = ', num2str(0), 'years'],'fontsize',14);
```

```
hold on
fig2=plot(0:.2:1,udata(1,:),'ro','erase','xor');
legend('Approximate Solution','Data','location','NorthWest')
axis([0 1 min(reshape(u,[],1)) max(reshape(u,[],1))]);
pause
for k=2:length(tvals)
set(fig1,'xdata',xvals,'ydata',u(k,:));
set(fig2,'xdata',0:.2:1,'ydata',udata(k,:));
title(['Temperature u as a function of x, Time = ',num2str(tvals(k)),'years'],'fontsize',14)
pause(.2)
end
%
%Files for solving the PDE
%
function [c,b,s] = eqn(x,t,u,DuDx)
%Define the PDE
a=.0577; T0 = 35.9064; mu=.0186;
c=1; b=mu*DuDx; s=-a*(u-T0);
%
function [pl,ql,pr,qr]=bc(xl,ul,xr,ur,t)
%Define boundary data
pl=0; ql=1; pr=0; qr=1;
%
function value = initial(x)
value = 50*exp(-5*(x-.5).^2);
```

The solution and data are plotted for $t = 10$, by which time x dependence has mostly been lost.

Finally, we compute the confidence intervals with *heatconf1.m.*

```
function heatconf1
%HEATCONF1: MATLAB function M-file that computes
%95% confidence intervals for parameters for the
%heat equation example.
load heatdata;
pbar = [.0577 34.9064 .0186];
%
m=0; %Recall that m is part of the PDE specification in MATLAB
tvals=0:.5:10; %These correspond with data; MATLAB uses an internal
%discretization
xvals=0:.01:1; %MATLAB uses this mesh, so we must evaluate u at data
points
%after the calculation
ubar = pdepe(m,@eqn,@initial,@bc,xvals,tvals,[],pbar);
```

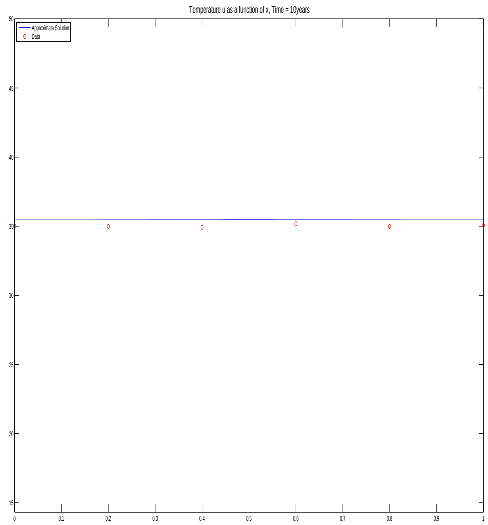

Figure 9: Model and data for the heat equation with Newton source.

```
delu = reshape(udata-ubar(:,1:20:101),[],1);
%
%Set increment for derivative approximations
h = 1e-6;
ubar1 = pdepe(m,@eqn,@initial,@bc,xvals,tvals,[],pbar+[h 0 0]);
ubar2 = pdepe(m,@eqn,@initial,@bc,xvals,tvals,[],pbar+[0 h 0]);
ubar3 = pdepe(m,@eqn,@initial,@bc,xvals,tvals,[],pbar+[0 0 h]);
F = [reshape(ubar1(:,1:20:101)-ubar(:,1:20:101),[],1), ...
reshape(ubar2(:,1:20:101)-ubar(:,1:20:101),[],1), ...
reshape(ubar3(:,1:20:101)-ubar(:,1:20:101),[],1)]/h;
delp = F\delu
pnew = pbar'+delp
%For 95% confidence interval, q=20 x 6 - 3 = 117, so l = 1.9804
l=1.9804;
V = inv(F'*F);
ssq = (norm(delu-F*delp)^2)/117;
s = sqrt(ssq)
error = l*sqrt(ssq*diag(V))
%
function [c,b,s] = eqn(x,t,u,DuDx,p)
%Define the PDE
a = p(1); T0=p(2); mu=p(3);
c=1; b=mu*DuDx; s=-a*(u-T0);
%
function [pl,ql,pr,qr]=bc(xl,ul,xr,ur,t,p)
%Define boundary data
```

```
pl=0; ql=1; pr=0; qr=1;
%
function value = initial(x,p)
value = 50*exp(-5*(x-.5).^2);
```

In this case, we give a diary session of the implementation.

```
>>heatconf1
delp =
-0.0044
-0.0133
0.0001
pnew =
0.0533
34.8931
0.0187
s =
0.0704
error =
0.0932
0.2799
0.0023
```

We conclude

$$\begin{aligned} a &= .0533 \pm .0932 \\ T_0 &= 34.8931 \pm .2799 \\ \mu &= .0187 \pm .0023. \end{aligned}$$