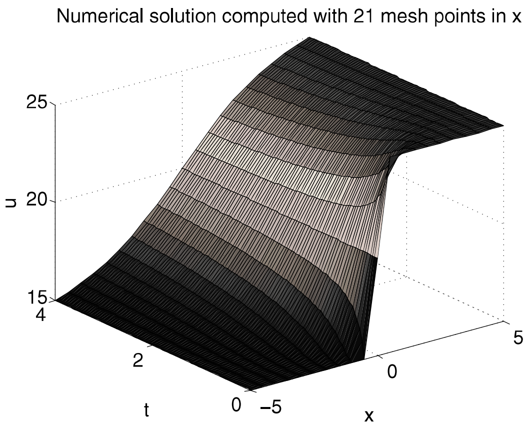
```
% A surface plot is often a good way to study a solution.
surf(x, t, u)
title('Numerical solution computed with 21 mesh points in x')
xlabel('x'), ylabel('t'), zlabel('u')
% ----------------------------------------------------------
function [c, f, s] = pdex(x, t, u, DuDx)
c = 1;
f = (1 + (x/5).^2)*DuDx;  % flux = conductivity times u_x
s = 0;
% ----------------------------------------------------------
function u0 = pdexic(x)
u0 = 20 + 5*sign(x);  % initial condition at t = 0
% ----------------------------------------------------------
function [pl, ql, pr, qr] = pdexbc(xl, ul, xr, ur, t)
% q's are zero since we have Dirichlet conditions
% pl = 0 at the left, pr = 0 at the right endpoint
pl = ul - 15;
ql = 0;
pr = ur - 25;
qr = 0;
```

Running it gives:

```
heateqex2
```



Again the results are very similar to those obtained before.

✰ A Model of Traffic Flow

Everyone has had the experience of sitting in a traffic jam, or of seeing cars bunch up on a road for no apparent good reason. MATLAB and Simulink are good tools for studying models of such behavior. Our analysis here will be based on so-called "follow-the-leader" theories of traffic flow, about which you can read more in **Kinetic Theory of Vehicular Traffic**, by Ilya Prigogine and Robert Herman, Elsevier, New

York, 1971, or in **The Theory of Road Traffic Flow**, by Winifred Ashton, Methuen, London, 1966. We will analyze here an extremely simple model that already exhibits quite complicated behavior. We consider a one-lane, one-way, circular road with a number of cars on it (a very primitive model of, say, the Outer Loop of the Capital Beltway around Washington, DC, since, in very dense traffic, it is hard to change lanes and each lane behaves like a one-lane road). Each driver slows down or speeds up on the basis of his own speed, the speed of the car ahead of him, and the distance to the car ahead of him. But human drivers have a finite reaction time. In other words, it takes them a certain amount of time (usually about a second) to observe what is going on around them and to press the gas pedal or the brake, as appropriate. The standard "follow-the-leader" theory supposes that

$$\ddot{u}_n(t+T) = \lambda\bigl(\dot{u}_{n-1}(t) - \dot{u}_n(t)\bigr), \qquad (\dagger)$$

where t is time, T is the reaction time, u_n is the position of the nth car, and the "sensitivity coefficient" λ may depend on $u_{n-1}(t) - u_n(t)$, the spacing between cars, and/or $\dot{u}_n(t)$, the speed of the nth car. The idea behind this equation is this. A driver will tend to decelerate if he is going faster than the car in front of him, or if he is close to the car in front of him, and will tend to accelerate if he is going slower than the car in front of him. In addition, a driver (especially in light traffic) may tend to speed up or slow down depending on whether he is going slower or faster (respectively) than a "reasonable" speed for the road (often, but not always, equal to the posted speed limit). Since our road is circular, in this equation u_0 is interpreted as u_N, where N is the total number of cars.

The simplest version of the model is the one in which the "sensitivity coefficient" λ is a (positive) constant. Then we have a homogeneous linear differential/difference equation with constant coefficients for the velocities $\dot{u}_n(t)$. Obviously there is a "steady-state" solution when all the velocities are equal and constant (i.e., traffic is flowing at a uniform speed), but what we are interested in is the stability of the flow, or the question of what effect is produced by small differences in the velocities of the cars. The solution of (†) will be a superposition of exponential solutions of the form

$$u_n(t) = \exp(\alpha t) v_n,$$

where the v_n's and α are (complex) constants, and the system will be unstable if the velocities are unbounded, i.e., there are any solutions where the real part of α is positive. Using vector notation, we have

$$\dot{\mathbf{u}}(t) = \exp(\alpha t)\mathbf{v}, \qquad \ddot{\mathbf{u}}(t+T) = \alpha \exp(\alpha T)\exp(\alpha t)\mathbf{v}.$$

Substituting back into (†), we get the equation

$$\alpha \exp(\alpha T) \exp(\alpha t)\mathbf{v} = \lambda(S - I) \exp(\alpha t)\mathbf{v},$$

where

$$S = \begin{pmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}$$

is the "shift" matrix that, when it multiplies a vector on the left, cyclically permutes the entries of the vector. We can cancel the $\exp(\alpha t)$ on each side to get

$$\alpha \exp(\alpha T)\mathbf{v} = \lambda(S - I)\mathbf{v},$$

or

$$\left(S - \left(1 + \frac{\alpha}{\lambda} \exp(\alpha T)\right) I\right) \mathbf{v} = 0, \qquad (**)$$

which says that v is an eigenvector for S with eigenvalue

$$1 + \frac{\alpha}{\lambda} \exp(\alpha T).$$

Since the eigenvalues of S are the Nth roots of unity, which are evenly spaced around the unit circle in the complex plane, and closely spaced together for large N, there is potential instability whenever

$$1 + \frac{\alpha}{\lambda} \exp(\alpha T)$$

has absolute value 1 for some α with positive real part; that is, whenever

$$\left(\frac{\alpha T}{\lambda T}\right) e^{\alpha T}$$

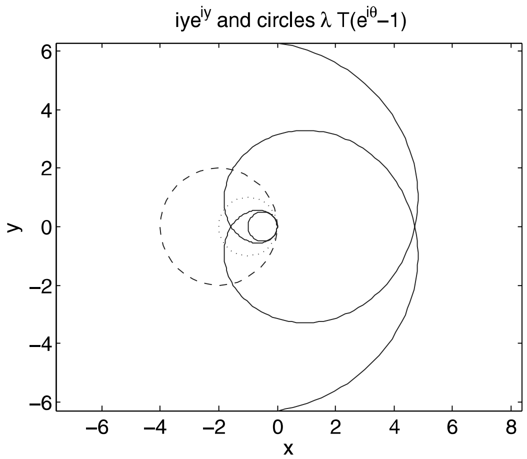
can be of the form $\exp(i\theta) - 1$ for some αT with positive real part. Whether instability occurs depends on the value of the product λT. We can see this by plotting values of $z\exp(z)$ for $z = \alpha T = iy$ a complex number on the critical line Re $z = 0$, and comparing with plots of $\lambda T(e^{i\theta} - 1)$ for various values of the parameter λT.

```
syms y; expand(i*y*(cos(y) + i*sin(y)))
```

```
ans =

i*y*cos(y)-y*sin(y)
```

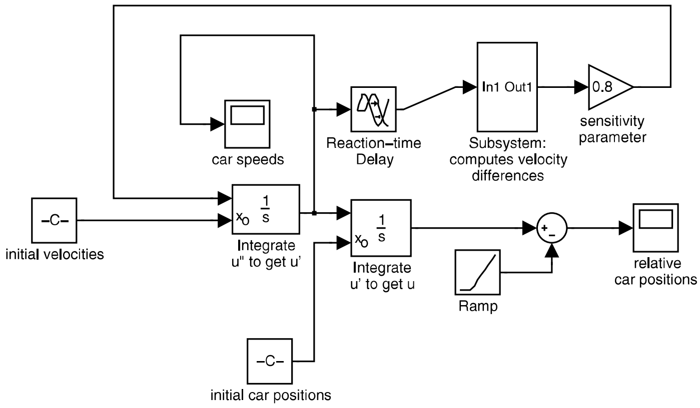
```
ezplot(-y*sin(y), y*cos(y), [-2*pi, 2*pi]); hold on
theta = 0:0.05*pi:2*pi;
plot((1/2)*(cos(theta) - 1), (1/2)*sin(theta), '-');
plot(cos(theta) - 1, sin(theta), ':')
plot(2*(cos(theta) - 1), 2*sin(theta), '--');
title('iye^{iy} and circles \lambda T(e^{i\theta}-1)')
hold off
```



Here the small solid circle corresponds to $\lambda T = 1/2$, and we are just at the limit of stability, since this circle does not cross the spiral produced by $z\exp(z)$ for z a complex number on the critical line Re $z = 0$, though it "hugs" the spiral closely. The dotted and dashed circles, corresponding to $\lambda T = 1$ or 2, do cross the spiral, so they correspond to unstable traffic flow.

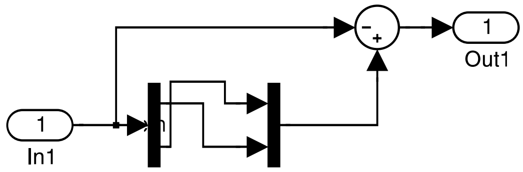
We can check these theoretical predictions with a simulation using Simulink. We'll give a picture of the Simulink model and then explain it.

```
open_system traffic
```



Here the Subsystem, which corresponds to multiplication by $S - I$, looks like this:

```
open_system 'traffic/Subsystem: computes velocity differences'
```



Most of the model is like the example in Chapter 8, except that our unknown function (called u), representing the car positions, is vector-valued, not scalar-valued. The major exceptions are these.

- We need to incorporate the reaction-time delay, so we've inserted a **Transport Delay** block from the **Continuous** block library, with the "Time delay" parameter T set to 0.5.
- The parameter λ shows up as the value of the gain in the "sensitivity parameter" **Gain** block in the upper right.
- Plotting car positions by themselves is not terribly useful, since only the relative positions matter. So before outputting the car positions to the **Scope** block labeled "relative car positions," we've subtracted a constant linear function (corresponding to uniform motion at the average car speed) created by the **Ramp** block from the **Sources** block library.
- We've made use of the option in the **Integrator** blocks to input the initial conditions, instead of having them built into the block. This makes the logical structure a little clearer.
- We've used the subsystem feature of Simulink. If you enclose a bunch of blocks with the mouse and then click on "Create subsystem" in the model's **Edit** menu, Simulink will package them as a subsystem. This is helpful if your model is large or if there is some combination of blocks that you expect to use more than once. Our subsystem sends a vector v to $(S - I)v = Sv - v$. A **Sum** block (with one of the signs changed to a minus) is used for vector subtraction.

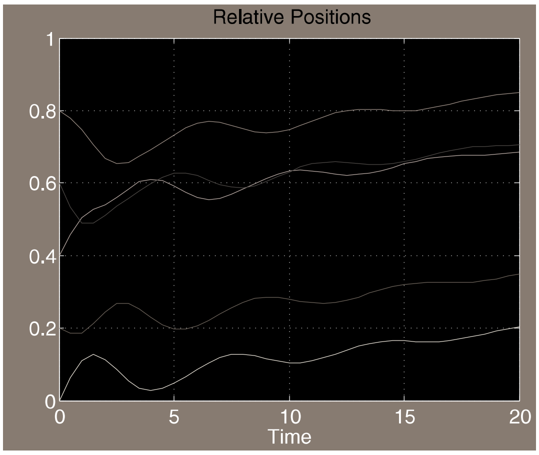
> To model the action of S, we've used the **Demux** and **Mux** blocks from the **Signal Routing** block library. The **Demux** block, with the "number of outputs" parameter set to `[4, 1]`, splits a five-dimensional vector into a pair consisting of a four-dimensional vector and a scalar (corresponding to the last car). Then we reverse the order of these and put them back together with the **Mux** block, with the "number of inputs" parameter set to `[1, 4]`.

Once the model has been assembled, it can be run with various inputs. You can see the results yourself in the two **Scope** windows, but here we've run the simulation from the command line and plotted the results with the `simplot` command, that does almost the same thing as a **Scope** but in a regular MATLAB figure window. The following pictures are produced with $\lambda = 0.8$, corresponding to stable flow (though, to be honest, we've let two cars cross through each other briefly!):

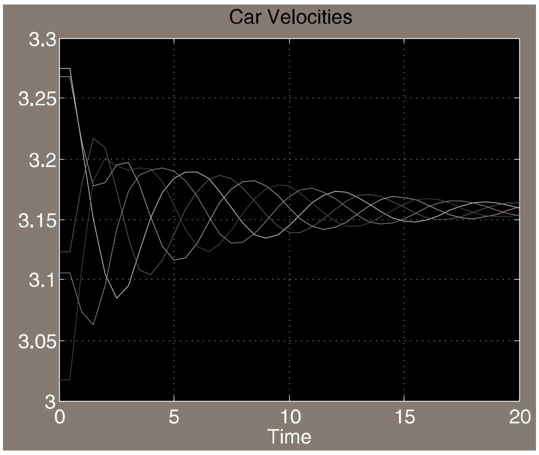
```
set_param('traffic/sensitivity parameter', 'Gain', '0.8');
[t, x] = sim('traffic');
```

The variable `t` stores the time parameter, the variable `x` stores car velocities in its first five columns and car positions in the second five columns. In this example, the average velocity is 3.15. First we plot the relative positions, then we plot the velocities.

```
relpos = x(:,6:10) - 3.15*t*ones(1,5);
simplot(t, relpos), title('Relative Positions')
axis([0, 20, 0, 1]), axis normal
```

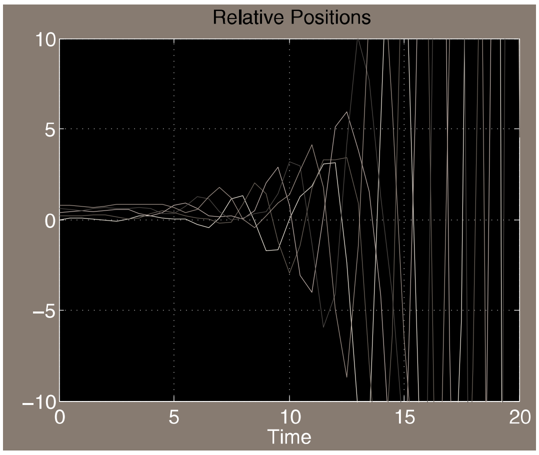



```
simplot(t, x(:,1:5)), title('Car Velocities')
axis([0, 20, 3, 3.3])
```

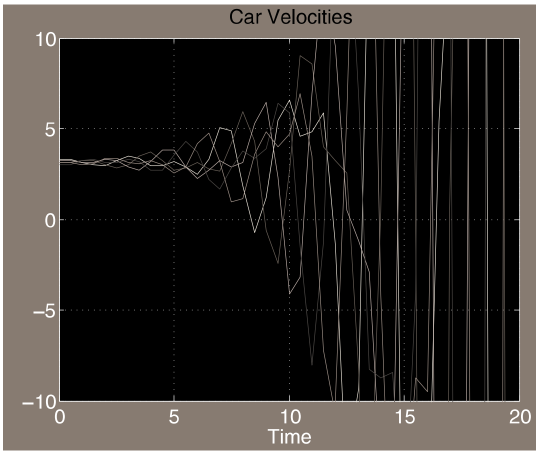


As you can see, the speeds fluctuate but eventually converge to a single value, and the separations between cars eventually stabilize. On the other hand, if λ is increased by changing the "sensitivity parameter" in the **Gain** block in the upper right, say from 0.8 to 2.0, we get the following output, which is typical of instability:

```
set_param('traffic/sensitivity parameter', 'Gain', '2.0');
[t, x] = sim('traffic');
relpos = x(:,6:10) - 3.15*t*ones(1,5);
simplot(t, relpos), title('Relative Positions')
axis([0, 20, -10, 10])
```



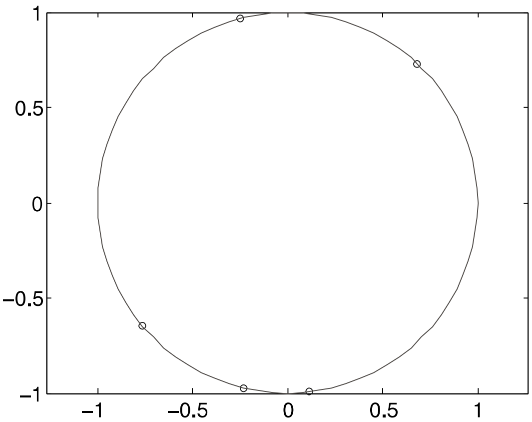

```
simplot(t, x(:,1:5)), title('Car Velocities')
axis([0, 20, -10, 10])
```



We encourage you to go back and tinker with the model (for instance using a sensitivity parameter that is also inversely proportional to the spacing between cars) and study the results.

Finally, you can create a movie with the following code:

```
clf reset
set(gcf, 'Color', 'White')
clear M
theta = (0:0.025:2)*pi;
for j = 1:length(t)
    plot(cos(x(j, 1:5)), sin(x(j, 1:5)), 'o');
    axis([-1, 1, -1, 1]);
    hold on; plot(cos(theta), sin(theta), 'r'); hold off
    axis equal;
    M(j) = getframe;
end
```



The idea here is that we have taken the circular road to have radius 1 (in suitable units), so that the command **plot(cos(theta),sin(theta),'r')** draws a red circle (representing the road) in each frame of the movie, and on top of that the cars are shown with moving little circles. The graph above is the last frame of the movie; you can view the entire movie by typing **`movie(M)`** or **`movieview(M)`**. Try it!

We should mention here one fine point needed to create a realistic movie. Namely, we need the values of **`t`** to be equally spaced – otherwise the cars will appear to be moving faster when the time steps are large, and will appear to be moving slower when the time steps are small. In its default mode of operation, Simulink uses a variable-step differential-equation solver based on MATLAB's command **`ode45`**, so the entries of **`t`** will not be equally spaced. We have fixed this by opening the **Configuration Parameters** dialog box using the **Simulation** menu in the model window, and, under the **Data Import/Export** item, changing the **Output options** box to read `Produce specified output only`, with `Output times` chosen to be **`[0:0.5:20]`**. Then the model will output the car positions only at times that are multiples of 0.5, and the MATLAB program above will produce a 41-frame movie.

Practice Set C

Developing Your MATLAB Skills

Problems 5, 7, 14, and 15 are a bit more advanced than the others. Problems 8 and 9 require either **simlp** from Simulink or else the Optimization Toolbox. Problem 11(a) requires the Symbolic Math Toolbox; the others do not. Simulink is needed for Problems 12 and 13.

1. Captain Picard is hiding in a square arena, 50 meters on a side, which is protected by a level-5 force field. Unfortunately, the Cardassians, who are firing on the arena, have a death ray that can penetrate the force field. The point of impact of the death ray is exposed to 10,000 *illumatons* of lethal radiation. It requires only 50 illumatons to dispatch the Captain; anything less has no effect. The number of illumatons that arrive at point (x, y) when the death ray strikes one meter above ground at point (x_0, y_0) is governed by an inverse square law, namely
$$\frac{10000}{4\pi((x - x_0)^2 + (y - y_0)^2 + 1)}.$$
The Cardassian sensors cannot locate Picard's exact position, so they fire at a random point in the arena.

   (a) Use **contour** to display the arena after five random bursts of the death ray. The half-life of the radiation is very short, so one can assume that it disappears almost immediately – only its initial burst has any effect. Nevertheless, include all five bursts in your picture, like a time-lapse photo. Where in the arena do you think Captain Picard should hide?

   (b) Suppose that Picard stands in the center of the arena. Moreover, suppose that the Cardassians fire the death ray 100 times, each shot landing at a random point in the arena. Is Picard killed?

   (c) Re-run the "experiment" in part (b) 100 times, and approximate the probability that Captain Picard can survive an attack of 100 shots.

   (d) Redo part (c) but place the Captain halfway to one side (i.e., at $x = 37.5$, $y = 25$ if the coordinates of the arena are $0 \le x \le 50, 0 \le y \le 50$).

   (e) Redo the simulation with the Captain completely to one side, and finally in a corner. What self-evident fact is reinforced for you?

2. Consider an account that has M dollars in it and pays monthly interest J. Suppose that, beginning at a certain point, an amount S is deposited monthly and no withdrawals are made.

   (a) Assume first that $S = 0$. Using the *Mortgage Payments* application in Chapter 10 as a model, derive an equation relating J, M, the number n of months elapsed, and the total T in the account after n months. Assume that the interest is credited on the last day of the month and the total T is computed on the last day after the interest is credited.

   (b) Now assume that $M = 0$, that S is deposited on the first day of the month, and that as before interest is credited on the last day of the month, and the total T is computed on the last day after the interest has been credited. Once again, using the mortgage application as a model, derive an equation relating J, S, the number n of months elapsed, and the total T in the account after n months.

   (c) By combining the last two models derive an equation relating all of M, S, J, n, and T, now of course assuming that there is an initial amount in the account (M) as well as a monthly deposit (S).

   (d) If the annual interest rate is 5%, and no monthly deposits are made, how many years does it take to double your initial stash of money? What if the annual interest rate is 10%?

   (e) In this and the next part, there is no initial stash. Assume an annual interest rate of 8%. How much do you have to deposit monthly to be a millionaire in 35 years (a career)?

   (f) If the interest rate remains as in (e) and you can afford to deposit only $300 each month, how long do you have to work to retire a millionaire?

   (g) You hit the lottery and win $100,000. You have two choices: take the money, pay the taxes, and invest what's left; or receive $100,000/240 monthly for 20 years, depositing what's left after taxes. Assume that a $100,000 windfall costs you $35,000 in federal and state taxes, but that the smaller monthly payoff causes only a 20% tax liability. In which way are you better off 20 years later? Assume a 5% annual interest rate here.

   (h) Historically, banks have paid roughly 5%, while the stock market has tended to return 8% on average over a 10-year period. So parts (e) and (f) relate more to investing than to saving. But suppose that the market in a 5-year period returns 13%, 15%, -3%, 5% and 10% in five successive years, and then repeats the cycle. (Note that the [arithmetic] average is 8%, though a geometric mean would be more relevant here.) Assume that $50,000 is invested at the start of a 5-year market period. How much does it grow to in 5 years? Now recompute four more times, assuming that you enter the cycle at the beginning of the second year, the third year, etc. Which choice yields the best/worst results? Can you explain why? Compare the results with a fixed-rate account paying 8%. Assume simple annual interest. Redo the five investment computations, assuming that $10,000 is invested at the start of each year. Again analyze the results.

3. Tony Gwynn had a lifetime batting average of .338. This means that, for every 1000 at bats, he had 338 hits. (For this exercise, we shall ignore walks, hit batsmen, sacrifices, and other plate appearances that do not result in an official at bat.) In an average year he amassed 500 official at bats.

   (a) Design a Monte Carlo simulation of a year in Tony's career. Run it. What is his batting average?

   (b) Now simulate a 20-year career. Assume 500 official at bats every year. What is his best batting average in his career? What is his worst? What is his lifetime average?

   (c) Now run the 20-year career simulation four more times. Answer the questions in part (b) for each of the four simulations.

   (d) Compute the average of the five lifetime averages you computed in parts (b) and (c). What do you think would happen if you ran the 20-year simulation 100 times and took the average of the lifetime averages for all 100 simulations?

The next four problems illustrate some basic MATLAB programming skills.

4. For a positive integer n, let $A(n)$ be the $n \times n$ matrix whose entry in the (i, j)-position is $a_{ij} = 1/(i + j - 1)$. For example,

$$A(3) = \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ \frac{1}{2} & \frac{1}{3} & \frac{1}{4} \\ \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \end{pmatrix}.$$

   The eigenvalues of $A(n)$ are all real numbers. Write a script M-file that prints the largest eigenvalue of $A(500)$, without any extraneous output. (*Hint*: the M-file may take a while to run if you use a loop within a loop to define A. Try to avoid this!)

5. ✰ Write a script M-file that draws a bulls-eye pattern with a central circle colored red, surrounded by alternating circular strips (annuli) of white and black, say ten of each. Make sure that the final display shows circles, not ellipses. (*Hint*: one way to color the region between two circles black is to color the entire inside of the outer circle black, then color the inside of the inner circle white.)

6. MATLAB has a function **`lcm`** that finds the least common multiple of two numbers. Write a function M-file `mylcm.m` that finds the least common multiple of an arbitrary number of positive integers, which may be given as separate arguments or in a vector. For example, **`mylcm(4, 5, 6)`** and **`mylcm([4 5 6])`** should both produce the answer **`60`**. The program should produce a

helpful error message if any of the inputs are not positive integers. (*Hint*: for three numbers you could use **lcm** to find the least common multiple m of the first two numbers, then use **lcm** again to find the least common multiple of m and the third number. Your M-file can generalize this approach.)

7. ☆ Write a function M-file that takes as input a string containing the name of a text file and produces a histogram of the number of occurrences of each letter from A to Z in the file. Try to label the figure and axes as usefully as you can.

8. Consider the following linear programming problem. Jane Doe is running for County Commissioner. She wants to personally canvass voters in the four main cities in the county: Gotham, Metropolis, Oz, and River City. She needs to figure out how many residences (private homes, apartments, etc.) to visit in each city. The constraints are as follows.

   (i) She intends to leave a campaign pamphlet at each residence; she has only 50,000 available.

   (ii) The travel costs she incurs for each residence are \$0.50 in each of Gotham and Metropolis, \$1 in Oz, and \$2 in River City; she has \$40,000 available.

   (iii) The number of minutes (on average) that her visits to each residence require are 2 minutes in Gotham, 3 minutes in Metropolis, a minute in Oz, and 4 minutes in River City; she has 300 hours available.

   (iv) Because of political profiles Jane knows that she should not visit any more residences in Gotham than she does in Metropolis, and that, however many residences she visits in Metropolis and Oz, the total of the two should not exceed the number she visits in River City.

   (v) Jane expects to receive, during her visits, on average, campaign contributions of one dollar from each residence in Gotham, a quarter from those in Metropolis, a half-dollar from the Oz residents, and three dollars from the folks in River City. She must raise at least \$10,000 from her entire canvass.

   Jane's goal is to maximize the number of supporters (those likely to vote for her). She estimates that for each residence she visits in Gotham the odds are 0.6 that she picks up a supporter, and the corresponding probabilities in Metropolis, Oz, and River City are, respectively, 0.6, 0.5, and 0.3.

   (a) How many residences should she visit in each of the four cities?

   (b) Suppose that she can double the time she can allot to visits. Now what is the profile for visits?

   (c) But suppose that the extra time (in part (b)) also mandates that she double the contributions she receives. What is the profile now?

9. Consider the following linear programming problem. The famous football coach Joe Glibb is trying to decide how many hours to spend with each component of his offensive unit during the coming week – that is, the quarterback, the running backs, the receivers, and the linemen. The constraints are as follows.

(i) The number of hours available to Joe during the week is 50.

(ii) Joe figures he needs 20 points to win the next game. He estimates that for each hour he spends with the quarterback, he can expect a point return of 0.5. The corresponding numbers for the running backs, receivers and linemen are 0.3, 0.4, and 0.1, respectively.

(iii) In spite of their enormous size, the players are relatively thin skinned. Each hour with the quarterback is likely to require Joe to criticize him once. The corresponding numbers of criticisms per hour for the other three groups are 2 for running backs, 3 for receivers, and 0.5 for linemen. Joe figures he can bleat out only 75 criticisms in a week before he loses control.

(iv) Finally, the players are *prima donnas* who engage in rivalries. Because of that, he must spend exactly the same number of hours with the running backs as he does with the receivers, at least as many hours with the quarterback as he does with the runners and receivers combined, and at least as many hours with the receivers as with the linemen.

Joe worries that he's going to be fired at the end of the season regardless of the outcome of the game, so his primary goal is to maximize his pleasure during the week. (The team's owner should only know.) He estimates that, on a sliding scale from 0 to 1, he gets 0.2 units of personal satisfaction for each hour with the quarterback. The corresponding numbers for the runners, receivers and linemen are 0.4, 0.3, and 0.6, respectively.

(a) How many hours should Joe spend with each group?

(b) Suppose that he needs only 15 points to win; then how many?

(c) Finally suppose, despite needing only 15 points, that the troops are getting restless and he can only dish out only 70 criticisms this week. Is Joe getting the most out of his week?

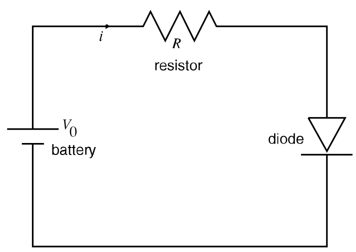


Figure C.1. A Nonlinear Circuit.

10. This problem, suggested to us by our colleague Tom Antonsen, concerns an electrical circuit, one of whose components does not behave linearly. Consider

the circuit in Figure C.1. Unlike the resistor, the diode is a non-linear element – it does not obey Ohm's Law. In fact its behavior is specified by the formula

$$i = I_0 \exp(V_D/V_T), \tag{C.1}$$

where i is the current in the diode (which is the same as that in the resistor by Kirchhoff's Current Law), V_D is the voltage across the diode, I_0 is the leakage current of the diode, and $V_T = kT/e$, where k is Boltzmann's constant, T is the temperature of the diode, and e is the electrical charge.

Now, by application of Ohm's Law to the resistor, we also know that $V_R = iR$, where V_R is the voltage across the resistor and R is its resistance. But, by Kirchhoff's Voltage Law, we also have $V_R = V_0 - V_D$. This gives a second equation relating the diode current and voltage, namely

$$i = (V_0 - V_D)/R. \tag{C.2}$$

Note now that (C.2) says that i is a decreasing linear function of V_D with value V_0/R when V_D is zero. At the same time (C.1) says that i is an exponentially growing function of V_D starting out at I_0. Since, typically, $RI_0 < V_0$, the two resulting curves (for i as a function of V_D) must cross once. Eliminating i from the two equations, we see that the voltage in the diode must satisfy the transcendental equation

$$(V_0 - V_D)/R = I_0 \exp(V_D/V_T),$$

or

$$V_D = V_0 - RI_0 \exp(V_D/V_T).$$

(a) Reasonable values for the electrical constants are $V_0 = 1.5$ volts, $R = 1000$ ohms, $I_0 = 10^{-5}$ amperes, and $V_T = 0.0025$ volts. Use **`fzero`** to find the voltage V_D and current i in the circuit.

(b) In the remainder of the problem, we assume that the voltage V_0 in the battery and the resistance R of the resistor are unchanged. But suppose that we have some freedom to alter the electrical characteristics of the diode. For example, suppose that I_0 is halved. What happens to the voltage?

(c) Suppose that, instead of halving I_0, we halve V_T. Then what is the effect on V_D?

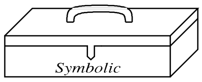
(d) Suppose that both I_0 and V_T are cut in half. What then?

(e) Finally, we want to examine the behavior of the voltage if both I_0 and V_T are decreased toward zero. For definitiveness, assume that we set $I_0 = 10^{-5}x$ and $V_T = 0.0025x$, and let $x \to 0$. Specifically, compute the solution for $x = 10^{-j}$, $j = 0, \ldots, 5$. Then, display a **`loglog`** plot of the solution values, for the voltage as a function of I_0. What do you conclude?

11. This problem is based on the *Population Dynamics* and *The 360° Pendulum* applications from Chapter 10. The growth of a species was modeled in the former by a *difference equation*. In this problem we will model population growth by a *differential equation*, akin to the second application mentioned above. In fact we can give a differential-equation model for the logistic growth of a population x as a function of time t by means of the equation

$$\dot{x} = x(1 - x) = x - x^2, \tag{C.3}$$

where $\dot{x}$ denotes the derivative of x with respect to t. We think of x as a fraction of some maximal possible population. One advantage of this continuous model over the discrete model in Chapter 10 is that we can get a "reading" of the population at any point in time (not just on integer intervals).

(a) 

The differential equation (C.3) is solved in any beginning course in ordinary differential equations, but you can do it easily with the MATLAB command **`dsolve`**. (Look up the syntax via online help.)

(b) Now find the solution assuming an initial value $x_0 = x(0)$ of x. Use the values $x_0 = 0, 0.25, \ldots, 2.0$. Graph the solutions and use your picture to justify the statement: "Regardless of $x_0 > 0$, the solution of (C.3) tends to the constant solution $x(t) \equiv 1$ in the long term."

The logistic model presumes two underlying features of population growth: (i) that ideally the population expands at a rate proportional to its current total (i.e., exponential growth – this corresponds to the x term on the right-hand side of (C.3)); and (ii) because of interactions between members of the species and natural limits to growth, unfettered exponential growth is held in check by the logistic term, given by the $-x^2$ expression in (C.3). Now assume that there are two species $x(t)$ and $y(t)$, competing for the same resources to survive. Then there will be another negative term in the differential equation that reflects the interaction between the species. The usual model presumes it to be proportional to the product of the two populations, and the larger the constant of proportionality, the more severe the interaction, as well as the resulting check on population growth.

(c) Here is a typical pair of differential equations that model the growth in population of two competing species $x(t)$ and $y(t)$:

$$\begin{aligned} \dot{x}(t) &= x - x^2 - 0.5xy \\ \dot{y}(t) &= y - y^2 - 0.5xy. \end{aligned} \tag{C.4}$$

The command **`dsolve`** can solve many pairs of ordinary differential equations – especially linear ones. But the mixture of quadratic terms in (C.4) makes it unsolvable symbolically, so we need to use a numerical

ODE solver as we did in the pendulum application. Using the commands in that application as a template, graph numerical solution curves to the system (C.4) for initial data

$$x(0) = 0 : 1/12 : 13/12$$
$$y(0) = 0 : 1/12 : 13/12.$$

(*Hint*: use **`axis`** to limit your view to the square $0 \leq x, y \leq 13/12$.)

(d) The picture you drew is called a *phase portrait* of the system. Interpret it. Explain the long-term behavior of any population distribution that starts with only one species present. Relate it to part (b). What happens in the long term if both populations are present initially? Is there an initial population distribution that remains undisturbed? What is it? Relate those numbers to the model (C.4).

(e) Now replace 0.5 in the model by 2, that is consider the new model

$$\begin{aligned} \dot{x}(t) &= x - x^2 - 2xy \\ \dot{y}(t) &= y - y^2 - 2xy. \end{aligned} \tag{C.5}$$

Draw the phase portrait. (Use the same initial data and viewing square.) Answer the same questions as in part (d). Do you see a special solution trajectory that emanates from near the origin and proceeds to the special fixed point? Do you see another trajectory from the upper right to the fixed point? What happens to all population distributions that do not start on these trajectories?

(f) Explain why model (C.4) is called "peaceful coexistence" and model (C.5) is called "doomsday." Now explain heuristically why the change in coefficient from 0.5 to 2 converts coexistence into doomsday.

12. Build a Simulink model corresponding to the pendulum equation

$$\ddot{x}(t) = -0.5\dot{x}(t) - 9.81\sin(x(t)) \tag{C.6}$$

from *The 360° Pendulum* in Chapter 10. You will need the Trigonometric Function block from the Math library. Use your model to redraw some of the phase portraits.

13. As you know, Galileo and Newton discovered that all bodies near the Earth's surface fall with the same acceleration g due to gravity, approximately 32.2 ft/sec^2. However, real bodies are also subjected to forces due to air resistance. If we take both gravity and air resistance into account, a moving ball can be modeled by the differential equation

$$\ddot{\mathbf{x}} = [0, -g] - c\,\|\dot{\mathbf{x}}\|\,\dot{\mathbf{x}}. \tag{C.7}$$

Here $\mathbf{x}$, a function of the time t, is the vector giving the position of the ball (the first coordinate is measured horizontally, the second one vertically), $\dot{\mathbf{x}}$ is the velocity vector of the ball, $\ddot{\mathbf{x}}$ is the acceleration of the ball, $\|\dot{\mathbf{x}}\|$ is the magnitude of the velocity, that is, the speed, and c is a constant depending on the surface characteristics and mass of the ball and the density of the air. (We are neglecting the lift force that comes from the ball's rotation, which can also play a major role in some situations, for instance in analyzing the path of a curve ball, as well as forces due to wind currents.) For a baseball, the constant c turns out to be approximately 0.0017, assuming that distances are measured in feet and time is measured in seconds. (See, for example, Chapter 18, "Balls and Strikes and Home Runs," in *Towing Icebergs, Falling Dominoes, and Other Adventures in Applied Mathematics*, by Robert Banks, Princeton University Press, Princeton, 1998.) Build a Simulink model corresponding to equation (C.7), and use it to study the trajectory of a batted baseball. Here are a few hints. Represent $\ddot{\mathbf{x}}$, $\dot{\mathbf{x}}$, and $\mathbf{x}$ as vector signals, joined by two Integrator blocks. The quantity $\ddot{\mathbf{x}}$, according to (C.7), should be computed from a Sum block with two vector inputs. One should be a Constant block with the vector value $[0, -32.2]$, representing gravity, and the other should represent the drag term on the right-hand side of equation (C.7), computed from the value of $\dot{\mathbf{x}}$. You should be able to change one of the parameters to study what happens both with and without air resistance (the cases of $c = 0.0017$ and $c = 0$, respectively). Attach the output to an XY Graph block, with the parameters x-min = 0, y-min = 0, x-max = 500, y-max = 150, so you can see the path of the ball out to a distance of 500 feet from home plate and up to a height of 150 feet.

(a) Let $\mathbf{x}(0) = [0, 4]$ and $\dot{\mathbf{x}}(0) = [80, 80]$. (This corresponds to the ball starting at $t = 0$ from home plate, 4 feet off the ground, with the horizontal and vertical components of its velocity both equal to 80 ft/sec. This corresponds to a speed off the bat of about 77 mph, which is not unrealistic.) How far (approximately – you can read this from your XY Graph output) will the ball travel before it hits the ground, both with and without air resistance? About how long will it take the ball to hit the ground, and how fast will the ball be traveling at that time (again, both with and without air resistance)? (The last parts of the question are relevant for outfielders.)

(b) Suppose that a game is played in Denver, Colorado, where because of thinning of the atmosphere due to the high altitude, c is only 0.0014. How far will the ball travel now (given the same initial velocity as in (a))?

(c) (This is not a MATLAB problem.) Estimate from a comparison of your answers to (a) and (b) what effect altitude might have on the team batting average and team earned run average of the Colorado Rockies.

14. ☆ Write a function M-file (with no arguments or outputs) that scans the current directory for the most recently modified M-file and opens it in the Editor/Debugger. If the current directory contains no M-files, your M-file should produce an error message. Ideally, you should try to write the M-file so it works

in either Windows or UNIX, but at least make it work with your own operating system.

15. ✰ Consider a sequence of complex numbers generated from a starting value z_0 by the rule $z_{n+1} = z_n^2 - 0.75$. For some values of z_0, the sequence of numbers will diverge to infinity as n increases, but for other values of z_0, the sequence will remain inside a bounded region forever. The boundary of the set consisting of the latter values of z_0 is called the *Julia set* for the function $f(z) = z^2 - 0.75$. Use **`image`** or **`imagesc`** to draw a picture of this Julia set. (*Hint*: the Julia set lies within the region where the real part of z_0 is between -2 and 2, and its imaginary part is between -1.5 and 1.5. Form a grid of z_0 values within this rectangle (we suggest using roughly a 300×400 array) and calculate the corresponding values of $z_1, z_2, \ldots, z_n$ for some n. Color the pixel of each z_0 according to how large the corresponding z_n is. The larger you make n, the more accurate (and interesting) the picture will be, but the computation will also take longer, so start with a relatively small value of n and work your way up.)