Modeling Basics

P. Howard

Fall 2009

Contents

1 Overview

All modeling projects begin with the identification of a situation of one form or another that appears to have at least some aspect that can be described mathematically. The first two steps of the project, often taken simultaneously, become: (1) gain a broad understanding of the situation to be modeled, and (2) collect data. Depending on the project, (1) and (2) can take minutes, hours, days, weeks, or even years. Asked to model the rebound height of a tennis ball, given an initial drop height, we immediately have a fairly broad understanding of the problem and suspect that collecting data won't take more than a few minutes with a tape measure and a stopwatch. Asked, on the other hand, to model the progression of Human Immunodeficiency Virus (HIV) as it attacks the body, we might find ourselves embarking on lifetime careers.

2 Curve Fitting and Parameter Estimation

Often, the first step of the modeling process consists of simply looking at data graphically and trying to recognize trends. In this section, we will study the most standard method of curve fitting and parameter estimation: the method of least squares.

Example 2.1. Suppose the Internet auctioneer, eBay, hires us to predict its net income for the year 2003, based on its net incomes for 2000, 2001, and 2002 (see Table 2.1).

Year	Net Income
2000	48.3 million
2001	90.4 million
2002	249.9 million

Table 2.1: Yearly net income for eBay.

We begin by simply plotting this data as a *scatterplot* of points. In MATLAB, we develop Figure 2.1 through the commands,

```
>>year=[0 1 2];
>>income=[48.3 90.4 249.9];
>>plot(year,income,'o')
>>axis([-.5 2.5 25 275])
```

Our first approach toward predicting eBay's future profits might be to simply find a curve that best fits this data. The most common form or curve fitting is *linear least squares regression.* △

2.1 Polynomial Regression

In order to develop an idea of what we mean by "best fit" in this context, we begin by trying to draw a line through the three points of Example 2.1 in such away that the distance between the points and the line is minimized (see Figure 2.2).

Labeling our three points (x_1, y_1), (x_2, y_2), and (x_3, y_3), we observe that the vertical distance between the line and the point (x_2, y_2) is given by the error $E_2 = |y_2 - mx_2 - b|$. The idea behind the least squares method is to sum these vertical distances and minimize the total error. In practice, we square the errors both to keep them positive and to avoid possible

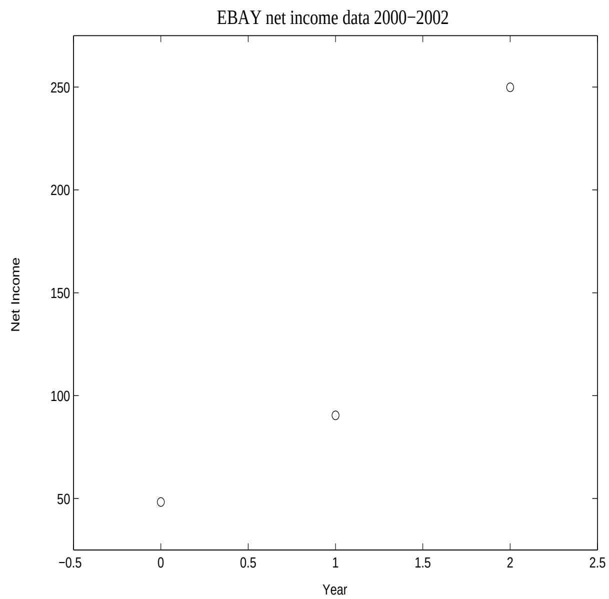

Figure 2.1: Net Income by year for eBay.

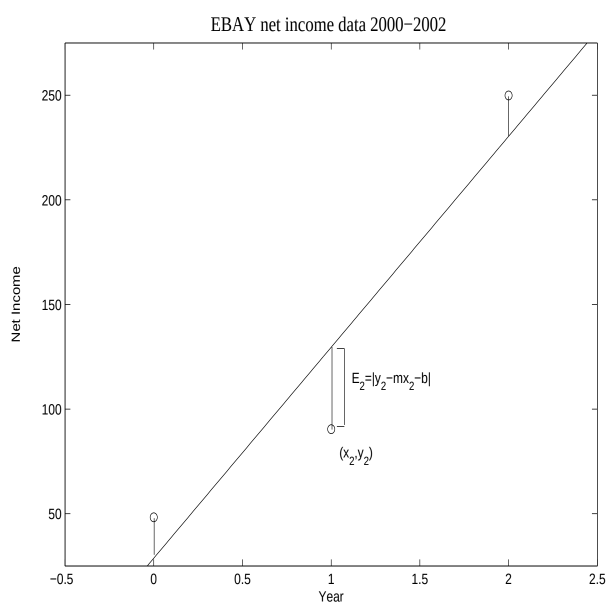

Figure 2.2: Least squares vertical distances.

difficulty with differentiation (recall that absolute values can be subtle to differentiate), which will be required for minimization. Our total least squares error becomes

$$E(m,b) = \sum_{k=1}^{n} (y_k - mx_k - b)^2.$$

In our example, $n = 3$, though the method remains valid for any number of data points.

We note here that in lieu of these vertical distances, we could also use horizontal distances between the points and the line or direct distances (the shortest distances between the points and the line). While either of these methods could be carried out in the case of a line, they both become considerably more complicated in the case of more general cures. In the case of a parabola, for example, a point would have two different horizontal distances from the curve, and while it could only have one shortest distance to the curve, computing that distance would be a fairly complicted problem in its own right.

Returning to our example, our goal now is to find values of m and b that minimize the error function $E(m,b)$. In order to maximize or minimize a function of multiple variables, we compute the partial derivative with respect to each variable and set them equal to zero. Here, we compute

$$\begin{aligned} \frac{\partial}{\partial m} E(m,b) =& 0 \\ \frac{\partial}{\partial b} E(m,b) =& 0. \end{aligned}$$

We have, then,

$$\begin{aligned} \frac{\partial}{\partial m} E(m,b) =& -2 \sum_{k=1}^{n} x_k (y_k - mx_k - b) = 0, \\ \frac{\partial}{\partial b} E(m,b) =& -2 \sum_{k=1}^{n} (y_k - mx_k - b) = 0, \end{aligned}$$

which we can solve as a linear system of two equations for the two unknowns m and b. Rearranging terms and dividing by 2, we have

$$\begin{aligned} m \sum_{k=1}^{n} x_k^2 + b \sum_{k=1}^{n} x_k =& \sum_{k=1}^{n} x_k y_k, \\ m \sum_{k=1}^{n} x_k + b \sum_{k=1}^{n} 1 =& \sum_{k=1}^{n} y_k. \end{aligned} \tag{2.1}$$

Observing that $\sum_{k=1}^{n} 1 = n$, we multiply the second equation by $\frac{1}{n} \sum_{k=1}^{n} x_k$ and subtract it from the first to get the relation,

$$m \Big(\sum_{k=1}^{n} x_k^2 - \frac{1}{n} (\sum_{k=1}^{n} x_k)^2 \Big) = \sum_{k=1}^{n} x_k y_k - \frac{1}{n} (\sum_{k=1}^{n} x_k)(\sum_{k=1}^{n} y_k),$$

or

$$m = \frac{\sum_{k=1}^n x_k y_k - \frac{1}{n}(\sum_{k=1}^n x_k)(\sum_{k=1}^n y_k)}{\sum_{k=1}^n x_k^2 - \frac{1}{n}(\sum_{k=1}^n x_k)^2}.$$

Finally, substituting m into equation (2.1), we have

$$\begin{aligned} b &= \frac{1}{n}\sum_{k=1}^n y_k - (\sum_{k=1}^n x_k)\frac{\sum_{k=1}^n x_k y_k - \frac{1}{n}(\sum_{k=1}^n x_k)(\sum_{k=1}^n y_k)}{n\sum_{k=1}^n x_k^2 - (\sum_{k=1}^n x_k)^2} \\ &= \frac{(\sum_{k=1}^n y_k)(\sum_{k=1}^n x_k^2) - (\sum_{k=1}^n x_k)(\sum_{k=1}^n x_k y_k)}{n\sum_{k=1}^n x_k^2 - (\sum_{k=1}^n x_k)^2}. \end{aligned}$$

We can verify that these values for m and b do indeed constitute a minimum by observing that by continuity there must be at least one local minimum for E, and that since m and b are uniquely determined this must be it. Alternatively, we have Theorem A.3 from Appendix A.

Observe that we can proceed similarly for any polynomial. For second order polynomials with general form $y = a_0 + a_1 x + a_2 x^2$, our error becomes

$$E(a_0, a_1, a_2) = \sum_{k=1}^n (y_k - a_0 - a_1 x_k - a_2 x_k^2)^2.$$

In this case, we must compute a partial derivative of E with respect to each of three parameters, and consequently (upon differentiation) solve three linear equations for the three unknowns.

The MATLAB command for polynomial fitting is *polyfit(x,y,n)*, where x and y are vectors and n is the order of the polynomial. For the eBay data, we have

```
>>polyfit(year,income,1)
ans =
100.8000 28.7333
```

Notice particularly that, left to right, MATLAB returns the coefficient of the highest power of x first, the second highest power of x second etc., continuing until the y-intercept is given last. Alternatively, for polynomial fitting up to order 10, MATLAB has the option of choosing it directly from the graphics menu. In the case of our eBay data, while Figure 1 is displayed in MATLAB, we choose **Tools, Basic Fitting**. A new window opens and offers a number of fitting options. We can easily experiment by choosing the **linear** option and then the **quadratic** option, and comparing. (Since we only have three data points in this example, the quadratic fit necessarily passes through all three. This, of course, does not mean that the quadratic fit is best, only that we need more data.) For this small a set of data, the linear fit is safest, so select that one and click on the black arrow at the bottom right corner of the menu. Checking that the fit given in the new window is linear, select the option **Save to Workspace**. MATLAB saves the polynomial fit as a *structure*, which is a MATLAB array variable that can hold data of varying types; for example, a string as its first element and a digit as its second and so on. The elements of a structure can be accessed through the notation *structurename.structureelement*. Here, the default structure name is *fit*, and the first element is *type*. The element *fit.type* contains a string describing the structure. The second element of the structure is *fit.coeff*, which contains the polynomial coefficients of our fit. Finally, we can make a prediction with the MATLAB command *polyval*,

```
>>polyval(fit.coeff,3)
```

for which we obtain the prediction 331.1333. Finally, we mention that MATLAB refers to the error for its fit as the *norm of residuals*, which is precisely the square root of E as we've defined it.

Example 2.2. (Crime and Unemployment.) Suppose we are asked to model the connection between unemployment and crime in the United States during the period 1994–2001. We might suspect that in some general way increased unemployment leads to increased crime ("idle hands are the devil's playground"), but our first step is to collect data. First, we contact the Federal Bureau of Investigation and study their Uniform Crime Reports (UCR), which document, among other things, the United States' crime rate per 100,000 citizens (we could also choose to use data on violent crimes only or gun-related crimes only etc., each of which is a choice our model will make). Next, we contact the U.S. Bureau of Labor and obtain unemployment percents for each year in our time period. Summarizing our data we develop Table 2.2.

Year	Crime Rate	Percent Unemployment
1994	5,373.5	6.1%
1995	5,277.6	5.6%
1996	5,086.6	5.4%
1997	4,922.7	4.9%
1998	4,619.3	4.5%
1999	4,266.8	4.2%
2000	4,124.8	4.0%
2001	4,160.5	4.8%

Table 2.2: Crime rate and unemployment data.

Proceeding as with Example 2.1, we first look at a scatterplot of the data, given in Figure 2.3.

Certainly the first thing we observe about our scatterplot is that there does seem to be a distinct connection: as unemployment increases, crime rate increases. In fact, aside from the point for 2001, the trend appears fairly steady. In general, we would study this point at the year 2001 very carefully and try to determine whether this is an anomaly or a genuine shift in the paradigm. For the purposes of this example, however, we're going to treat it as an *outlyer*—a point that for one reason or another doesn't follow an otherwise genuinely predictive model. The important point to keep in mind is that discarding outlyers when fitting data is a perfectly valid approach, *so long as you continue to check and reappraise your model as future data becomes available.*

Discarding the outlyer, we try both a linear fit and a quadratic fit, each shown on Figure 2.4.

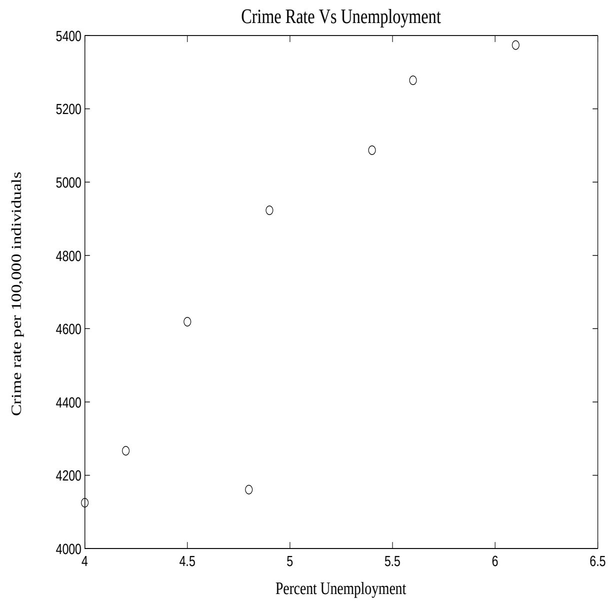

Figure 2.3: Scatterplot of crime rate versus unemployment.

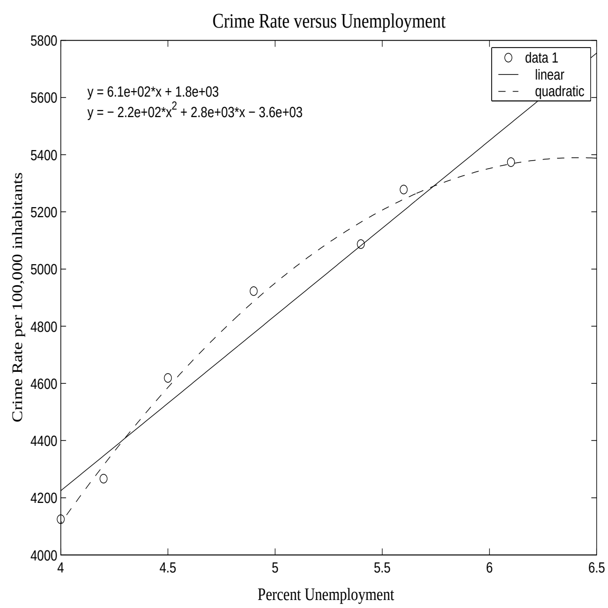

Figure 2.4: Best fit curves for crime–unemployment data.

Clearly, the quadratic fit is better, and though we haven't yet quantitatively developed the socioeconomic reasons for this particular relation,[1] we do have a genuinely predictive model that we can test against future data. For example, the percent unemployment in 2002 was 5.8%, which our quadratic model would predict should be associated with a crime rate of 5,239.2 crimes per 100,000 inhabitants. The actual crime rate for 2002 (not yet finalized) was 4,214.6. At this point, we are led to conclude that our model is not sufficiently accurate. In this case, the problem is most likely a lag effect: while a short-term rise in the unemployment rate doesn't seem to have much effect on crime, perhaps a sustained rise in unemployment would. Though for now we will leave this as a question for someone else to grapple with. △

2.2 Alternative Methods of Curve Fitting

Though least squares regression is certainly the most popular form of basic curve fitting, it is not the only method that can be applied. We could also think, for example, of defining our error as the maximum of all the vertical distances between the points and our curve. That is, for n data points, we could define

$$E = \max |y_k - f(x_k)|; \quad k = 1, ..., n.$$

(This is typically referred to as the Chebyshev method.) We could also sum the unsquared absolute values,

$$E = \sum_{k=1}^{n} |y_k - f(x_k)|,$$

or take any other reasonable measure.

2.3 Regression with more general functions

Example 2.3. Yearly temperature fluctuations are often modeled by trigonometic expressions, which lead to more difficult regression analyses. Here, we'll consider the case of monthly average maximum temperatures in Big Bend National Park. In Table 2.3 the first column lists raw data of average maximum temperatures each month. In order to model this data with a simple trigonometic model, we'll subtract the mean (which gives Column 3) and divide by the maximum absolute value (which gives Column 4) to arrive at a column of dependent variables that vary like sin and cos between -1 and +1.

A reasonable model for this data is $T(m) = \sin(m - a)$ (T represents scaled temperatures and m represents a scaled index of months ($m \in [0, 2\pi]$)), where by our reductions we've limited our analysis to a single parameter, a. Proceeding as above, we consider the regression error

$$E(a) = \sum_{k=1}^{n} (T_k - \sin(m_k - a))^2.$$

Computing $\partial_a E(a) = 0$, we have

$$2 \sum_{k=1}^{n} (T_k - \sin(m_k - a)) \cos(m_k - a) = 0,$$

[1] And don't worry, we're not going to.

Month	Average Max Temp	Minus Mean	Scaled
Jan.	60.9	-18.0	-1.00
Feb.	66.2	-12.7	-.71
Mar.	77.4	-1.5	-.08
Apr.	80.7	1.8	.10
May	88.0	9.1	.51
June	94.2	15.3	.85
July	92.9	14.0	.78
Aug.	91.1	12.2	.68
Sept.	86.4	7.5	.42
Oct.	78.8	-.1	-.01
Nov.	68.5	-10.4	-.58
Dec.	62.2	-16.7	-.93

Table 2.3: Average maximum temperatures for Big Bend National Park.

a nonlinear equation for the parameter a. Though nonlinear algebraic equations are typically difficult to solve analytically, they can certainly be solve numerically. In this case, we will use MATLAB's $fzero()$ function. First, we write an M-file that contains the function we're setting to zero, listed below as *bigbend.m*.

```
function value = bigbend(a);
%BIGBEND: M-file containing function for
%fitting Big Bend maximum temperatures.
scaledtemps = [-1.0 -.71 -.08 .10...
.51 .85 .78 .68 .42 -.01 -.58 -.93];
value = 0;
for k=1:12
m=2*pi*k/12;
value = value + (scaledtemps(k)-sin(m-a))*cos(m-a);
end
```

Finally, we solve for a and compare our model with the data, arriving at Figure 2.5.[2]

```
>>months=1:12;
>>temps=[60.9 66.2 77.4 80.7 88.0 94.2 92.9 91.1 86.4 78.8 68.5 62.2];
>>fzero(@bigbend,1)
ans =
1.9422
>>modeltemps=18*sin(2*pi*months/12-1.9422)+78.9;
>>plot(months,temps,'o',months,modeltemps)
```

△

[2]In practice, it's better to write short M-files to carry out this kind of calculation, rather than working at the Command Window, but for the purposes of presentation the Command Window prompt (>>) helps distinguished what I've typed in from what MATLAB has spit back.

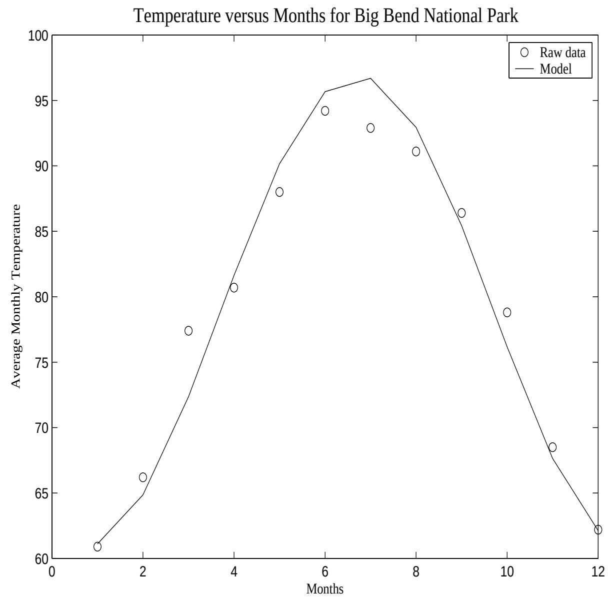

Figure 2.5: Trigonometric model for average monthly temperatures in Big Bend National Park.

For multiple parameters, we must solve a system of nonlinear algebraic equations, which can generally be quite difficult even computationally. For this, we will use the MATLAB function *lsqcurvefit()*.

Example 2.4. Let's consider population growth in the United States, beginning with the first government census in 1790 (see Table 2.4).

Year	1790	1800	1810	1820	1830	1840	1850	1860	1870	1880	1890	1900
Pop	3.93	5.31	7.24	9.64	12.87	17.07	23.19	31.44	39.82	50.16	62.95	75.99

Year	1910	1920	1930	1940	1950	1960	1970	1980	1990	2000
Pop	91.97	105.71	122.78	131.67	151.33	179.32	203.21	226.50	249.63	281.42

Table 2.4: Population data for the United States, 1790–2000, measured in millions.

If we let $p(t)$ represent the population at time t, the *logistic* model of population growth is given by

$$\frac{dp}{dt} = rp(1 - \frac{p}{K}); \quad p(0) = p_0 \tag{2.2}$$

where p_0 represents the initial population of the inhabitants, r is called the "growth rate" of the population, and K is called the "carrying capacity." Observe that while the rate at which the population grows is assumed to be proportional to the size of the population, the population is assumed to have a maximum possible number of inhabitants, K. (If $p(t)$ ever grows larger than K, then $\frac{dp}{dt}$ will become negative and the population will decline.) Equation (2.2) can be solved by separation of variables and partial fractions, and we find

$$p(t) = \frac{p_0 K}{(K - p_0)e^{-rt} + p_0}. \tag{2.3}$$

Though we will take year 0 to be 1790, we will assume the estimate that year was fairly crude and obtain a value of p_0 by fitting the entirety of the data. In this way, we have three parameters to contend with, and carrying out the full regression analysis would be tedious.

The first step in finding values for our parameters with MATLAB consists of writing our function $p(t)$ as an M-file, with our three parameters p_0, r, and K stored as a parameter vector $p = (p(1), p(2), p(3))$:

```
function P = logistic(p,t);
%LOGISTIC: MATLAB function file that takes
%time t, growth rate r (p(1)),
%carrying capacity K (p(2)),
%and initial population P0 (p(3)), and returns
%the population at time t.
P = p(2).*p(3)./((p(2)-p(3)).*exp(-p(1).*t)+p(3));
```

MATLAB's function *lsqcurvefit()* can now be employed at the Command Window, as follows. (The data vectors *decades* and *pops* are defined in the M-file *pop.m*, available on [1].)

```
>>decades=0:10:210;
>>pops=[3.93 5.31 7.24 9.64 12.87 17.07 23.19 31.44 39.82 50.16 62.95 75.99...
91.97 105.71 122.78 131.67 151.33 179.32 203.21 226.5 249.63 281.42];
>>p0 = [.01 1000 3.93];
>>[p error] = lsqcurvefit(@logistic,p0,decades,pops)
Optimization terminated: relative function value
changing by less than OPTIONS.TolFun.
p =
0.0215 445.9696 7.7461
error =
440.9983
>>sqrt(error)
ans =
21.0000
```

After defining the data, we have entered our initial guess as to what the parameters should be, the vector p_0. (Keep in mind that MATLAB is using a routine similar to *fzero(),* and typically can only find roots reasonably near our guesses.) In this case, we have guessed a small value of r corresponding very roughly to the fact that we live 70 years and have on average (counting men and women) 1.1 children per person[3] ($r \cong 1/70$), a population carrying capacity of 1 billion, and an initial population equal to the census data. Finally, we use *lsqcurvefit()*, entering repectively our function file, our initial parameter guesses, and our data. The function *lsqcurvefit()* renders two outputs: our parameters and a sum of squared errors, which we have called error. Though the error looks enormous, keep in mind that this is a sum of all errors squared,

$$\text{error} = \sum_{\text{decades}} (\text{pops(decade)-modelpops(decade)})^2.$$

[3] According to census 2000.

A more reasonable measure of error is the square root of this, from which we see that over 22 decades our model is only off by around 21.54 million people. Though MATLAB's iteration has closed successfully, we will alter our implementation slightly in order to see how changes can be made in general. Let's specify a lower bound of 0 on each of the parameter values (certainly none should be negative physically), and let's reduce MATLAB's default value for this variable *TolFun.* (The function should be changing by less than the value of *TolFun* with sufficiently small changes in the parameter values; MATLAB's default is 10^{-6}. If you want to see the the default values for all options use

```
>>optimset lsqcurvefit
```

Here, we omit the output, which is quite lengthy.

```
>>lowerbound=[0 0 0];
>>options=optimset('TolFun',1e-8);
>>[p error]=lsqcurvefit(@logistic,p0,decades,pops,lowerbound,[],options)
Optimization terminated: relative function value
changing by less than OPTIONS.TolFun.
p =
0.0215 445.9696 7.7461
error =
440.9983
>>sqrt(error)
ans =
21.0000
>>modelpops=logistic(p,decades);
>>plot(decades,pops,'o',decades,modelpops)
```

In our new *lsqcurvefit()* command, the vector *lowerbound* is added at the end of the required inputs, followed by square brackets, [], which signify that we are not requiring an upper bound in this case. Finally, the options defined in the previous line are incorporated as the final input. Since the fit was already terminating successfully we don't see very little improvement (the error is slightly smaller in the last two decimal places, before we take square root.) For the parameters, we observe that r has remained small (roughly 1/50), our carrying capacity is 446 million people, and our initial population is 7.75 million. In the last two lines of code, we have created Figure 2.6, in which our model is compared directly with our data. △

2.4 Multivariate Regression

Often, the phenomenon we would like to model depends on more than one independent variable. (Keep in mind the following distinction: While the model in Example 2.4 depended on three parameters, it depended on only a single independent variable, t.)

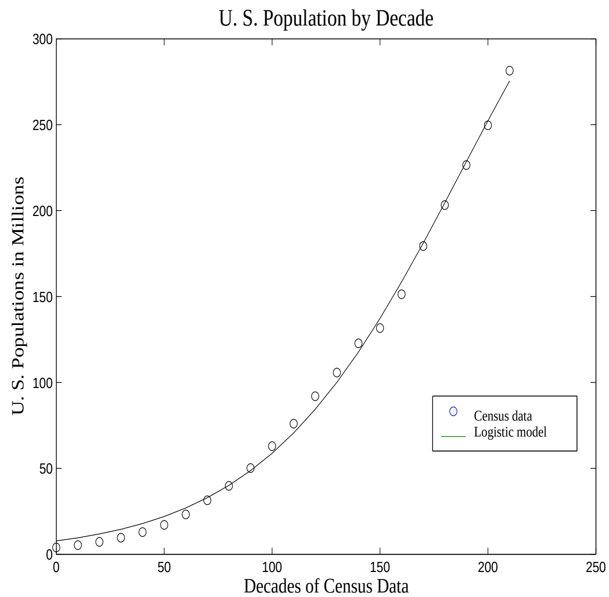

Figure 2.6: U.S. Census data and logistic model approximation.

2.4.1 Linear Fits

Example 2.5. Film production companies such as Paramount Studios and MGM employ various techniques to predict movie ticket sales. In this example, we will consider the problem of predicting final sales based on the film's first weekend. The first difficulty we encounter is that first weekend sales often depend more on hype than quality. For example, *Silence of the Lambs* and *Dude, Where's My Car?* had surprisingly similar first-weekend sales: \$13,766,814 and \$13,845,914 respectively.[4] (*Dude* did a little better.) Their final sales weren't so close: \$130,726,716 and \$46,729,374, again respectively. Somehow, our model has to *numerically* distinguish between movies like *Silence of the Lambs* and *Dude, Where's My Car?* Probably the easiest way to do this is by considering a second variable, the movie's rating. First weekend sales, final sales and TV Guide ratings are listed for ten movies in Table 2.5.[5]

Letting S represent first weekend sales, F represent final sales, and R represent ratings, our model will take the form

$$F = a_0 + a_1 S + a_2 R,$$

where a_0, a_1, and a_2 are parameters to be determined; that is, we will use a linear two-variable polynomial. For each set of data points from Table 2.5 (S_k, R_k, F_k) $(k = 1, ..., 10)$ we have an equation

$$F_k = a_0 + a_1 S_k + a_2 R_k.$$

[4] All sales data for this example were obtained from http://www.boxofficeguru.com. They represent domestic (U. S.) sales.

[5] TV Guide's ratings were easy to find, so I've used them for the example, but they're actually pretty lousy. FYI.

Movie	First Weekend Sales	Final Sales	Rating
Dude, Where's My Car?	13,845,914	46,729,374	1.5
Silence of the Lambs	13,766,814	130,726,716	4.5
We Were Soldiers	20,212,543	78,120,196	2.5
Ace Ventura	12,115,105	72,217,396	3.0
Rocky V	14,073,170	40,113,407	3.5
A.I.	29,352,630	78,579,202	3.0
Moulin Rouge	13,718,306	57,386,369	2.5
A Beautiful Mind	16,565,820	170,708,996	3.0
The Wedding Singer	18,865,080	80,245,725	3.0
Zoolander	15,525,043	45,172,250	2.5

Table 2.5: Movie Sales and Ratings.

Combining the ten equations (one for each k) into matrix form, we have

$$\begin{pmatrix} 1 & S_1 & R_1 \\ 1 & S_2 & R_2 \\ \vdots & \vdots & \vdots \\ 1 & S_{10} & R_{10} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ F_{10} \end{pmatrix}, \tag{2.4}$$

or $Ma = F$, where

$$M = \begin{pmatrix} 1 & S_1 & R_1 \\ 1 & S_2 & R_2 \\ \vdots & \vdots & \vdots \\ 1 & S_{10} & R_{10} \end{pmatrix}, \qquad a = \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix}, \qquad \text{and} \qquad F = \begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ F_{10} \end{pmatrix}.$$

In general, equation (2.4) is over-determined—ten equations and only three unknowns. In the case of over-determined systems, MATLAB computes $a = M^{-1}F$ by linear least squares regression, as above, starting with the error $E = \sum_{k=1}^{10}(F_k - a_0 - a_1 S_k - a_2 R_k)^2$. Hence, to carry out regression in this case on MATLAB, we use the following commands (The data vectors *S*, *F* and *R* are defined in the M-file *movies.m*, available on [1].):

```
>>S=[13.8 13.8 20.2 12.1 14.1 29.4 13.7 16.6 18.9 15.5]';
>>F=[46.7 130.7 78.1 72.2 40.1 78.6 57.4 170.7 80.2 45.2]';
>>R=[1.5 4.5 2.5 3.0 3.5 3.0 2.5 3.0 3.0 2.5]';
>>M=[ones(size(S)) S R];
>>a=M\F
a =
-6.6986
0.8005
25.2523
```

MATLAB's notation for $M^{-1}F$ is $M\backslash F$, which was one of MATLAB's earliest conventions, from the days when it was almost purely meant as a tool for matrix manipulations. Note finally that in the event that the number of equations is the same as the number of variables, MATLAB solves for the variables exactly (assuming the matrix is invertible).

At his point, we are in a position to make a prediction. In 2003, the Affleck–Lopez debacle *Gigli* opened with an initial weekend box office of \$3,753,518, and a *TV Guide* rating of 1.0 stars. Following the code above, we can predict the final box office for *Gigli* with the code:

```
>>a(1)+a(2)*3.8+a(3)*1
ans =
21.5957
```

The actual final sales of *Gigli* were \$6,087,542, wherein we conclude that our model is not quite sufficient for predicting that level of disaster.

Though scatterplots can be difficult to read in three dimensions, they are often useful to look at. In this case, we simply type *scatter3(S,R,F)* at the MATLAB Command Window prompt to obtain Figure 2.7.

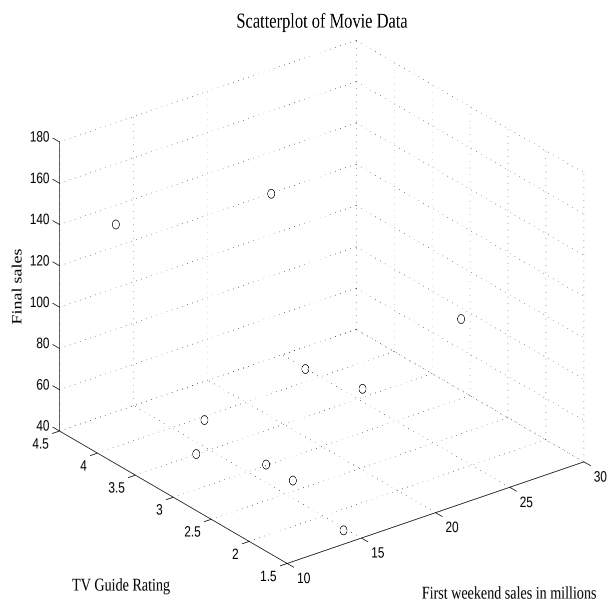

Figure 2.7: Scatterplot for movie sales data.

While in the case of a single variable, regression found the best-fit line, for two variables it finds the best-fit plane. Employing the following MATLAB code, we draw the best fit plane that arose from our analysis (see Figure 2.8).

```
>>hold on
>>x=10:1:30;
>>y=1:.5:5;
>>[X,Y]=meshgrid(x,y);
>>Z=a(1)+a(2)*X+a(3)*Y;
>>surf(X,Y,Z)
```

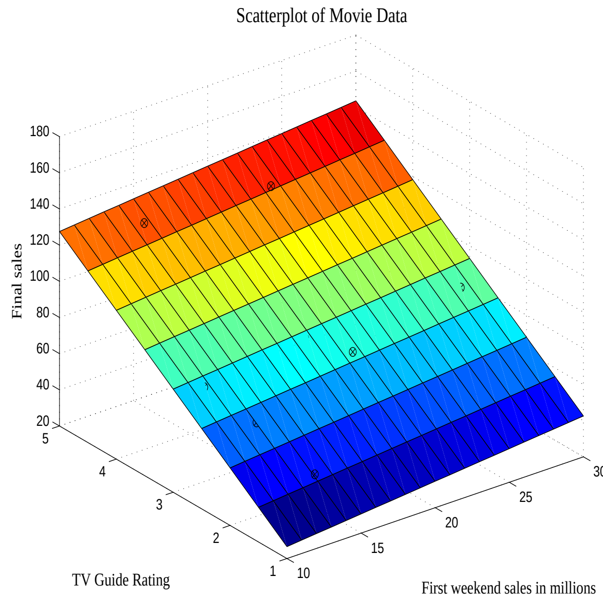

Figure 2.8: Movie sales data along with best-fit plane.

2.4.2 Multivariate Polynomials

In the previous section we fit data to a first order polynomial in two variables. More generally, we can try to fit data to higher order multidimensional polynomials. For example, a second order polynomial for a function of two variables has the form

$$z = f(x, y) = p_1 + p_2x + p_3y + p_3x^2 + p_4xy + p_5y^2.$$

If we have n data points $\{(x_k, y_k, z_k)\}_{k=1}^n$ then as in the previous section we obtain a system of n equations for the six unknowns $\{p_j\}_{j=1}^6$:

$$z_k = p_1 + p_2x_k + p_3y_k + p_4x_k^2 + p_5x_ky_k + p_6y_k^2, \quad k = 1, \ldots, n.$$

In matrix form,

$$\begin{pmatrix} 1 & x_1 & y_1 & x_1^2 & x_1y_1 & y_1^2 \\ 1 & x_2 & y_2 & x_2^2 & x_2y_2 & y_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & y_n & x_n^2 & x_ny_n & y_n^2 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix}.$$

Example 2.6. Use a second order polynomial in two variables to fit the movie data from Example 2.5.

In this case we need to create the 10×6 matrix

$$M = \begin{pmatrix} 1 & S_1 & R_1 & S_1^2 & S_1R_1 & R_1^2 \\ 1 & S_2 & R_2 & S_2^2 & S_2R_2 & R_2^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & S_n & R_n & S_n^2 & S_nR_n & R_n^2 \end{pmatrix},$$

and compute (in the MATLAB sense)

$$\vec{p} = M \backslash \vec{F}.$$

We have:

```
>>M=[ones(size(S)) S R S.^2 S.*R R.^2]
M =
1.0000 13.8000 1.5000 190.4400 20.7000 2.2500
1.0000 13.8000 4.5000 190.4400 62.1000 20.2500
1.0000 20.2000 2.5000 408.0400 50.5000 6.2500
1.0000 12.1000 3.0000 146.4100 36.3000 9.0000
1.0000 14.1000 3.5000 198.8100 49.3500 12.2500
1.0000 29.4000 3.0000 864.3600 88.2000 9.0000
1.0000 13.7000 2.5000 187.6900 34.2500 6.2500
1.0000 16.6000 3.0000 275.5600 49.8000 9.0000
1.0000 18.9000 3.0000 357.2100 56.7000 9.0000
1.0000 15.5000 2.5000 240.2500 38.7500 6.2500
>>p=M\F
p =
136.1870
2.3932
-126.6739
-0.6425
8.5905
5.5219
```

Our model is

$$F(S,R) = 136.1870 + 2.3932S - 126.6739R - .6425S^2 + 8.5905SR + 5.5219S^2.$$

Below, we carry out some example calculations, respectively for *Dude, Where's My Car*, *Silence of the Lambs*, and *Gigli*. Note that while this model seems to work well for the data it models, it is not a good predictor. Clearly, this is not an appropriate model for movie sales.

```
>>F=@(S,R) p(1)+p(2)*S+p(3)*R+p(4)*S^2+p(5)*S*R+p(6)*R^2
F =
@(S,R)p(1)+p(2)*S+p(3)*R+p(4)*S^2+p(5)*S*R+p(6)*R^2
>>F(13.8,1.5)
ans =
47.0945
>>F(13.8,4.5)
ans =
122.1153
>>F(3.8,1)
ans =
47.4956
```

2.4.3 Linear Multivariate Regression

We can generalize the approach taken in the previous section to any collection of functions $\{h_j(\vec{x})\}_{j=1}^m$, where $\vec{x} \in \mathbb{R}^r$. We look for models of the form

$$f(\vec{x}) = \sum_{j=1}^{m} p_j h_j(\vec{x}).$$

For example, this is the form of a second order polynomial with two variables with

$$\begin{aligned} h_1(\vec{x}) &= 1 \\ h_2(\vec{x}) &= x_1 \\ h_3(\vec{x}) &= x_2 \\ h_4(\vec{x}) &= x_1^2 \\ h_5(\vec{x}) &= x_1 x_2 \\ h_6(\vec{x}) &= x_2^2. \end{aligned}$$

In the general case, if we have n data points $\{(\vec{x}_k, z_k)\}_{k=1}^n$ we obtain a linear system of n equations for m unknowns

$$z_k = \sum_{j=1}^{m} p_j h_j(\vec{x}_k).$$

We set

$$H = \begin{pmatrix} h_1(\vec{x}_1) & h_2(\vec{x}_1) & \cdots & h_m(\vec{x}_1) \\ h_1(\vec{x}_2) & h_2(\vec{x}_2) & \cdots & h_m(\vec{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ h_1(\vec{x}_n) & h_2(\vec{x}_n) & \cdots & h_m(\vec{x}_n) \end{pmatrix},$$

and compute (in the MATLAB sense)

$$\vec{p} = H \backslash \vec{z}.$$

2.5 Transformations to Linear Form

We saw in Section 2.3 that in principle nonlinear regression can be used to estimate the parameters for arbitrary functions, so in particular we certainly don't require that our functions be in linear form. On the other hand, lines are certainly the easiest fits to recognize visually, and so it's often worthwhile at least looking for a linear relationship for your data.

2.5.1 Functions of a Single Variable

Each of the following nonlinear relationships can be put into a linear form:

$$\begin{aligned} y &= ax^b \\ y &= a \cdot b^x \\ y &= a \ln x + b \\ y &= \frac{1}{ax + b}. \end{aligned}$$

(This certainly isn't an exhaustive list, but these are some very useful examples.) For example, if we suspect our data has the form $y = ax^b$ we might take the natural log of both sides to write

$$\ln y = \ln a + b \ln x.$$

If we now plot $\ln y$ versus $\ln x$ the slope of the plot will be b and the y-intercept will be $\ln a$, from which we can obtain a value for a by exponentiation.

2.5.2 Functions of Multiple Variables

Each of the following multivariate relationships can be put into a linear form:

$$\begin{aligned} z &= ax^b y^c \\ z &= ab^x c^y \\ z &= a + b \ln x + c \ln y \\ z &= \frac{1}{a + bx + cy}. \end{aligned}$$

(Again this is certainly not an exhaustive list.) For example, if we suspect our data has the form $z = ax^b y^c$ we can compute

$$\ln z = \ln a + b \ln x + c \ln y,$$

and so $\ln z$ can be fit as a linear function of two variables $\ln x$ and $\ln y$.

2.6 Parameter Estimation Directly from Differential Equations

Modeling a new phenomenon, we often find ourselves in the following situation: we can write down a differential equation that models the phenomenon, but we cannot solve the differential equation analytically. We would like to solve it numerically, but all of our techniques thus far for parameter estimation assume we know the exact form of the function.

2.6.1 Derivative Approximation Method

Example 2.6. In population models, we often want to study the interaction between various populations. Probably the simplest interaction model is the *Lotka–Volterra* predator–prey model,

$$\begin{aligned} \frac{dx}{dt} &= ax - bxy \\ \frac{dy}{dt} &= -ry + cxy, \end{aligned} \tag{2.5}$$

where $x(t)$ represents the population of prey at time t and $y(t)$ represents the population of predators at time t. Observe that the interaction terms, -bxy and $+cxy$, correspond respectively with death of prey in the presence of predators and proliferation of predators in the presence of prey.

Though certainly instructive to study, the Lotka–Volterra model is typically too simple to capture the complex dynamics of species interaction. One famous example that it does model fairly well is the interaction between lynx (a type of wildcat) and hare (mammals in the same biological family as rabbits), as measured by pelts collected by the Hudson Bay Company between 1900 and 1920. Raw data from the Hudson Bay Company is given in Table 2.6.

Year	Lynx	Hare	Year	Lynx	Hare	Year	Lynx	Hare
1900	4.0	30.0	1907	13.0	21.4	1914	45.7	52.3
1901	6.1	47.2	1908	8.3	22.0	1915	51.1	19.5
1902	9.8	70.2	1909	9.1	25.4	1916	29.7	11.2
1903	35.2	77.4	1910	7.4	27.1	1917	15.8	7.6
1904	59.4	36.3	1911	8.0	40.3	1918	9.7	14.6
1905	41.7	20.6	1912	12.3	57.0	1919	10.1	16.2
1906	19.0	18.1	1913	19.5	76.6	1920	8.6	24.7

Table 2.6: Number of pelts collected by the Hudson Bay Company (in 1000's).

Our goal here will be to estimate values of a, b, r, and c without finding an exact solution to (2.5). Beginning with the predator equation, we first assume the predator population is not zero and re-write it as

$$\frac{1}{y}\frac{dy}{dt} = cx - r.$$

If we now treat the expression $\frac{1}{y}\frac{dy}{dt}$ as a single variable, we see that c and r are respectively the slope and intercept of a line. That is, we would like to plot values of $\frac{1}{y}\frac{dy}{dt}$ versus x and fit a line through this data. Since we have a table of values for x and y, the only difficulty in this is finding values of $\frac{dy}{dt}$. In order to do this, we first recall the definition of derivative,

$$\frac{dy}{dt}(t) = \lim_{h\to 0}\frac{y(t+h)-y(t)}{h}.$$

Following the idea behind Euler's method for numerically solving differential equations, we conclude that for h sufficiently small,

$$\frac{dy}{dt}(t) \cong \frac{y(t+h)-y(t)}{h},$$

which we will call the *forward difference* derivative approximation.

Critical questions become, how good an approximation is this and can we do better? To answer the first, we recall that the Taylor series for any function, $f(x)$, which admits a power series expansion, is given by

$$f(x) = \sum_{k=0}^{\infty}\frac{f^{(k)}(a)}{k!}(x-a)^k = f(a) + f'(a)(x-a) + \frac{f''(a)}{2}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \ldots.$$

(A precise statement of Taylor's theorem on polynomial expansion is included in the appendix.) Letting $x = t + h$ and $a = t$, we obtain the expansion,

$$f(t+h) = f(t) + f'(t)h + \frac{f''(t)}{2}h^2 + \frac{f'''(t)}{3!}h^3 + ...$$

Finally, we subtract $f(t)$ from both sides and divide by h to arrive at our approximation,

$$\frac{f(t+h) - f(t)}{h} = f'(t) + \frac{f''(t)}{2}h + \frac{f'''(t)}{3!}h^2 + ...,$$

from which we see that the error in our approximation is proportional to h. We will say, then, that the forward difference derivative approximation is an *order one* approximation, and write

$$f'(t) = \frac{f(t+h) - f(t)}{h} + \mathbf{O}(|h|); \quad t \text{ typically confined to some bounded interval, } t \in [a, b],$$

where $g(h) = \mathbf{O}(|h|)$ simply means that $|\frac{g(h)}{h}|$ remains bounded as $h \to 0$.

Our second question above was, can we do better? In fact, it's not difficult to show (see homework) that the *central difference* derivative approximation is second order:

$$f'(t) = \frac{f(t+h) - f(t-h)}{2h} + \mathbf{O}(h^2).$$

Returning to our data, we observe that h in our case will be 1, not particularly small. But keep in mind that our goal is to estimate the parameters, and we can always check the validity of our estimates by checking the model against our data. Since we cannot compute a central difference derivative approximation for our first year's data (we don't have $y(t-h)$), we begin in 1901, and compute

$$\frac{1}{y(t)}\frac{dy}{dt} \cong \frac{1}{y(t)}\frac{y(t+h) - y(t-h)}{2h} = \frac{1}{6.1}\frac{9.8 - 4.0}{2} = c \cdot 47.2 - r.$$

Repeating for each year up to 1919 we obtain the system of equations that we will solve by regression. In MATLAB, the computation becomes (The data vectors *H,* and *L* are defined in the M-file *lvdata.m*, available on [1].),

```
>>H=[30 47.2 70.2 77.4 36.3 20.6 18.1 21.4 22 25.4 27.1 ...
40.3 57 76.6 52.3 19.5 11.2 7.6 14.6 16.2 24.7];
>>L=[4 6.1 9.8 35.2 59.4 41.7 19 13 8.3 9.1 7.4 ...
8 12.3 19.5 45.7 51.1 29.7 15.8 9.7 10.1 8.6];
>>for k=1:19
Y(k)=(1/L(k+1))*(L(k+2)-L(k))/2;
X(k)=H(k+1);
end
>>plot(X,Y,'o')
```

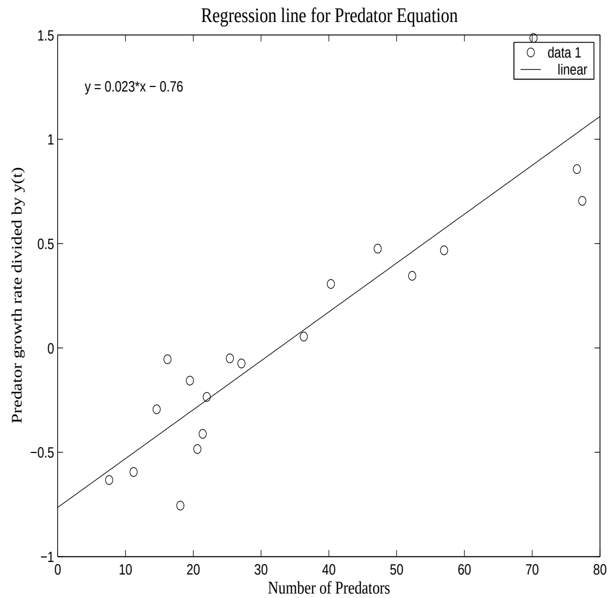

Figure 2.9: Linear fit for parameter estimation in prey equation.

From Figure 1, we can read off our first two parameter values $c = .023$ and $r = .76$. Proceeding similarly for the prey equation, we find $a = .47$ and $b = .024$, and define our model in the M-file *lv.m*.

```
function yprime = lv(t,y)
%LV: Contains Lotka-Volterra equations
a = .47; b = .024; c = .023; r = .76;
yprime = [a*y(1)-b*y(1)*y(2);-r*y(2)+c*y(1)*y(2)];
```

Finally, we check our model with Figure 2.10.

```
>>[t,y]=ode23(@lv,[0 20],[30 4]);
>>years=0:20;
>>subplot(2,1,1);
>>plot(t,y(:,1),years,H,'o')
>>subplot(2,1,2)
>>plot(t,y(:,2),years,L,'o')
```

2.6.2 Direct Method

The primary advantage of the derivative approximation method toward parameter estimation is that the computation time is typically quite reasonable. The primary drawback is that the derivative approximations we're forced to make are often quite crude. Here, we consider a second method of parameter estimation for ODE for which we carry out a regression argument directly with the ODE. In general, the method works as follows. (Warning: The

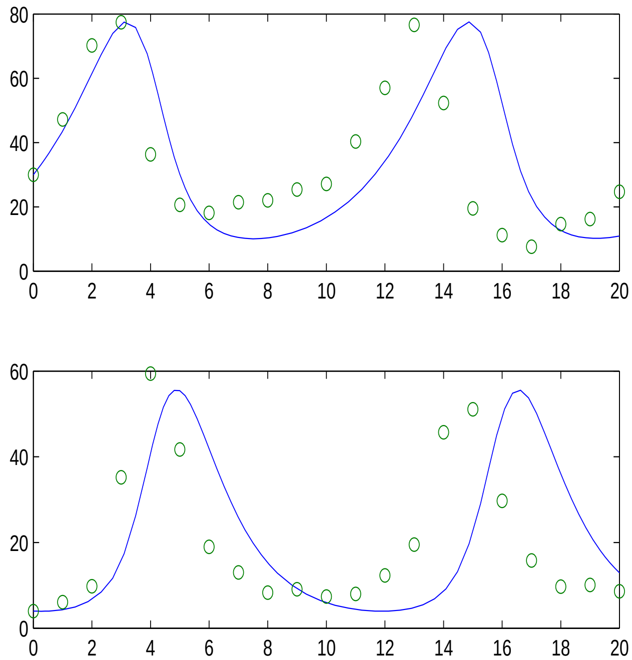

Figure 2.10: Model and data for Lynx–Hare example.

next couple of lines will make a lot more sense once you've worked through Example 2.7.) Suppose, we have a system of ODE

$$\mathbf{y}' = f(t, \mathbf{y}; \mathbf{p}), \quad \mathbf{y} \in \mathbb{R}^n, \mathbf{f} \in \mathbb{R}^n$$

(here, bold case denotes a vector quantitiy), where $\mathbf{p}$ is some vector of paramters $\mathbf{p} \in \mathbb{R}^m$, and a collection of k data points: $(t_1, \mathbf{y}_1), (t_2, \mathbf{y}_2), ..., (t_k, \mathbf{y}_k)$. In this notation, for example, the Lotka–Volterra model takes the form

$$\begin{aligned} y_1' =& p_1 y_1 - p_2 y_1 y_2 \\ y_2' =& -p_4 y_2 + p_3 y_1 y_2, \end{aligned}$$

where

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}; \quad \mathbf{p} = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{pmatrix}.$$

(Notice particularly that a subscript on a bold y indicates a data point, whereas a subscript on a normal y indicates a component.)

We can now determine optimal values for the parameters $\mathbf{p}$ by minimizing the error,

$$E(\mathbf{p}) = \sum_{j=1}^{k} |\mathbf{y}(t_j; \mathbf{p}) - \mathbf{y}_j|^2,$$

where $|\cdot|$ denotes standard Euclidean vector norm,

$$\left| \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} \right| = \sqrt{y_1^2 + y_2^2 + y_3^2}.$$

Example 2.7. Let's revisit the equations of Example 2.6 with this new method, so we can compare the final results. First, we write a MATLAB function M-file that contains the Lotka–Volterra model. In system form, we take $y_1(t)$ to be prey population and $y_2(t)$ to be predator population. We record the vector of parameters p as $p = (p(1), p(2), p(3), p(4)) = (a, b, c, r)$.

```
function value = lvpe(t,y,p)
%LVPE: ODE for example Lotka-Volterra parameter
%estimation example. p(1)=a, p(2) = b, p(3) = c, p(4) = r.
value=[p(1)*y(1)-p(2)*y(1)*y(2);-p(4)*y(2)+p(3)*y(1)*y(2)];
```

We now develop a MATLAB M-file that takes as input possible values of the parameter vector p and returns the squared error $E(p)$. Observe in particular here that in our *ode23()* statement we have given MATLAB a vector of times (as opposed to the general case, in which we give only an initial time and a final time). In this event, MATLAB returns values of the dependent variables only at the specified times.

```
function error = lverr(p)
%LVERR: Function defining error function for
%example with Lotka-Volterra equations.
lvdata
[t,y] = ode45(@lvpe,years,[H(1);L(1)],[],p);
value = (y(:,1)-H').^2+(y(:,2)-L').^2;
%Primes transpose data vectors H and L
error = sum(value);
```

Finally, we write an M-file *lvperun.m* that minimizes the function defined in *lverr.m* with MATLAB's built-in nonlinear minimizing routine *fminsearch()*. Observe that we take as our initial guess the values for a, b, c, and r we found through the derivative approximation method. Often, the best way to proceed with parameter estimation is in exactly this way, using the derivative approximation method to narrow the search and the direct method to refine it.

```
%LVPERUN: MATLAB script M-file to run Lotka-Volterra
%parameter estimation example.
guess = [.47; .024; .023; .76];
[p,error]=fminsearch(@lverr, guess)
%Here, error is our usual error function E
lvdata
[t,y]=ode45(@lvpe,[0,20],[30.0; 4.0],[],p);
subplot(2,1,1)
```

```
plot(t,y(:,1),years,H,'o')
subplot(2,1,2)
plot(t,y(:,2),years,L,'o')
```

The implementation is given below, and the figure created is given in Figure 2.11.

```
>>lvperun
p =
0.5486
0.0283
0.0264
0.8375
error =
744.7935
```

We find the parameter values $a = .5486$, $b = .0283$, $c = .0264$, and $r = .8375$.

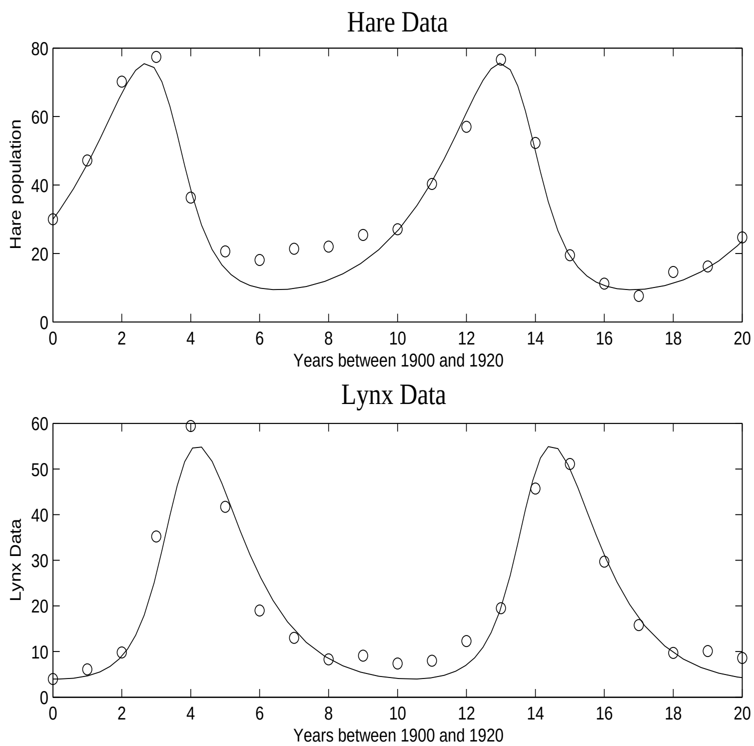

Figure 2.11: Refined model and data for Lynx–Hare example.

2.7 Parameter Estimation using Equilibrium Points

Another useful, though considerably less stable method for evaluating parameters involves a study of the equilibrium points for the problem (for a definition and discussion of equilibrium points, see the course notes *Modeling with ODE*).

Example 2.8. Suppose a certain parachutist is observed to reach a terminal velocity $v_T = -10$ m/s, and that his fall is modeled with constant gravitational acceleration and linear drag. Determine the drag coefficient b.

In this example, we are taking the model

$$y'' = -g - by',$$

which for velocity $v = y'$, can be written as the first order equation

$$v' = -g - bv.$$

By definition, terminal velocity is that velocity at which the force due to air resistance precisely balances the force due to gravity and the parachutist's velocity quits changing. That is,

$$0 = v' = -g - bv_T.$$

Put another way, the terminal velocity v_T is an equilibrium point for the system. We can now solve immediately for b as

$$b = \frac{9.81}{10} = .981 \text{ s}^{-1}.$$

3 Dimensional Analysis

In general, a *dimension* is any quantity that can be measured. If we want to compute the average velocity of an object, we typically measure two quantities: the distance the object traveled and the amount of time that passed while the object was in motion. We say, then, that the dimensions of velocity are length, L, divided by time, T, and write

$$\text{dimensions of velocity} = [v] = LT^{-1}, \tag{3.1}$$

where length and time are regarded as *fundamental* dimensions and velocity is regarded as a derived dimension. Notice in particular that (3.1) holds true regardless of the units we choose—feet, meters, etc. According to the International System of Units (SI),[6] there are seven fundamental dimensions, which we list in Table 3.1 along with their associated base units and their years of international adoption in their current form. (When discussing population dynamics, we will often refer to population as *biomass* and treat this is an eighth fundamental dimension, B.)

Typical physical quantities and their associated dimensions are listed in Table 3.2.

Remark on units. Units are the convention we take to measure a dimension. For example, we can measure length in meters, feet, astronomical units, or any of a number of other

[6] Taken from the French: Système Internationale d'Unitès.

Dimension	Base SI Unit	Year adopted
length L	meter (m)	1983
mass M	kilogram (kg)	1889
time T	second (s)	1967
temperature Θ	kelvin (K)	1967
electric current E	ampere (A)	1946
luminous intensity I	candela (cd)	1967
amount of substance A	mole (mol)	1971

Table 3.1: Fundamental dimensions and their base units.

Quantity	Dimensions
Length	L
Time	T
Mass	M
Velocity	LT^{-1}
Acceleration	LT^{-2}
Force	MLT^{-2}
Energy	ML^2T^{-2}
Momentum	MLT^{-1}
Work	ML^2T^{-2}

Quantity	Dimensions
Frequency	T^{-1}
Density	ML^{-3}
Angular momentum	ML^2T^{-1}
Viscosity	$ML^{-1}T^{-1}$
Pressure	$ML^{-1}T^{-2}$
Power	ML^2T^{-3}
Entropy	ML^2T^{-2}
Heat	ML^2T^{-2}
Momentum	MLT^{-1}

Table 3.2: Dimensions of common physical quantities.

units (1 astronomical unit (AU) is the average distance between the earth and sun, roughly 92.9×10^6 miles). The natural question that arises is, how do we determine the units for a given dimension. One of the first units was the *cubit*, the distance from the tip of a man's elbow to the end of his middle finger. This convention had the convenience that a person never found himself without a "cubit stick," but it suffered from the obvious drawback that every cubit was different. In the SI system, units are defined relative to some fixed measurable object or process. For example, in 1983 the meter was standardized as the length traveled by light in a vacuum in 1/299,792,458 seconds.[7] (I.e., the speed of light in vacuum is by definition $c = 299,792,458$ meters per second.)

In general, choosing a base unit can be tricky, and in order to gain an appreciation of this, we will consider the case of temperature. While we could fairly easily define and use our own length scale in the lecture hall (choose for example the cubit of any particular student), temperature would be more problematic. In general, we measure an effect of temperature rather than temperature itself. For example, the element mercury expands when heated, so we often measure temperature by measuring the height of a column of mercury. We might take a column of mercury, mark its height in ice and label that 0, mark its height in boiling water and label that 100, and evenly divide the distance between these two marks into units of measurement. (Though your scale would vary with atmospheric conditions;

[7] You probably don't want to know how seconds are defined, but here goes anyway, quoted directly from the source: a second is "the duration of 9,192,631,770 periods of the radiation corresponding to the transition between two hyperfine levels of the ground state of the cesium-133 atom." This won't be on the exam.

in particular, with variance in pressure.) The SI convention for temperature hinges on the observation that liquid water, solid ice, and water vapor can coexist at only one set of values for temperature and pressure. By international agreement (1967), the triple point of water is taken to correspond with a temperature of 273.16 K (with corresponding pressure 611.2 Pascals). We take 0 to be "absolute zero," the temperature at which molecules possess the absolute minimum kinetic energy allowed by quantum mechanics. (The Heisenberg Uncertainty Principle places a lower limit, greater than 0, on the kinetic energy of any molecule.) More succinctly, 1 Kelvin is precisely 1/273.16 of the (unique) temperature for the triple point of water.

3.1 Finding Simple Relations

Dimensional analysis can be an effective tool for determining basic relations between physical quantities.

Example 3.1. Suppose an object is fired straight upward from the earth with initial velocity v, where v is assumed small enough so that the object will remain close to the earth. Ignoring air resistance, we can use dimensional analysis to determine a general form for the time at which the object lands.

We begin by determining what quantities the final time will depend on, in this case only initial velocity and acceleration due to gravity, g. We write

$$t = t(v, g) \propto v^a g^b \Rightarrow T = L^a T^{-a} L^b T^{-2b},$$

which leads to the dimensions equations,

$$\begin{aligned} T:&\quad 1 = -a - 2b \\ L:&\quad 0 = a + b, \end{aligned}$$

from which we observe that $b = -1$ and $a = 1$. We conclude that $t \propto \frac{v}{g}$, where it's important to note that we have not found an exact form for t, only proportionality. In particular, we have $t = k\frac{v}{g}$ for some unknown dimensionless constant k. *This is as far as dimensional analysis will take us.* (We only obtained an exact form in Example 3.1 because the constant G is well known.) At this point, we should check our expression to insure it makes sense physically. According to our expression, the larger v is, the longer the object will fly, which agrees with our intuition. Also, the stronger g is, the more rapidly the object will descend.

Though in this case the constant of proportionality, k, is straightforward to determine from basic Newtonian mechanics, we typically determine proportionality constants experimentally. In this case, we would launch our object at several different initial velocities and determine k by the methods of Section 2. △

Example 3.2. Consider an object of mass m rotating with velocity v a distance r from a fixed center, in the absence of gravity or air resistance (see Figure 3.1). The *centripetal* force on the object, F_p, is the force required to keep the object from leaving the orbit. We can use dimensional analysis to determine a general form for F_p.

We begin by supposing that F_p depends only on the quantities m, r, and v, so that,

$$F_p = F_p(m, r, v) \propto m^a r^b v^c \Rightarrow MLT^{-2} = M^a L^b L^c T^{-c},$$

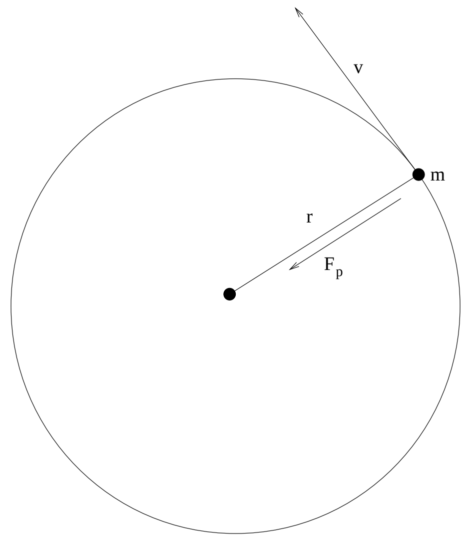

Figure 3.1: Centripetal force on a rotating object.

from which we obtain the dimensions equations,

$$\begin{aligned} M: &\quad 1 = a \\ L: &\quad 1 = b + c \\ T: &\quad -2 = -c. \end{aligned}$$

We have, then, $a = 1$, $c = 2$, and $b = -1$, so that

$$F_p = k\frac{mv^2}{r}.$$

△

Example 3.3. Given that the force of gravity between two objects depends on the mass of each object, m_1 and m_2, the distance between the objects, r, and Newton's gravitational constant G, where

$$[G] = M^{-1}L^3T^{-2},$$

we can determine Newton's law of gravitation. We begin by writing $F = F(m_1, m_2, r, G)$, which is simply a convenient way of expressing that the force due to gravity depends only on these four variables. We now guess that the relation is a simple multiple of powers of these variables and write

$$F(m_1, m_2, r, G) = m_1^a m_2^b r^c G^d.$$

(If this is a bad guess, we will not be able to find values for a, b, c, and d, and we will have to use the slightly more involved analysis outlined in later sections.) Recalling that the dimensions of force are MLT^{-2}, we set the dimensions of each side equal to obtain,

$$MLT^{-2} = M^a M^b L^c M^{-d} L^{3d} T^{-2d}.$$

Equating the exponents of each of our dimensions, we have three equations for our four unknowns:

$$\begin{aligned} M: &\quad 1 = a + b - d \\ L: &\quad 1 = c + 3d \\ T: &\quad -2 = -2d. \end{aligned}$$

We see immediately that $d = 1$ and $c = -2$, though a and b remain undetermined since we have more equations than unknowns. By symmetry, however, we can argue that a and b must be the same, so that $a = b = 1$. We conclude that Newton's law of gravitation must take the form

$$F = G\frac{m_1 m_2}{r^2}.$$

△

In practice the most difficult part of applying dimensional analysis can be choosing the right quantities of dependence. In Examples 3.1 through 3.3 these quantities were given, so let's consider an example in which they are less obvious.

Example 3.4. Determine a general form for the radius created on a moon or planet by the impact of a meteorite.

We begin by simply listing the quantities we suspect might be important: mass of the meteorite, m, density of the earth, ρ_e, volume of the meteorite, V_m, impact velocity of the meteorite, v, and gravitational attraction of the moon or planet, g (which affects how far the dirt is displaced). (In a more advanced model, we might also consider density of the atmosphere, heat of the meteorite, etc.) We see immediately that we're going to run into the problem of having three equations (one for each of M, L, and T) and five unknowns, m, ρ_e, V_m, v, and g, so in order to apply the method outlined in the previous examples, we will need to make some reductions. First, let's suppose we don't need to consider both the mass and volume of the meteorite and remove V_m from our list. Next, let's try to combine parameters. Noticing that m and v can be combined into kinetic energy ($\frac{1}{2}mv^2$), we can drop them and consider the new quantity of dependence E. Finally, we are prepared to begin our analysis. We have,

$$r = r(E, \rho_e, g) \propto E^a \rho_e^b g^c \Rightarrow L = M^a L^{2a} T^{-2a} M^b L^{-3b} L^c T^{-2c},$$

from which we obtain the dimensions equations,

$$\begin{aligned} M: &\quad 0 = a + b \\ L: &\quad 1 = 2a - 3b + c \\ T: &\quad 0 = -2a - 2c. \end{aligned}$$

Subsituting $a = -b$ into the second two equations, we find $a = \frac{1}{4}$, $b = -\frac{1}{4}$, and $c = -\frac{1}{4}$, so that

$$r = k(\frac{E}{\rho_e g})^{1/4}.$$

Again, we observe that the basic dependences make sense: higher energies create larger craters, while planets with greater density or gravitational pull receive smaller craters. (Consider, for example, craters on the moon as opposed to craters on the earth.) △

3.2 More General Dimensional Analysis

Example 3.5. Consider the following slight variation on the problem posed in Example 3.1: Suppose an object is fired straight upward from a height h above the earth, and use dimensional analysis to determine a basic form for the time at which the object strikes the earth. The only real difference here is that t now depends on h as well as v and g. Proceeding as before, we have

$$t = t(h, v, g) \propto h^a v^b g^c \Rightarrow T = L^a L^b T^{-b} L^c T^{-2c},$$

from which we obtain the dimensions equations,

$$\begin{aligned} T: &\quad 1 = -b - 2c \\ L: &\quad 0 = a + b + c. \end{aligned}$$

Since mass M does not appear in any of our quantities of dependence (and according to Galileo it shouldn't), we have two equations and three unknowns. We overcame a similar problem in Example 3.4 by dropping a quantity of dependence and by combining variables, but in general, and here in particular, we cannot reasonably do this.

Before introducing our more general method of dimensional analysis, let's see what's happening behind the scenes in Example 3.5. According to Newton's second law of motion, the height of our object at time t is given by

$$y(t) = -gt^2/2 + vt + h.$$

In order to find the time at which our object strikes the earth, we need only solve $y(t) = 0$, which gives

$$t = \frac{-v \pm \sqrt{v^2 + 2gh}}{-g}. \tag{3.2}$$

We have the right quantities of dependence; it's our assumption that t is a simple product of powers that breaks down.

Returning to the problem posed in Example 3.5, let's take a slightly different tack. Instead of beginning with the expression $t = t(h, v, g)$, we will begin now searching for *dimensionless products*,

$$\pi = \pi(h, v, g, t);$$

that is, variable combinations that have no dimension. (The designation of dimensionless products by π is standard, if perhaps unfortunate. Very generally, we are interested in dimensionless products because they are objects we can study without considering what units the problem might be posed in.) We have, then,

$$\pi = \pi(h, v, g, t) \propto h^a v^b g^c t^d \Rightarrow 1 = L^a L^b T^{-b} L^c T^{-2c} T^d,$$

from which we obtain the dimensions equations

$$\begin{aligned} T: &\quad 0 = -b - 2c + d \\ L: &\quad 0 = a + b + c. \end{aligned}$$

Since we have two equations and four unknowns, two of our unknowns will remain undetermined and can be chosen (we have two degrees of freedom). For example, we might choose $d = 1$ and $c = 0$, which determines $b = 1$ and $a = -1$. Our first dimensionless product becomes $\pi_1 = \frac{vt}{h}$. Alternatively, we can choose $d = 0$ and $c = 1$, which determines $b = -2$ and $a = 1$, making our second dimensionless product $\pi_2 = \frac{hg}{v^2}$. Finally, we will take $a = 1$ and $b = 1$, which determines $c = -2$ and $d = -3$, providing a third dimensionless product $\pi_3 = \frac{hv}{g^2t^3}$. Notice, however, that π_3 is nothing more than π_1^{-3} multiplied by π_2^{-2} ($\pi_3 = \pi_1^{-3}\pi_2^{-2} = \frac{h^3}{v^3t^3} \cdot \frac{v^4}{h^2g^2} = \frac{hv}{g^2t^3}$) and in this sense doesn't give us any new information. In fact, *any* other dimensionless product can be written as some multiplication of powers of π_1 and π_2, making them a *complete set* of dimensionless products. We will prove this last assertion below, but for now let's accept it and observe what will turn out to be the critical point in the new method: our defining equation for t (nondimensionalized by dividing by h),

$$-gt^2/(2h) + vt/h + 1 = 0,$$

can be rewritten entirely in terms of π_1 and π_2, as

$$-\pi_1^2\pi_2/2 + \pi_1 + 1 = 0.$$

Solving for π_1, we find

$$\pi_1 = \frac{-1 \pm \sqrt{1 + 2\pi_2}}{-\pi_2},$$

from which we conclude

$$t = \frac{h}{v} \cdot \frac{-1 \pm \sqrt{1 + 2\frac{hg}{v^2}}}{-\frac{hg}{v^2}},$$

which corresponds exactly with (3.2). *Notice that it was critical that our dependent variable t appeared in only one of π_1 and π_2.* Otherwise, we would not have been able to solve for it. △

Our general method of dimensional analysis hinges on the observation that we can always proceed as above. To this end, we have the following theorem.

Theorem 3.1. *(Buckingham's Theorem) If the dimensions for an equation are consistent and $\{\pi_1, \pi_2, ..., \pi_n\}$ form a complete set of dimensionless products for the equation, then there exists some function f so that the equation can be written in the form*

$$f(\pi_1, \pi_2, ..., \pi_n) = 0.$$

In Example 3.5, the function guaranteed by Buckingham's Theorem is

$$f(\pi_1, \pi_2) = -\pi_1^2\pi_2/2 + \pi_1 + 1,$$

though it should be stressed that in general all we know from Buckingham's Theorem is that such a function exists; we do not know the precise form of it. (If we did, we wouldn't need to use dimensional analysis.) Giving this function f, we now suppose that it defines a relationship between π_1 and π_2 of the form

$$\pi_1 = \phi(\pi_2),$$

where in Example 3.5

$$\phi(\pi_2) = \frac{-1 - \sqrt{1 + 2\pi_2}}{-\pi_2},$$

while more generally its form must be found from regression (see Example 3.6 below). Since we don't know the precise form of f from Buckingham's Theorem we cannot actually verify that such a ϕ exists, but we are somewhat justified in expecting it to exist by the Implicit Function Theorem, which is stated in the appendix in a form most useful for us.

Before proving Theorem 3.1, we will consider two further examples that illustrate its application.

Example 3.6. Suppose the radius of the shock wave for a certain explosion depends on time t, initial energy E, air density ρ, and air pressure p. Use dimensional analysis to determine a general form for the radius as a function of the other variables.

Observing that we have too many variables for the simple method, we search for dimensionless products

$$\pi = \pi(t, E, \rho, p, R) = t^a E^b \rho^c p^d R^e,$$

which gives

$$1 = T^a M^b L^{2b} T^{-2b} M^c L^{-3c} M^d L^{-d} T^{-2d} L^e.$$

Matching exponents, we have three equations.

$$\begin{aligned} T: \quad & 0 = a - 2b - 2d \\ M: \quad & 0 = b + c + d \\ L: \quad & 0 = 2b - 3c - d + e. \end{aligned}$$

In order to insure that our dependent variable appears in a simple form in π_1, we choose $e = 1$ and $d = 0$. This gives $c = 1/5$, $b = -1/5$, and $a = -2/5$, or

$$\pi_1 = \frac{\rho^{1/5} R}{t^{2/5} E^{1/5}}.$$

For π_2, we eliminate the dependent variable by choosing $e = 0$ and $d = 1$, from which we conclude $c = -3/5$, $b = -2/5$, and $a = 6/5$. We conclude

$$\pi_2 = \frac{t^{6/5} p}{E^{2/5} \rho^{3/5}}.$$

Of course, other choices of e and d (or other variables) give different dimensionless products. Some other choices for π_1 include

$$\pi_1 = \frac{R p^{1/3}}{E^{1/3}}$$

$$\pi_1 = \frac{R \rho^{1/2}}{t p^{1/2}}$$

while some other choices of π_2 include

$$\pi_2 = \frac{tp^{5/6}}{E^{1/3}\rho^{1/2}}$$
$$\pi_2 = \frac{E\rho^{3/2}}{t^3 p^{5/2}}$$
$$\pi_3 = \frac{E^{2/3}\rho}{t^2 p^{5/3}},$$

each of which can be raised to any power to give another equally valid π_2. For any choice of π_1 and π_2, we conclude from Buckinghan's Theorem that there exists some function f so that the relationship we're after can be written as

$$f(\pi_1, \pi_2) = 0$$

and from the Implicit Function Theorem that there (probably) exists some function ϕ so that

$$\pi_1 = \phi(\pi_2).$$

In terms of the first pair of dimensionless products given here, this becomes

$$R = \frac{E^{1/5}t^{2/5}}{\rho^{1/5}}\phi(\frac{t^{6/5}p}{E^{2/5}\rho^{3/5}}).$$

Finally, let's consider how we would use data to find the form of ϕ. In Table 3.3, values for time and shock radius are given for the first atomic explosion, conducted in New Mexico in 1945. The remaining variables were constant during this experiment: air density $\rho = 1.21$ kg/m^3, air pressure $p = 101300$ kg $\cdot$ m$^{-1} \cdot$ s^{-2}, and an initial energy $E = 9 \times 10^{13}$ kg $\cdot$ m$^2 \cdot$ s^{-2}.

t(s)	1	2	3	4	5	6	7	8	9
R (m)	584.7	804.7	912.9	1135.4	1207.7	1275.2	1356.2	1442.1	1503.8

Table 3.3: Data for Example 3.6.

In searching for the form of ϕ, we proceed by plotting values of π_1 on the vertical axis and values of π_2 on the horizontal axis, and using regression to obtain an approximate form for the resulting curve. The following MATLAB code creates Figure 3.2.

```
>>p=101300; rho=1.21; E=9e+13; t=1:9;
>>R=[584.7 804.7 912.9 1135.4 1207.7 1275.2 1356.2 1442.1 1503.8];
>>pi1=(rho.^(1/5).*R)./(t.^(2/5).*E.^(1/5))
pi1 =
0.9832 1.0255 0.9892 1.0966 1.0668 1.0472 1.0471 1.0555 1.0500
>>pi2=(t.^(6/5).*p)./(E.^(2/5).*rho.^(3/5))
pi2 =
0.2367 0.5438 0.8847 1.2494 1.6331 2.0325 2.4455 2.8704 3.3062
>>plot(pi2,pi1,'o')
```

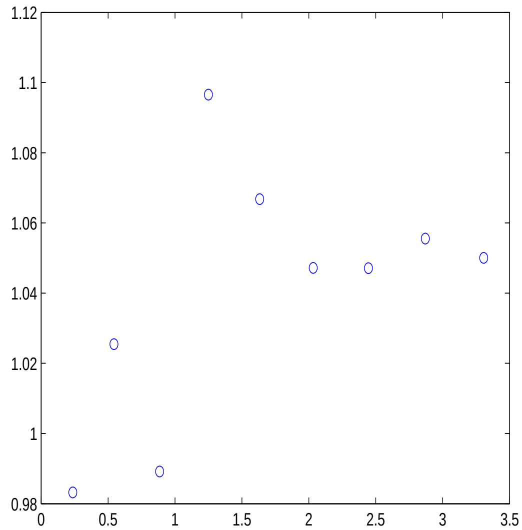

Figure 3.2: Plot of π_1 versus π_2 for shock wave data.

At first glance, this scatterplot might not seem to indicate much of a relationship, but observe that the values of π_1 (on the vertical axis) are all clustered near $\pi_1 = 1.05$, and so it appears that π_1 is either constant or has a very small fit. If we carry out a linear fit for this data, we find

$$\pi_1 = .017\pi_2 + 1.$$

That is, $\phi(\pi_2) = .017\pi_2 + 1$. Returning to our original variables, we have

$$\frac{\rho^{1/5}R}{t^{2/5}E^{1/5}} = .017(\frac{t^{6/5}p}{E^{2/5}\rho^{3/5}}) + 1,$$

or

$$R = \frac{t^{2/5}E^{1/5}}{\rho^{1/5}}\Big[.017(\frac{t^{6/5}p}{E^{2/5}\rho^{3/5}}) + 1\Big].$$

According to images declassified in 1947 the radius of that explosion after .006 seconds was 80 meters. As a final calculation, we will compare this figure with a prediction according to our model. In MATLAB the only value we need to change is t. We have

```
>>t=.006;
>>pi2=(t.^(6/5).*p)./(E.^(2/5).*rho.^(3/5))
pi2 =
5.1053e-04
>>R=(t^(2/5)*E^(1/5)/rho^(1/5))*(.017*pi2+1)
R =
76.8344
```

Note finally that while it would have been much easier to make this prediction by simply plotting R as a function of t, we would not in that case have developed the general relationship involving the dependence of R on all relevant variables. △

Before considering our final example, we review the steps of our general method for dimensional analysis.

1. Identify the variables of dependence.

2. Determine a complete set of dimensionless products, $\{\pi_1, \pi_2,...,\pi_n\}$, making sure that the dependent variable appears in only one, say π_1.

3. Apply Buckingham's Theorem to obtain the existence of a (typically unknown) function f satisfying
$$f(\pi_1, \pi_2, ..., \pi_n) = 0.$$

4. Apply the Implicit Function Theorem to obtain the existence of a (typically unknown) function ϕ presumably satisfying
$$\pi_1 = \phi(\pi_2, \pi_3, ..., \pi_n).$$

5. Solve the equation from Step 4 for the dependent variable and use experimental data to determine the form for ϕ.

Example 3.7. How long should a turkey be roasted?

The variables that cooking time should depend on are (arguably): size of the turkey, measured by characteristic length r, the initial temperature of the turkey, T_t, the temperature of the oven (assumed constant) T_o, and the *coefficient of heat conduction* for the turkey, k ($[k] = L^2T^{-1}$). (If u denotes the turkey's temperature, then the flow of heat energy per unit time per unit area is $-c\rho k u_x$, where c is the specific heat of the turkey and ρ is the turkey's density.) Typically, temperature is measured in *Kelvin*s, an SI (metric) unit which uses as its base the temperature at which all three phases of water can exist in equilibrium (see remarks on units at the beginning of this section). Since we are already using T to denote time, we will use Θ as our dimension for temperature. Our dimensionless products take the form
$$\pi = \pi(r, T_t, T_o, k, t) = r^a T_t^b T_o^c k^d t^e \Rightarrow 1 = L^a \Theta^b \Theta^c L^{2d} T^{-d} T^e,$$
from which we obtain the dimensions equations
$$\begin{aligned} L:&\quad 0 = a + 2d \\ \Theta:&\quad 0 = b + c \\ T:&\quad 0 = -d + e. \end{aligned}$$
As in Example 3.6, we have three equations and five unknowns and require two dimensionless products. Taking $e = 1$ and $b = 0$, we find $c = 0$, $d = 1$, and $a = -2$, so that our first dimensionless product is $\pi_1 = \frac{kt}{r^2}$. On the other hand, taking $e = 0$ and $b = 1$, we find $c = -1$, $d = 0$, and $a = 0$, so that our second dimensionless product is $\pi_2 = \frac{T_t}{T_o}$. According to Buckingham's theorem, there exists a function f so that
$$f(\pi_1, \pi_2) = 0,$$

and by the Implicit Function Theorem another function ϕ so that

$$\pi_1 = \phi(\pi_2).$$

We conclude that

$$t = \frac{r^2}{k}\phi(\frac{T_t}{T_o}).$$

$\triangle$

3.3 A Proof of Buckingham's Theorem

The proof of Buckingham's Theorem depends on an application of linear algebra, so let's first consider the general technique for solving the types of systems of equations that arose in Examples 3.1 through 3.7. Recalling Example 3.5, we have the system

$$\begin{aligned} T: &\quad 0 = -b - 2c + d \\ L: &\quad 0 = a + b + c, \end{aligned}$$

which we now re-write in matrix form

$$\begin{pmatrix} 0 & -1 & -2 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{3.3}$$

We proceed now by standard Gauss–Jordon elimination. The *augmented matrix* for this system is

$$\left(\begin{array}{cccc|c} 0 & -1 & -2 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{array}\right),$$

and only three row operations are required: we swap rows, then add the new Row 2 to the new Row 1, and finally multiply Row 2 by -1, giving

$$\left(\begin{array}{cccc|c} 1 & 0 & -1 & 1 & 0 \\ 0 & 1 & 2 & -1 & 0 \end{array}\right),$$

which is typically referred to as *reduced row echelon form* (RREF). We conclude that with c and d chosen arbitrarily, we require only

$$\begin{aligned} a &= c - d \\ b &= -\, 2c + d. \end{aligned}$$

Choosing $d = 1$ and $c = 0$ determines $a = -1$ and $b = 1$, while choosing $d = 0$ and $c = 1$ determines $a = 1$ and $b = -2$, so that we have two solutions to equation (3.3),

$$V_1 = \begin{pmatrix} -1 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \quad \text{and} \quad V_2 = \begin{pmatrix} 1 \\ -2 \\ 1 \\ 0 \end{pmatrix}.$$

Accordingly, *any* solution to equation (3.3) can be written as a *linear combination* of V_1 and V_2, $V = c_1V_1 + c_2V_2$, where c_1 and c_2 are scalars.

Recalling, now, Example 3.5, we observe that V_1 corresponds with the dimensionless product π_1, while V_2 corresponds with the dimensionless product π_2. Given any new dimensionless product π_3 (for example, $\pi_3 = \frac{hv}{g^2t^3}$), there corresponds a V_3 (e.g., $V_3 = (1, 1, -2, -3)^{\text{tr}}$) so that $V_3 = c_1V_1 + c_2V_2$ (in the example, $c_1 = -3$ and $c_2 = -2$). Consequently, π_3 can be written as $\pi_3 = \pi_1^{c_1}\pi_2^{c_2}$ $(= \pi_1^{-3}\pi_2^{-2})$, which establishes that π_1 and π_2 do indeed form a complete set of dimensionless products for the governing equation of Example 3.5. This means that every expression in the governing equation of Example 3.5 can be written as a product of powers of π_1 and π_2 and consequently that a function f must exist so that $f(\pi_1, \pi_2) = 0$.

In the general proof of Buckingham's Theorem, we will proceed by extending our exponent vectors (in our example, V_1 and V_2) to a full basis of $\mathbb{R}^k$, where k represents the number of physical quantities under consideration. In the context of Example 3.5, we have four physical quantities: time, height, velocity, and acceleration due to gravity, and we take $k = 4$. In this case V_1 and V_2 can readily be extended to the full basis for $\mathbb{R}^4$:

$$V_1 = \begin{pmatrix} -1 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \quad V_2 = \begin{pmatrix} 1 \\ -2 \\ 1 \\ 0 \end{pmatrix}, \quad V_3 = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \quad V_4 = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}.$$

We have, then, the two dimensionless products $\pi_1 = \frac{vt}{h}$ and $\pi_2 = \frac{hg}{v^2}$, corresponding with V_1 and V_2, and two additional products, not necessarily dimensionless, corresponding with V_3 and V_4: in this case, $\tilde{\pi}_3 = h$ and $\tilde{\pi}_4 = v$. We observe now that each physical quantity in Example 3.5 can be written in terms of our four products π_1, π_2, $\tilde{\pi}_3$, and $\tilde{\pi}_4$. This is a direct consequence of the fact that V_1, V_2, V_3, and V_4 form a full basis for $\mathbb{R}^4$, and this case can be easily verified: $h = \tilde{\pi}_3$, $v = \tilde{\pi}_4$, $t = \pi_1\tilde{\pi}_3\tilde{\pi}_4^{-1}$ and $g = \pi_2\tilde{\pi}_3^{-1}\tilde{\pi}_4^{-2}$. Since we can write each physical quantity in terms of the products, we must be able to write the underlying physical law in terms of the products. Suppose we do this with our example. Substituting directly into our governing equation

$$-\frac{g}{2h}t^2 + \frac{vt}{h} + 1 = 0,$$

we find

$$-\frac{\pi_2\tilde{\pi}_3^{-1}\tilde{\pi}_4^{-2}}{2\tilde{\pi}_3}(\pi_1\tilde{\pi}_3\tilde{\pi}_4^{-1})^2 + \frac{\tilde{\pi}_4\pi_1\tilde{\pi}_3\tilde{\pi}_4^{-1}}{\tilde{\pi}_3} + 1 = 0.$$

Cancelling terms, we see that all the $\tilde{\pi}_3$ and $\tilde{\pi}_4$ cancel, leaving our equation written entirely in terms of π_1 and π_2. In fact, the extra $\tilde{\pi}$ *must* always cancel, because since the other products are dimensionless, we could change units without affecting them, but while affecting the $\tilde{\pi}$.

General proof of Buckingham's Theorem. Let $x_1, x_2, ..., x_k$ denote the physical quantities under consideration and define a (one-to-one) function on their products by

$$h(x_1^{a_1}x_2^{a_2}\cdots x_k^{a_k}) = (a_1, a_2, ..., a_k).$$

Similarly, define a second function ϕ (n is the number of fundamental dimensions, usually for us $n = 3$: M, L, T)

$$\phi(x_1^{a_1}x_2^{a_2}\cdots x_k^{a_k}) = (d_1, d_2, ..., d_n),$$

where d_1, d_2, ..., d_n are the powers on the fundamental dimensions (e.g., for $n = 3$, $M^{d_1}L^{d_2}T^{d_3}$). Consider, now, the map $\phi h^{-1} : \mathbb{R}^k \to \mathbb{R}^n$. Let $\{b_1, b_2, ..., b_j\}$ be a basis for the null space of ϕh^{-1} ($\phi h^{-1}(b_m) = 0_n$ for all $m = 1, ..., j$), and extend it (if necessary) to a basis for $\mathbb{R}^k$, $\{b_1, b_2, ..., b_k\}$. Observe that the first j elements of this basis correspond with dimensionless products (they have been chosen so that the powers on the dimenions are all 0). Define $\pi_m = h^{-1}(b_m), m = 1, ..., k$, and observe that since $\{b_1, b_2, ..., b_k\}$ forms a basis for $\mathbb{R}^k$, any vector $(a_1, a_2, ..., a_k)$ can be written as a linear combination of the b_m. Notice also that $h(\pi_1^{i_1}\pi_2^{i_2}\cdots\pi_k^{i_k}) = i_1h(\pi_1) + i_2h(\pi_2) + ... + i_kh(\pi_k) = i_1b_1 + i_2b_2 + ... + i_kb_k$. In particular, the vector $(a_1, a_2, ..., a_k) = (1, 0, ..., 0) = \sum_{l=1}^k c_lb_l$, so that $x_1 = h^{-1}(\sum_{l=1}^k c_lb_l) = h^{-1}((\sum_{l=1}^k c_lh(\pi_l)) = \pi_1^{c_1}\pi_2^{c_2}\cdots\pi_k^{c_k}$, and similarly for $x_2, ..., x_k$ so that the x_m can all be written as products of powers of the π_m. Hence, any equation that can be written in terms of the x_m can be written in terms of the π_m. Finally, we must resolve the issue that for $m > j$, π_m is *not* dimensionless. For these π_m there exist changes of units (meters to feet etc.) which change π_m but do not change the $\pi_1, ..., \pi_j$. But we have assumed that our physical law is independent of units, and so it cannot depend on the π_m. $\square$

3.4 Nondimensionalizing Equations

A final useful trick associated with dimensional analysis is the *nondimensionalization of equations*: writing equations in a form in which each summand is dimensionless. Through nondimensionalization, we can often reduce the number of parameters in an equation and consequently work with the most "bare bones" version of the model.

Example 3.8. Consider a projectile traveling under the influences of gravity and linear air resistance. According to Newton's second law of motion, a projectile traveling vertically under these forces has height $y(t)$ given by

$$y''(t) = -g - by'(t), \tag{3.4}$$

where the coefficient of air resistance, b, has units T^{-1}. Letting A and B represent dimensional constants to be chosen, we define the nondimensional variables $\tau = \frac{t}{A}$ and $Y(\tau) = \frac{y(t)}{B}$. We calculate

$$y'(t) = B\frac{d}{dt}Y(\tau) = B\frac{dY}{d\tau}\frac{d\tau}{dt} = \frac{B}{A}Y'(\tau)$$

and similarly

$$y''(t) = \frac{B}{A^2}Y''(\tau)$$

so that (3.4) can be re-written as

$$\frac{B}{A^2}Y''(\tau) = -g - b\frac{B}{A}Y'(\tau),$$

or

$$Y''(\tau) = -g\frac{A^2}{B} - bAY'(\tau).$$

Choosing finally

$$A = \frac{1}{b} \quad \text{and} \quad B = \frac{g}{b^2},$$

we obtain the reduced dimensionless equation

$$Y''(\tau) = -1 - Y'(\tau).$$

$\triangle$

Example 3.9. Consider an extension of Example 3.8 to the case in which gravitational acceleration is not considered constant. In this case, according to Newton's second law of motion, our model becomes

$$y''(t) = -\frac{GM_e}{(R+y)^2} - by',$$

where G denotes Newton's gravitational constant, R denotes the radius of the earth (we take the mean radius in practice), and M_e denotes the mass of the earth. Proceeding as in Example 3.8 with $\tau = \frac{t}{A}$ and $Y(\tau) = \frac{y(t)}{B}$, we find

$$Y''(\tau) = -\frac{GM_e}{(R+BY(\tau))^2}\frac{A^2}{B} - bAY'(\tau).$$

In this case, we cannot eliminate all four of the constants in our problem, but we can reduce the number of parameters to one by taking

$$A = b^{-1}; \quad B = \frac{GM_e}{b^2R^2}.$$

We find

$$Y''(\tau) = -\frac{1}{(1+\frac{GM_e}{b^2R^3}Y(\tau))^2} - Y'(\tau).$$

Observing that the dimensionless combination $\frac{GM_e}{b^2R^3}$ will typically be small, we denote it by ϵ, and conclude the dimensionless equation

$$Y''(\tau) = -\frac{1}{(1+\epsilon Y(\tau))^2} - Y'(\tau).$$

Example 3.10. Establish a dimensionless form for the Lotka–Volterra predator–prey model,

$$\begin{aligned}\frac{dx}{dt} &= ax - bxy\\ \frac{dy}{dt} &= -ry + cxy.\end{aligned}$$

First, for population dynamics, we require a new dimension, typically referred to as *biomass,* B ($[x] = B$, $[y] = B$). Assuming the Lotka–Volterra model is dimensionally consistent, we can now determine the dimensions of each parameter:

$$[\frac{dx}{dt}] = [a][x] - [b][x][y] \Rightarrow BT^{-1} = [a]B - [b]B^2 \Rightarrow [a] = T^{-1} \text{ and } [b] = T^{-1}B^{-1}.$$

Similarly, $[r] = T^{-1}$ and $[c] = T^{-1}B^{-1}$. Now, we search for dimensionless variables,

$$\tau = \frac{t}{T}, \quad X(\tau) = \frac{x(t)}{B_1}, \quad Y(\tau) = \frac{y(t)}{B_2},$$

with

$$\begin{aligned}\frac{d}{d\tau}X(\tau) =& \frac{T}{B_1}x'(\tau T) = \frac{T}{B_1}(ax(\tau T) - by(\tau T)x(\tau T))\\ =& \frac{T}{B_1}(aB_1X(\tau) - bB_1B_2X(\tau)Y(\tau)) = TaX(\tau) - bB_2TX(\tau)Y(\tau),\end{aligned}$$

and similarly,

$$Y'(\tau) = -rTY(\tau) - bB_2TX(\tau)Y(\tau).$$

In this case, we have four parameters and only three scalings, so we will not be able to eliminate all the paramters we did in Example 3.9. We can, however, eliminate three. To this end, we choose $T = a^{-1}$ and $B_2 = ab^{-1}$ to eliminate both parameters from the $X(\tau)$ equation, and we choose $B_1 = ac^{-1}$ to arrive at the dimensionless system,

$$\begin{aligned}X' =& X - XY\\ Y' =& -\frac{r}{a}X + XY,\end{aligned}$$

where $k := -\frac{r}{a}$ becomes our single dimensionless parameter. △

Appendix

One of the most useful theorems from calculus is the Implict Function Theorem, which addresses the question of existence of solutions to algebraic equations. Instead of stating its most general version here, we will state exactly the case we use.

Theorem A.1. (Implicit Function Theorem) Suppose the function $f(x_1, x_2, ..., x_n)$ is C^1 in a neighborhood of the point $(p_1, p_2, ..., p_n)$ (the function is continuous at this point, and its derivatives with respect to each variable are also continuous at this point). Suppose additionally that

$$f(p_1, p_2, ..., p_n) = 0$$

and

$$\partial_{x_1} f(p_1, p_2, ..., p_n) \neq 0.$$

Then there exists a neighborhood N_p of $(p_2, p_3, ..., p_n)$ and a function $\phi : N_p \to \mathbb{R}$ so that

$$p_1 = \phi(p_2, p_3, ..., p_n),$$

and for every $x \in N_p$,

$$f(\phi(x_2, x_3, ..., x_n), x_2, x_3, ..., x_n) = 0.$$

Another fundamental theorem of applied mathematics is the Taylor theorem, whereby information at a single point can provide information about a function on an entire set.

Theorem A.2. (Taylor expansion with remainder) Suppose $f(x)$ and its first n derivatives are continuous for $x \in [a, b]$, and suppose the $(n+1)$st derivative $f^{(n+1)}(x)$ exists for $x \in (a, b)$. Then there is a value $X \in (a, b)$ so that

$$f(b) = f(a) + f'(a)(b-a) + ... + \frac{f^{(n)}(a)}{n!}(b-a)^n + \frac{f^{(n+1)}(X)}{(n+1)!}(b-a)^{n+1}.$$

Theorem A.3. Suppose $f(x, y)$ is differentiable in both independent variables on some domain $D \subset \mathbb{R}^2$ and that (a, b) is an interior point of D for which

$$f_x(a, b) = f_y(a, b) = 0.$$

If moreover f_{xx}, f_{xy}, f_{yx}, and f_{yy} exist at the point (a, b), then the nature of (a, b) can be determined from $A = f_{xx}(a, b)$, $B = f_{xy}(a, b)$, and $C = f_{yy}(a, b)$ as follows:

1. If $B^2 - AC < 0$ and $A > 0$, then f has a relative minimum at (a, b);
2. If $B^2 - AC < 0$ and $A < 0$, then f has a relative maximum at (a, b);
3. If $B^2 - AC > 0$, then f has neither a maximum nor a minimum at (a, b);
4. If $B^2 - AC = 0$, further analysis is required, and any of 1–3 remain possible.

References

[1] http://www.math.tamu.edu˜/phoward/M442.html

Index