Modeling Basics

Peter Howard

Spring 2022

Contents

1 Overview

All modeling projects begin with the identification of a phenomenon 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 phenomenon to be modeled, and (2) collect and analyze 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 have to spend quite a bit of time learning enough biology to get started.

These notes address two important tools in the process of mathematical modeling, least squares regression and dimensional analysis. For the former, the vast majority of mathematical models that arise in practice include unspecified parameters, and the determination of values for these parameters is generally accomplished with least squares regression. The first half of these notes will comprise a discussion of this topic. In the second half of these notes, we will discuss dimensional analysis, which is a collection of methods for understanding physical events and processes based solely on the dimensions of the quantitites involved. In the end, we will see that the two methods are often used together, and form a general and powerful approach that can be productively employed in a wide range of applications.

2 Linear Least Squares Regression

Although the method of least squares regression is typically attributed to the German mathematician Carl Friedrich Gauss (1777-1855), the terminology *regression* was coined by the British anthropologist Francis Galton (1822-1911). Galton first used it to describe studies he was conducting on the correlation between the heights of offspring and the heights of progenitors (i.e., parents). His first study in this regard involved sweet peas (1877), but

we'll discuss it in terms of his more famous analysis of human height (1886)[1]: predicting the height of a child based on the heights of the parents.

In order to combine parent heights into a single height, Galton defined the *midheight*[2] to be

$$\text{midheight} = \frac{\text{father's height} + 1.08 \times \text{mother's height}}{2}.$$

Here, the factor 1.08 was introduced to match the data, and in computing the child heights Galton multiplied the height of each daughter by 1.08.[3] Galton's goal was to identify a functional relation of the form

$$\text{child height} = f(\text{midheight}),$$

and his expectation was that f would have a linear form so that

$$\text{child height} = m \times \text{midheight} + b,$$

for some slope m and y-intercept b. Galton's data is available at the following web site, developed by Kyle Siegrist, at the University of Alabama, Huntsville:

http://www.randomservices.org/random/

Once the data has been downloaded in *.csv* form[4], it can be processed with the following M-file (*galton_heights.m*), which writes the data as a set of vectors that can easily be manipulated with MATLAB. (The *.csv* file has been saved as *Galton_Data.csv*.)

```
%GALTON_HEIGHTS: MATLAB script M-file that defines
%Galton's original data from the .csv file
%Galton_Data.csv.
%First, import the data as a table
T=readtable('Galton_Data.csv');
%NOTE: On some systems, entries in Galton_Data.csv need
%to be converted from strings to double precision values.
%For this, exchange the specifications for mother, father,
%and child with its alternative line in the code below.
%Mother heights
mother = T.Mother;
%mother = str2double(T.Mother);
```

[1]F. Galton, *Regression toward mediocrity in hereditary stature*, Anthropological miscellanea, 1886, 246-263.

[2]Galton's actual terminology was *"mid-parental" height*, where the quotes are his.

[3]In his article, Galton writes: "In every case I transmuted the female statures to their corresponding male equivalents and used them in their transmuted form, so that no objection grounded on sexual difference of stature need be raised when I speak of averages. The factor I used was 1.08, which is equivalent to adding a little less than one-twelfth to each female height. It differs a little from the factors employed by other anthropologists, who, moreover, differ a trifle between themselves; anyhow, it suits my data better than 1.07 or 1.09."

[4]I.e., *comma-separated values* form, a rudimentary form of data file in which entries are separated by (you guessed it) commas.

```
%Father heights
father = T.Father;
%father = str2double(T.Father);
%Child heights
child = T.Height;
%child = str2double(T.Height);
%Gender
gender = T.Gender;
%Galton multiplied the height of each female
%child by 1.08. We can accomplish this as follows:
for k=1:length(gender)
if isequal(gender(k),{'F'}) == 1
child(k) = 1.08*child(k);
end
end
midheight = (father+1.08*mother)/2;
```

In order to begin analyzing this data, we plot it as points with

```
>>plot(midheight,child,'o')
```

This command creates Figure 2.1.

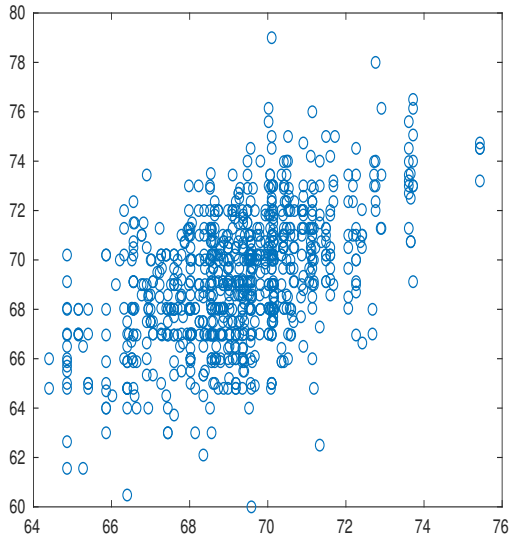


Figure 2.1: Scatter plot of Francis Galton's original data from 1886.

This arguably looks more like a blob than a line, but nonetheless we can fit it with the "best possible" line. Below, we will see how to find such a line analytically, but for this example, let's find it with MATLAB. For this, we can go to the figure window and select **Tools**, **Basic Fitting**, and then select the **linear** option. If we then click on the black

arrow at the bottom right of the pop-up menu, we will see that MATLAB has carried out the linear fit and found $m = .729$ and $b = 18.767$. I.e., our model for predicting the height of a child is

$$\text{child height} = .729 \times \text{midheight} + 18.767,$$

where for daughters we subsequently need to divide by 1.08.[5] This gives us Figure 2.2.

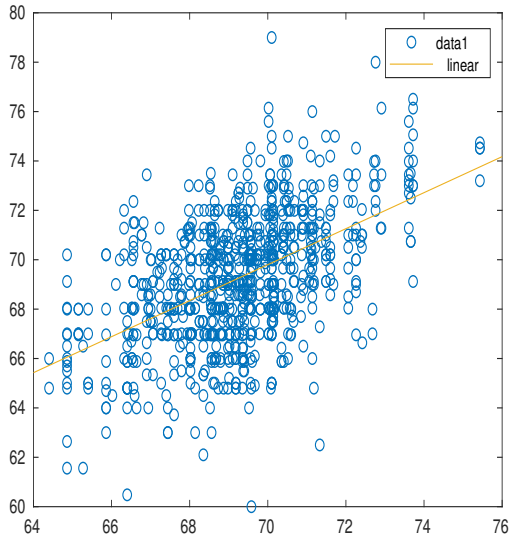


Figure 2.2: Scatter plot and "best fit" line for Galton's 1886 data.

Before exiting the fitting window, we can choose **Save to Workspace**, and let's save the fit as *galton*, which by default will be a MATLAB structure with two fields. We can access these in the Command Window as follows:

```
>>galton.type
ans =
'polynomial degree 1'
>>galton.coeff
ans =
0.7291 18.7670
```

Working with the plot window allows us to visually evaluate the data, but in some cases we already know that we will fit the data with a line, and for this we can more quickly use the MATLAB built-in function *polyfit.m*. For Galton's data the command and output is as follows:

```
>>polyfit(midheight,child,1)

ans =
```

[5] While Galton's view of the data was the same as ours, his approach to identifying this best-fit line was more visual, and he reported a slope of exactly $\frac{2}{3}$ and an intercept of exactly 22.75.

0.7291 18.7670

At this point, let's pause to think about where the terminology *regression* comes from. For this, consider the Figure 2.3, in which Galton's data has been omitted, and his model has been plotted along with the line $y = x$, which appears in red.

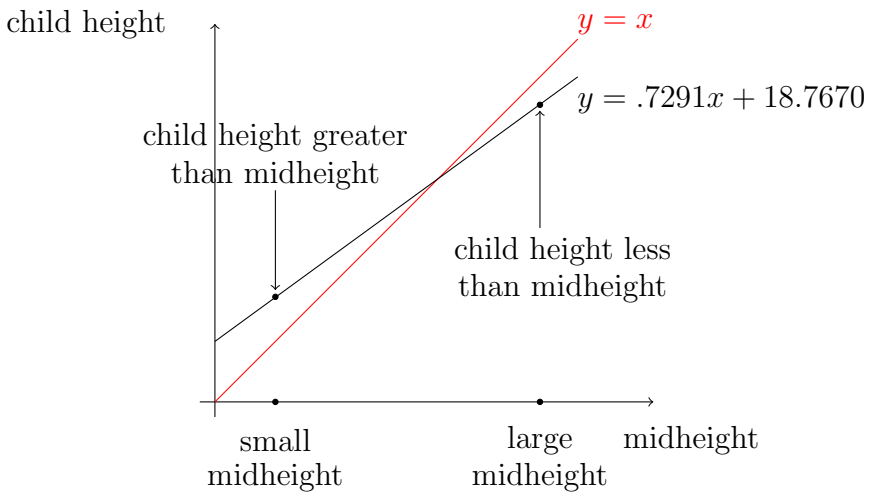


Figure 2.3: "Regression toward mediocrity" for Galton's fit.

We see that if the midheight is relatively small then the child's height will be greater than the midheight, while if the midheight is relatively large, the child's height will be less than the midheight. Galton referred to this as "regression towards mediocrity," often described now as "regression toward the mean." Although this method has come to be known as regression, it's clear that not all least-squares fits correspond with this dynamic. In particular, the key point for the above argument is that the slope of the fit is on the interval $(0, 1)$. Nonetheless, this dynamic of regression toward the mean is quite common. As another example,[6] studies show that businesses regress toward the mean in the following way: if we take a collection of businesses from the same sector (e.g., restaurants, retail stores, convenience stores etc.) and try to predict their success in 2022 based on their success in 2012, we often find that the level of success regresses toward the mean. I.e., the businesses that were most successful in 2012 will tend to be a bit less successful in 2022 (though possibly still well above average), and the businesses that were least successful in 2012 will tend to be a bit more successful in 2022. Why does this happen?

Galton's explanation in the case of heights ran as follows. He argued that height was partly inherited from the parents and partly (though to a lesser extent) inherited from previous ancestors (grandparents, great grandparents etc.). For the ancestor portion, he writes, "Speaking generally, the further his [i.e., the child's] genealogy goes back, the more numerous and varied will his ancestry become, until they cease to differ from any equally numerous sample taken at haphazard from the race at large. Their mean stature will then

[6]Taken from Jordan Ellenberg's boldly entitled book *How Not to Be Wrong: The Power of Mathematical Thinking*, Penguin Press 2014.

be the same as that of the race; in other words, it will be mediocre." In his view, if the midheight was large, the child's height would generally be reduced from this large value by the contribution from his ancestry, and the opposite would happen if the midheight was small. Generally, we might expect this phenomenon to occur whenever a trait depends on a combination of determined and random factors (the parent heights viewed as determined, the ancestral heights viewed as a random mix of the heights of countless ancestors). Returning to our example of businesses, while planning and decision-making are fairly deterministic matters, a certain amount of success of businesses is determined by luck. The most successful businesses in 2012 likely benefited from both good decision-making and good fortune, and while the decision-making may well have continued through 2022, there's no reason to expect that the luck would. On average, then, it stands to reason that these businesses won't be quite as successful as they were in 2012, leading to the observed regression.

So, why aren't we all the same height? That is, if heights regress toward the mean, and this has been going on for hundreds of thousands of years (homo sapiens date back about 300,000 years), why does there continue to be variability in height? Perhaps the easiest way to understand this is to imagine a population in which all midheights had indeed become uniform, so that every child was predicted to have the same height. In this case, the natural variability from the ancestry (not to overly discount the natural variability in direct heredity and enviroment) would naturally lead to variation in heights of the next generation. If we tried to capture this in a line fit of the data, something very strange would happen: since the midheights are all the same, the data would lie on a vertical line! If we return with this idea to our original discussion, we see that our development of regression depended on the assumption that we had a fairly robust spread of midheights. In this way, we can view the random aspects of a process as an outward pressure, pushing traits away from uniformity. To end this discussion with a final quote from Galton, he writes, "The answer is that the process comprises two opposite sets of actions, one concentrative and the other dispersive, and of such a character that they necessarily neutralize one another, and fall into a state of stable equilibrium."

2.1 Regression for Lines

In order to begin understanding how linear least squares regression works, let's suppose we have a set of data points $\{(x_k, y_k)\}_{k=1}^N$, and that we would like to identify a line $y = mx + b$ that best identifies its trend. (See Figure 2.4, drawn with $N = 3$.)

In Figure 2.4, the vertical distance between the point (x_1, y_1) and the line described by $y = mx + b$ is $mx_1 + b - y_1$, the vertical distance between the point (x_2, y_2) and this line is $y_2 - mx_2 - b$, and the vertical distance between the point (x_3, y_3) and this line is $mx_3 + b - y_3$. One way to measure the distance between this set of points and the line is to sum up these distances, giving (for N data points)

$$\sum_{k=1}^{N} |y_k - mx_k - b|,$$

where absolute values have been introduced so that we don't have to keep track of whether a point is above or below the line. As we'll discuss a bit more below, this is a perfectly

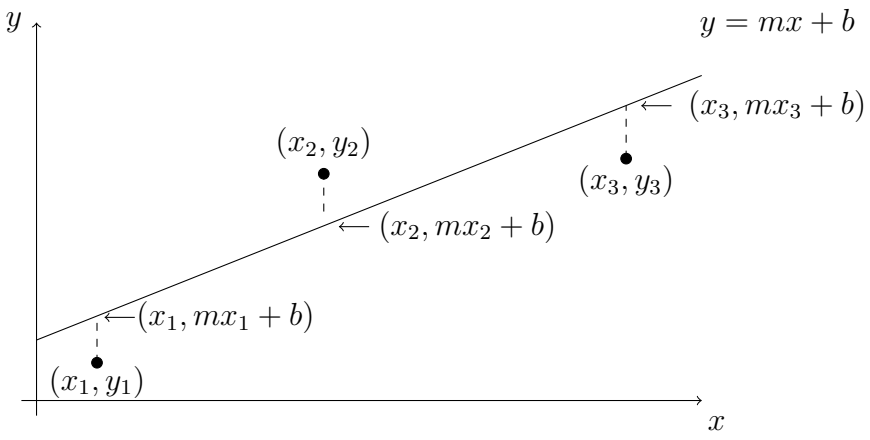


Figure 2.4: Line regression with three points.

reasonable measure of the distance between a set of points and a given line, but more commonly we measure the distance with the *sum of squared residuals (SSR)*

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

Alternatively, this same function $E(m, b)$ is sometimes referred to as the *residual sum of squares (RSS)* or the *sum of squared errors (SSE)*. We emphasize here that since the values $\{(x_k, y_k)\}_{k=1}^{N}$ are all fixed numerical values, E is a function of only two variables, m and b, as indicated.

It's clear from our development that the SSR isn't the only measure we could use. Alternatives include sums involving:

- absolute values instead of squares, as we initially wrote;
- different even powers such as 4 or 6, or (reduced) fraction powers with even numerators such as 2/3 or 4/5;
- horizontal distances;
- direct (shortest) distance between the points and the line;
- "weights" added to any of the above choices in order to emphasize the importance of certain measurements over others.

In general, each of these methods will lead to different values of m and b, so while for these notes we will set aside other options and focus exclusively on least squares regression, we should be mindful that in doing so we are making a non-trivial choice.

Having decided to take $E(m, b)$ as our measure of the distance between the set of points and a line drawn through them, we see that in order to find the best possible line we should

minimize $E(m,b)$. This is a standard problem from calculus, and we proceed in the usual way by finding values m and b so that

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

Differentiating, we find

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

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, upon substituting m into equation (2.1), we find

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

As long as the denominator in these expressions is not zero (and more on that possibility just below), the values m and b are uniquely determined, so $E(m,b)$ has precisely one local extreme point. Since E is quadratic in m and b, opening upward, this point must be a minimizer, as expected.

2.1.1 Mean, Variance, and Covariance of the Data

The expressions for m and b obtained above are often expressed in terms of the mean, variance, and covariance of the data. For data $\{(x_k, y_k)\}_{k=1}^N$, it's natural to define two means,

$$\mu_x := \frac{1}{N}\sum_{k=1}^N x_k \quad \text{and} \quad \mu_y := \frac{1}{N}\sum_{k=1}^N y_k.$$

The *variance* for $\vec{x}$ (with $\vec{x}$ viewed here and below as a vector whose elements are the data values $\{x_k\}_{k=1}^N$) is defined by

$$\mathrm{Var}(\vec{x}) := \frac{1}{N}\sum_{k=1}^N (x_k - \mu_x)^2,$$

and for reasons we will set aside for the moment, it's often useful to work with the sample variance[7]

$$\mathrm{SVar}(\vec{x}) := \frac{1}{N-1}\sum_{k=1}^N (x_k - \mu_x)^2.$$

Likewise, we define the covariance of $\vec{x}$ and $\vec{y}$ to be

$$\mathrm{Cov}(\vec{x}, \vec{y}) := \frac{1}{N}\sum_{k=1}^N (x_k - \mu_x)(y_k - \mu_y),$$

with the sample covariance obtained by replacing N in the denominator with $N-1$.

With this notation, it's straightforward to check that the values m and b obtained above can be expressed in the compact form

$$\begin{aligned} m &= \frac{\mathrm{Cov}(\vec{x}, \vec{y})}{\mathrm{Var}(\vec{x})} \\ b &= \mu_y - m\mu_x. \end{aligned}$$

We see that m and b are uniquely defined unless $\mathrm{Var}(\vec{x}) = 0$, and this condition is only possible if x_k is the same value for all $k = 1, 2, \ldots, N$. This, of course, is the case in which all data points lie on the same vertical line.

2.1.2 Correlation Coefficient and Coefficient of Determination

One of the things regression addresses is the extent to which two variables are correlated. (This is another concept that substantially goes back to Galton.) For line regression, correlation is often measured by the *correlation coefficient*, which has an elegant geometric

[7] In order to get some intuition about this, suppose there is only a single data point x_1, so that $\mu_x = x_1$. The variance is necessarily 0, and the sample variance is undefined. This is essentially a philosophical point: with only one data point, do we think there is no variance, or do we think we have no information about the variance (encoded mathematically as an undetermined ratio of the form $\frac{0}{0}$)? For readers with experience in probability theory, if $\{x_k\}_{k=1}^N$ is a collection of realizations of a random variable then the sample variance is an *unbiased* estimator of the variance of the random variable.

interpretation. For data $\{(x_k, y_k)\}_{k=1}^N$, let $\vec{x}$ and $\vec{y}$ denote data vectors, and set

$$\vec{\mu}_x := \begin{pmatrix} \mu_x \\ \mu_x \\ \vdots \\ \mu_x \end{pmatrix}, \quad \vec{\mu}_y := \begin{pmatrix} \mu_y \\ \mu_y \\ \vdots \\ \mu_y \end{pmatrix}.$$

The pair of vectors $\vec{x} - \vec{\mu}_x$ and $\vec{y} - \vec{\mu}_y$ determine a two-dimensional plane, which we can sketch as in Figure 2.5.

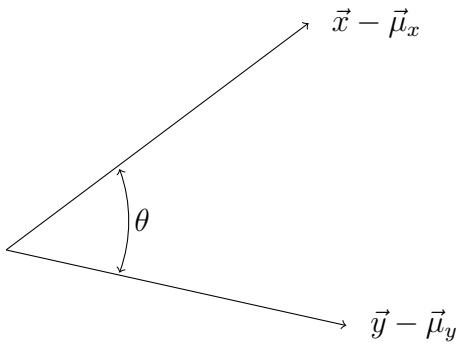


Figure 2.5: Geometric interpretation of the correlation coefficient.

The correlation coefficient is defined to be the value $\cos\theta$, which can be expressed as

$$\cos\theta = \frac{(\vec{x} - \vec{\mu}_x) \cdot (\vec{y} - \vec{\mu}_y)}{|\vec{x} - \vec{\mu}_x||\vec{y} - \vec{\mu}_y|},$$

where $\cdot$ denotes the usual dot product. Here, the purpose of subtracting the mean vectors $\vec{\mu}_x$ and $\vec{\mu}_y$ respectively from $\vec{x}$ and $\vec{y}$ is to ensure that the resulting quantities (i.e., the differences $\vec{x} - \vec{\mu}_x$ and $\vec{y} - \vec{\mu}_y$) are invariant under translations such as replacing $\vec{x}$ with $\vec{x} + \vec{a}$, where $\vec{a}$ is a vector with the same constant value a in every component.

Recalling our definitions of variance and covariance, we can express the correlation coefficient, often denoted R, by

$$R = \cos\theta = \frac{\mathrm{Cov}(\vec{x}, \vec{y})}{\sqrt{\mathrm{Var}(\vec{x})\mathrm{Var}(\vec{y})}}.$$

Since R is the cosine of an angle, its value must lie on the interval $[-1, 1]$. If $R = 1$, corresponding with $\theta = 0$, then the vectors $\vec{x} - \vec{\mu}_x$ and $\vec{y} - \vec{\mu}_y$ are colinear (i.e., lie on the same line), so there must exist a constant c so that

$$\vec{y} - \vec{\mu}_y = c(\vec{x} - \vec{\mu}_x).$$

In this case, every point in our data set $\{(x_k, y_k)\}_{k=1}^N$ must lie on the same line,

$$y_k = cx_k + (\mu_y - c\mu_x), \quad \forall\, k \in \{1, 2, \ldots, N\},$$

and the value c will just be the slope m of the line. Since $R = 1$, we must have $\mathrm{Cov}(\vec{x}, \vec{y}) > 0$, and so $m > 0$. On the other hand, if $R = -1$ (corresponding with $\theta = \pi$), then $\vec{x} - \vec{\mu}_x$ and $\vec{y} - \vec{\mu}_y$ are again colinear, but this time $m < 0$. In general, if $R > 0$ we refer to the correlation as positive, while if $R < 0$ we refer to the correlation as negative. In summary, values of R near 1 correspond with strong positive correlation, and values of R near -1 correspond with strong negative correlation. If $R = 0$, then $\mathrm{Cov}(\vec{x}, \vec{y}) = 0$, so $m = 0$. With a horizontal regression line, the value of x gives no information about the corresponding value of y, and we say the data is uncorrelated.

The *coefficient of determination* in the case of line regression is simply the value R^2. The value R^2 generalizes more naturally than the value of R, so we will have more to say about it later.

2.1.3 Generalizations of Regression for Lines

Our approach to finding a regression line to fit a set of data generalizes naturally to the following cases.

1. **Polynomials.** For any polynomial relation

$$y = p(x; \{a_j\}_{j=0}^m) = \sum_{j=0}^{m} a_j x^j,$$

the SSR is

$$E(a_0, a_1, \ldots, a_m) = \sum_{k=1}^{N} (y_k - \sum_{j=0}^{m} a_j x_k^j)^2.$$

Here, just as with regression for lines, E is quadratic in its variables, and so its partial derivatives will be linear as functions of a_0, a_1, ..., a_m. This is still linear regression.

2. **Multi-dimensional polynomials.** For $\vec{x} \in \mathbb{R}^2$, suppose we have a two-dimensional polynomial relation

$$y = p(\vec{x}; \{a_{ij}\}_{i,j=0}^{\ell,m}) = \sum_{i=0}^{\ell} \sum_{j=0}^{m} a_{ij} x_1^i x_2^j.$$

In this case, we denote our data

$$\{(\vec{x}_k, y_k)\}_{k=1}^N = \{(x_{k1}, x_{k2}, y_k)\}_{k=1}^N,$$

and the SSR is

$$E(\{a_{ij}\}_{i,j=0}^{\ell,m}) = \sum_{k=1}^{N} (y_k - \sum_{i=0}^{\ell} \sum_{j=0}^{m} a_{ij} x_{k1}^i x_{k2}^j)^2.$$

Again, this is linear regression. We can proceed similarly for $\vec{x} \in \mathbb{R}^n$ for any $n \in \mathbb{N}$ simply by introducing more indices.

3. **Functions linear in their parameters.** For $\vec{x} \in \mathbb{R}^n$, consider any family of functions $\{f_j(\vec{x})\}_{j=1}^m$, which we often take to be the basis for some finite-dimensional linear space (e.g., the space of polynomials up to order $m-1$). The function

$$y = f(\vec{x}; \vec{p}) = \sum_{j=1}^{m} p_j f_j(\vec{x})$$

is linear in its parameters, and so regression with this expression is still linear. The SSR in this case is

$$E(\vec{p}) = \sum_{k=1}^{N} (y_k - \sum_{j=1}^{m} p_j f_j(\vec{x}_k))^2.$$

In the same setting, suppose the parameters $\{p_j\}_{j=1}^m$ appear nonlinearly in the form

$$y = \sum_{j=1}^{m} q_j(\vec{p}) f_j(\vec{x})$$

for some functions $\{q_j(\vec{p})\}_{j=1}^m$. Then we can use linear regression to obtain optimal values $\{q_j^*\}_{j=1}^m$, and we can recover the optimal values $\vec{p}$ if we can solve the algebraic system

$$\vec{q}(\vec{p}) = \vec{q}^*.$$

We will see a specific example of this just below in Case 4.

4. **Functions that can be converted to a form linear in their parameters.** Consider a relationship of the form

$$y = f(x; \vec{p}) = p_1 e^{p_2 x}. \tag{2.2}$$

The associated SSR is

$$E(\vec{p}) = \sum_{k=1}^{N} (y_k - p_1 e^{p_2 x_k})^2, \tag{2.3}$$

and it's clear that if we take derivatives of E with respect to p_1 and p_2 we won't obtain relations linear in p_1 and p_2, so this is not linear regression. Nonetheless, if we take a natural logarithm of (2.2), we obtain the relation

$$\ln y = \ln(p_1 e^{p_2 x}) = \ln p_1 + p_2 x.$$

We can view this as the relation

$$Y = q_1 + q_2 x,$$

where $Y = \ln y$, $q_1 = \ln p_1$, and $q_2 = p_2$, and this relation is clearly linear in the parameters (in fact, it's just a line). If we now convert our data to the form $\{(x_k, Y_k)\}_{k=1}^N = \{(x_k, \ln y_k)\}_{k=1}^N$, then we can carry out linear regression in the usual way by minimizing the SSR

$$\tilde{E}(\vec{q}) = \sum_{k=1}^{N} (Y_k - q_1 - q_2 x_k)^2. \tag{2.4}$$

We then solve for p_1 and p_2 with the relations $p_1 = e^{q_1}$ and $p_2 = q_2$. This is a common approach taken throughout the mathematical sciences, but we need to be aware that the best-fit values for p_1 and p_2 obtained by directly minimizing (2.3) will generally be different from the values for p_1 and p_2 obtained by minimizing (2.4) and solving for p_1 and p_2. We will return to this point in our discussion below of nonlinear regression.

2.2 General Linear Regression

The form we will use for general linear regression is

$$y = f(\vec{x}; \vec{p}) = \sum_{j=1}^{m} p_j f_j(\vec{x}),$$

with data in the usual form $\{(x_k, y_k)\}_{k=1}^{N}$. In order to get some intuition about how to proceed, we can view each data point as corresponding with an equation that we would like to solve:

$$\begin{aligned}
k = 1: &\quad y_1 = p_1 f_1(\vec{x}_1) + p_2 f_2(\vec{x}_1) + \cdots + p_m f_m(\vec{x}_1) \\
k = 2: &\quad y_2 = p_1 f_1(\vec{x}_2) + p_2 f_2(\vec{x}_2) + \cdots + p_m f_m(\vec{x}_2) \\
&\quad \vdots \quad \vdots \\
k = N: &\quad y_N = p_1 f_1(\vec{x}_N) + p_2 f_2(\vec{x}_N) + \cdots + p_m f_m(\vec{x}_N).
\end{aligned}$$

We expect to have $N \gg m$ (many more data points than parameters), so this is an overdetermined systems of N equations (one for each data point) for m unknowns (the m parameters). In order to express this system in matrix form, we'll write

$$\vec{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}, \quad \vec{p} = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_m \end{pmatrix},$$

and specify the *design matrix* as[8]

$$F = \begin{pmatrix} f_1(\vec{x}_1) & f_2(\vec{x}_1) & \cdots & f_m(\vec{x}_1) \\ f_1(\vec{x}_2) & f_2(\vec{x}_2) & \cdots & f_m(\vec{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ f_1(\vec{x}_N) & f_2(\vec{x}_N) & \cdots & f_m(\vec{x}_N) \end{pmatrix} \in \mathbb{R}^{N \times m}.$$

In matrix form, we can now express the above system as

$$F\vec{p} = \vec{y}.$$

It's important to keep in mind that we have already seen that this system is generally overdetermined, and so won't have an exact solution. In fact, if we could solve it exactly, it

[8] In statistics courses, the design matrix is often denoted X due to its dependence on the variable $\vec{x}$.

would mean that every single data point is exactly on our fit. For example, for a line, every single data point would actually lie on the best-fit line. Since F isn't a square matrix, it certainly isn't invertible, but there is a nice intuitive way to think about how we can find the "best possible" values for $\vec{p}$. Notice that if we multiply both sides of this system by the matrix F^T (i.e., the transpose of F), we obtain the equation

$$F^T F\vec{p} = F^T\vec{y}. \tag{2.5}$$

This is called the *normal equation*, and a key observation is that $F^T F$ is always a square $m \times m$ matrix. Of course, this new system was obtained in an ad hoc way, so we don't immediately know that its solutions are the best-fit values of $\vec{p}$ that we're looking for. In order to check that they are, let's express our SSR in the form

$$E(\vec{p}) = \sum_{k=1}^{N}(y_k - \sum_{j=1}^{m} p_j f_j(\vec{x}_k))^2.$$

and let's observe that

$$\sum_{j=1}^{m} p_j f_j(\vec{x}_k) = \sum_{j=1}^{m} p_j F_{kj} = (F\vec{p})_k,$$

where the far right-hand side indicates the k^{th} component of the vector $F\vec{p}$. Writing

$$E(\vec{p}) = \sum_{k=1}^{N}(y_k - \sum_{j=1}^{m} F_{kj}p_j)^2,$$

we compute (for each $l = 1, 2, \ldots, m$)

$$\frac{\partial E}{\partial p_l} = \sum_{k=1}^{N} 2(y_k - \sum_{j=1}^{m} F_{kj}p_j)(-F_{kl}) = 0,$$

which can be expressed as

$$\sum_{k=1}^{N} F_{kl}(F\vec{p})_k = \sum_{k=1}^{N} F_{kl}y_k,$$

or equivalently

$$\sum_{k=1}^{N} F^T_{lk}(F\vec{p})_k = \sum_{k=1}^{N} F^T_{lk}y_k.$$

This final equality is just the index form of the relation

$$(F^T F\vec{p})_l = (F^T\vec{y})_l,$$

and since this is true for all $l = 1, 2, \ldots, m$, we see that

$$F^T F\vec{p} = F^T\vec{y},$$

which is precisely the normal equation.

In the event that F^TF is invertible, we can solve for the best-fit parameter values with

$$\vec{p} = (F^TF)^{-1}F^T\vec{y}.$$

At the end of this section, we will review a few notions from linear algebra and then verify that F^TF is invertible if and only if the columns of F are linearly independent. In practical terms, this means that F^TF is almost always invertible in applications. To get a feel for why this is true, we can think about the case of line regression, for which F will have exactly two columns, the first with each element having the value 1, and the second with the k^{th} element having the value x_k. I.e., if we think of the relation as $y = p_1 + p_2x$, then we will have $f_1(x) = 1$ and $f_2(x) = x$ so that

$$F = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & \vdots \\ 1 & x_N \end{pmatrix}.$$

These two columns are linearly dependent if and only if one is a multiple of the other: i.e., there exists a constant $c \in \mathbb{R}$ so that

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} = c \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix},$$

in which case $x_k = c$ for all $k = 1, 2, \ldots, N$, and we are in the unlikely setting discussed above in which all data points lie on the same vertical line.

2.2.1 Generalizing the Coefficient of Determination

For line regression, we saw that the correlation coefficient is

$$R = \frac{\mathrm{Cov}(\vec{x}, \vec{y})}{\sqrt{\mathrm{Var}(\vec{x})\mathrm{Var}(\vec{y})}},$$

and the coefficient of determination is R^2. These values are closely tied to the geometry of line regression (in particular to the slope of the regression line), but R^2 generalizes fairly naturally to the general case. To see this, let's continue to denote by E the usual SSR, and let's now also set

$$T := \sum_{k=1}^{N}(y_k - \mu_y)^2 = N\mathrm{Var}(\vec{y}), \tag{2.6}$$

often called the *total sum of squares*. We will base our generalization on the following lemma.

Lemma 2.1. *For line regression,*

$$R^2 = 1 - \frac{E}{T}.$$

Proof. First, we recall that for line regression,

$$m = \frac{\mathrm{Cov}(\vec{x}, \vec{y})}{\mathrm{Var}(\vec{x})}, \quad b = \mu_y - m\mu_x.$$

Now, we compute:

$$\begin{aligned}
E &= \sum_{k=1}^{N} (y_k - mx_k - b)^2 \\
&= \sum_{k=1}^{N} (y_k - mx_k - \mu_y + m\mu_x)^2 \\
&= \sum_{k=1}^{N} \Big\{ (y_k - \mu_y)^2 + 2(y_k - \mu_y)(-mx_k + m\mu_x) + (-mx_k + m\mu_x)^2 \Big\} \\
&= \sum_{k=1}^{N} (y_k - \mu_y)^2 - 2m \sum_{k=1}^{N} (y_k - \mu_y)(x_k - \mu_x) + m^2 \sum_{k=1}^{N} (x_k - \mu_x)^2 \\
&= T - 2mN\mathrm{Cov}(\vec{x}, \vec{y}) + m^2 N \mathrm{Var}(\vec{x}) \\
&= T - 2N \frac{\mathrm{Cov}(\vec{x}, \vec{y})^2}{\mathrm{Var}(\vec{x})} + N \frac{\mathrm{Cov}(\vec{x}, \vec{y})^2}{\mathrm{Var}(\vec{x})} \\
&= T - N \frac{\mathrm{Cov}(\vec{x}, \vec{y})^2}{\mathrm{Var}(\vec{x})}.
\end{aligned}$$

Rearranging terms, we find that

$$1 - \frac{E}{T} = \frac{T - E}{T} = \frac{N\mathrm{Cov}(\vec{x}, \vec{y})^2}{N\mathrm{Var}(\vec{y})\mathrm{Var}(\vec{x})} = R^2,$$

which is precisely the sought relation. □

Since E and T are both defined in the general case, we can take Lemma 2.1 as justification for *defining* the coefficient of determination in the general case to be

$$R^2 := 1 - \frac{E}{T}.$$

The values E and T are both necessarily non-negative, so it's clear that $R^2 \leq 1$. In addition, if the model has an intercept so that we can express it as

$$y = p_1 + p_2 f_2(\vec{x}) + \cdots + p_m f_m(\vec{x}),$$

then one possible set of parameter values is $p_1 = \mu_y$, $p_2 = p_3 = \cdots = p_m = 0$, in which case E would equal T. Since E is the minimum over all possible parameter values, we must have $E \leq T$, and so $R^2 \geq 0$. More generally, despite the notation R^2, it's possible for the coefficient of determination to have negative values.[9]

[9]This happens in cases for which the data clearly has a non-zero intercept, and so a model without an intercept is certainly a bad fit. See, for example, *Cautionary Note about* R^2, The American Statistician **39** (1985) 279-285, by Tarold O. Kvålseth, for a detailed discussion of related issues.

2.2.2 Variance on Predictions and the Adjusted Coefficient of Determination

Recall that the sample variance for a data vector $\vec{y}$ is

$$\mathrm{SVar}(\vec{y}) = \frac{1}{N-1}\sum_{k=1}^{N}(y_k - \mu_y)^2 = \frac{T}{N-1}.$$

The associated standard deviation is $\sqrt{T/(N-1)}$, and we often use this as a measure of how far measurements on the dependent variable y may differ from the mean μ_y. For example, under certain assumptions, we expect that about 68%[10] of measured values y will lie on the interval

$$(\mu_y - \sqrt{\frac{T}{N-1}}, \mu_y + \sqrt{\frac{T}{N-1}}).$$

(For a more precise discussion of this notion of *confidence intervals*, see the M442 course notes *Modeling with Probability*.) More important for our current purposes, we would like to estimate the variance and standard deviation on predictions of the independent variable y made by our model. Again under appropriate assumptions, we can compute this as

$$s^2 = \frac{E}{q}, \quad q = N - m.$$

Notice that if $N = m$, then we have the same number of data points as parameters, and so our model will generally be a perfect fit. In this case, we don't expect to have any information about the variance, and so we're in a setting similar to that of Footnote 7. We can interpret these considerations as follows: if our model is $y = f(\vec{x};\vec{p})$, and we want to predict a specific value y_0 corresponding with some given input $\vec{x}_0$, then there is a roughly 68% chance that y_0 will lie on the interval

$$(f(\vec{x}_0;\vec{p}) - s, f(\vec{x}_0;\vec{p}) - s).$$

We often write

$$y = f(\vec{x}_0;\vec{p}) \pm s.$$

Last, the same considerations lead to the *adjusted coefficient of determination*

$$\begin{aligned}\bar{R}^2 :=& 1 - \frac{\frac{1}{q}E}{\frac{1}{N-1}T} = 1 - \frac{N-1}{q}\frac{E}{T}\\ =& 1 - \frac{N-1}{q}(1-R^2).\end{aligned}$$

In cases (as discussed above) for which R^2 ranges on the interval $[0,1]$, $\bar{R}^2$ ranges on the interval $[1 - \frac{N-1}{q}, 1]$.

[10] Precisely, this is the percentage associated with plus or minus one standard deviation for the standard normal random variable; i.e., $\int_{-1}^{1}\frac{1}{\sqrt{2\pi}}e^{-x^2/2}dx \cong .6827$.

2.2.3 MATLAB Implementation

In the MATLAB M-file *fueltempwind.m* provided below, we have data relating temperature and wind speed to the amount of fuel required to heat a home.[11] Our model for this data will be

$$\text{fuel} = p_1 + p_2 * \text{temp} + p_3 * \text{wind},$$

and we will express this as

$$y = p_1 + p_2 x_1 + p_3 x_2.$$

I.e., we have our general form

$$y = \sum_{j=1}^{m} p_j f_j(\vec{x}),$$

with $m = 3$, $f_1(\vec{x}) = 1$, $f_2(\vec{x}) = x_1$, and $f_3(\vec{x}) = x_2$. As usual, we express our data as

$$\{(\vec{x}_k, y_k)\}_{k=1}^{N} = \{(x_{k1}, x_{k2}, y_k)\}_{k=1}^{N}.$$

If we write this out as we did for our general development of the design matrix F, we get

$$\begin{aligned} k = 1: &\quad y_1 = p_1 + p_2 x_{11} + p_3 x_{12} \\ k = 2: &\quad y_2 = p_1 + p_2 x_{21} + p_3 x_{22} \\ \vdots &\quad \vdots \\ k = N: &\quad y_N = p_1 + p_2 x_{N1} + p_3 x_{N2}, \end{aligned}$$

and we obtain $F\vec{p} = \vec{y}$, with

$$F = \begin{pmatrix} 1 & x_{11} & x_{12} \\ 1 & x_{21} & x_{22} \\ \vdots & \vdots & \vdots \\ 1 & x_{N1} & x_{N2} \end{pmatrix}.$$

We have seen that the three columns of F will likely be linearly independent, and we verify this in the code below by checking that $\det(F^T F) \neq 0$. This allows us to compute the least-squares parameter vector as

$$\vec{p} = (F^T F)^{-1} F^T \vec{y},$$

and the syntax for this calculation in the MATLAB code is

```
>>p=F\y
```

The code for this powerful and versatile MATLAB functionality was written by Tim Davis, who is a professor at Texas A&M University, in the department of Computer Science and Engineering. According to Professor Davis, the code runs to about 120,000 lines.

[11] This data is taken from the book *Probability and Statistical Inference vol. 2: Statistical inference*, second edition, Springer-Verlag 1995, by J. G. Kalbfleisch.

```
%FUELTEMPWIND1: MATLAB script M-file that computes parameter
%values for a multilinear fit to the fuel-temperature-wind
%example from "Probability and Statistical Inference, vol. 2:
%Statistical Inference," by J. G. Kalbfleisch (see p. 249).
fuel = [14.96 14.1 23.76 13.2 18.6 16.79 21.83 16.25 20.98 16.88]
temp = [-3 -1.8 -10 .7 -5.1 -6.3 -15.5 -4.2 -8.8 -2.3]
wind = [15.3 16.4 41.2 9.7 19.3 11.4 5.9 24.3 14.7 16.1]
%
F=[ones(size(fuel))' temp' wind']
%Check invertibility of F'*F
det(F'*F)
%Brute force calculation of p
p = inv(F'*F)*F'*fuel'
%MATLAB calculation of p (much more efficient)
pmat=F\fuel'
%SSR
ssr = norm(fuel'-F*p)^2
%Standard deviation for the fit
q = length(fuel)-length(p);
s = sqrt(ssr/q)
%R-squared calculations
fuelbar = mean(fuel);
T = norm(fuel-fuelbar)^2;
%Coefficient of determination
Rsq = 1-ssr/T
%Adjusted coefficient of determination
Rsqa = 1-(1-Rsq)*(length(fuel)-1)/q
```

Running *fueltempwind1.m*, we find that

$$\begin{aligned} p_1 &= 11.9339 \\ p_2 &= -.6285 \\ p_3 &= .1298, \end{aligned}$$

with corresponding standard deviation

$$s = 1.2267,$$

and coefficient of determination $R^2 = .9039$. The modified coefficient of determination is $\bar{R}^2 = .8765$.

2.2.4 Invertibility of F^TF and Interpreting the relation $F\vec{p} = \vec{y}$

For this discussion, we need to review two results from linear algebra, the first of which we will state without proof. These address existence and uniqueness for matrix equations of the form $A\vec{x} = \vec{b}$, and we will start with uniqueness.

Theorem 2.1 (Uniquenesss Theorem for Matrix Equations). *For any matrix $A \in \mathbb{C}^{m\times n}$ (i.e., any $m \times n$ matrix with complex-valued entries, which of course includes any $m \times n$ matrix with real-valued entries), and any column vector $\vec{b} \in \mathbb{C}^m$, a solution to $A\vec{x} = \vec{b}$, if it exists, is unique if and only if one (and so both) of the following equivalent conditions hold:*

(i) $\vec{x} = 0$ is the only solution of $A\vec{x} = 0$;

(ii) The columns of A are linearly independent.

Moreover, if A is a square matrix, then Items (i) and (ii) are equivalent to the following:

(iii) $\det A \neq 0$*;*

(iv) A is invertible;

(v) The rows of A are linearly independent.

For the next theorem (our existence theorem for solutions to equations of the form $A\vec{x} = \vec{b}$), we will denote by A^* the adjoint of a matrix A, which is the matrix obtained by transposing A and taking a complex conjugate of each entry. It's clear from this definition that if the entries of A are all real-valued, then $A^* = A^T$. In addition, we will use the notation $\langle \vec{x}, \vec{y} \rangle$ to denote the usual inner product between two vectors $\vec{x}, \vec{y} \in \mathbb{C}^n$,

$$\langle \vec{x}, \vec{y} \rangle = \sum_{j=1}^{n} x_j \bar{y}_j,$$

where the bar denotes complex conjugate. It's straightforward to check that for any matrix $A \in \mathbb{C}^{m\times n}$, and any vectors $\vec{x} \in \mathbb{C}^n$ and $\vec{y} \in \mathbb{C}^m$, we have the relation

$$\langle A\vec{x}, \vec{y} \rangle = \langle \vec{x}, A^*\vec{y} \rangle.$$

Theorem 2.2 (The Fredholm Alternative for Matrices).[12] *For any matrix $A \in \mathbb{C}^{m\times n}$, and any column vector $\vec{b} \in \mathbb{C}^m$, the equation $A\vec{x} = \vec{b}$ has a solution if and only if $\langle \vec{b}, \vec{v} \rangle = 0$ for every vector $\vec{v}$ satisfying $A^*\vec{v} = 0$.*

Since the Fredholm Alternative is sometimes skipped in introductory courses in linear algebra, we include a proof. For this, we need a couple of definitions and one preliminary result, known as the Orthogonal Decomposition Theorem.

Definition 2.1. *For any two subspaces X and Y of $\mathbb{C}^n$ such that $X \cap Y = \{0\}$, we define the direct sum*

$$Z = X \oplus Y$$

to be the collection of vectors $\vec{z} \in \mathbb{C}^n$ that can be expressed as $\vec{z} = \vec{x} + \vec{y}$, where $\vec{x} \in X$ and $\vec{y} \in Y$.

Lemma 2.2. *With X and Y as in Definition 2.1, the decomposition $\vec{z} = \vec{x} + \vec{y}$ is unique. I.e., if $\vec{z} = \vec{u} + \vec{v}$ for some $\vec{u} \in X$ and $\vec{v} \in Y$, then it must be the case that $\vec{u} = \vec{x}$ and $\vec{v} = \vec{y}$.*

[12] This theorem is named for the Swedish mathematician Erik Fredholm (1866-1927).

Proof. Upon subtracting $\vec{z} = \vec{u} + \vec{v}$ from $\vec{z} = \vec{x} + \vec{y}$, we obtain the relation

$$(\vec{x} - \vec{u}) + (\vec{y} - \vec{v}) = 0,$$

where by linearity $\vec{x} - \vec{u} \in X$ and $\vec{y} - \vec{v} \in Y$. But we have

$$(\vec{x} - \vec{u}) = -(\vec{y} - \vec{v}) \in Y,$$

so $\vec{x} - \vec{u} \in X \cap Y$. This implies $\vec{x} - \vec{u} = 0$, and we can conclude $\vec{u} = \vec{x}$. But then we immediately have $\vec{v} = \vec{y}$ as well. □

Definition 2.2. *For any subspace X of $\mathbb{C}^n$, we define the orthogonal complement of X to be*

$$X^\perp := \{\vec{y} \in \mathbb{C}^n : \langle \vec{x}, \vec{y} \rangle = 0 \text{ for all } \vec{x} \in X\}.$$

Theorem 2.3 (Orthogonal Decomposition Theorem). *If X is any subspace of $\mathbb{C}^n$, then*

$$\mathbb{C}^n = X \oplus X^\perp.$$

Proof. First, let's check that $X \cap X^\perp = \{0\}$. For this, we suppose $\vec{v} \in X \cap X^\perp$, and compute

$$|\vec{v}|^2 = \langle \vec{v}, \vec{v} \rangle = 0,$$

where we obtain 0 because $\vec{v} \in X^\perp$ implies that $\langle \vec{v}, \vec{x} \rangle = 0$ for all $\vec{x} \in X$, and we are taking $\vec{v} \in X$. Next, we need to show that given any $\vec{z} \in \mathbb{C}^n$ we can write $\vec{z} = \vec{x} + \vec{y}$, where $\vec{x} \in X$ and $\vec{y} \in X^\perp$. First, if $\dim X = 0$ then the only element in X is 0, so $X^\perp = \mathbb{C}^n$, and we see that the statement holds with $\vec{z} = \vec{y}$. If $\dim X = \ell \geq 1$, we let $\{\vec{x}_j\}_{j=1}^\ell$ denote an orthogonal basis for X.[13] We then set

$$\vec{x} = \sum_{j=1}^{\ell} \frac{\langle \vec{z}, \vec{x}_j \rangle}{|\vec{x}_j|^2} \vec{x}_j.$$

Since $\vec{x}$ is a linear combination of the basis elements of X, we certainly have $\vec{x} \in X$. We'll set $\vec{y} = \vec{z} - \vec{x}$, and we will obtain the sought relation if we can show that $\vec{y} \in X^\perp$. To this end, we take any $\vec{v} \in X$, noting that we can express $\vec{v}$ as $\vec{v} = \sum_{j=1}^\ell c_j \vec{x}_j$ for some constants $\{c_j\}_{j=1}^\ell$. We need to show that $\langle \vec{y}, \vec{v} \rangle = 0$, and for this we compute

$$\begin{aligned}
\langle \vec{y}, \vec{v} \rangle &= \langle \vec{z} - \vec{x}, \vec{v} \rangle = \langle \vec{z}, \vec{v} \rangle - \langle \vec{x}, \vec{v} \rangle \\
&= \langle \vec{z}, \sum_{j=1}^{\ell} c_j \vec{x}_j \rangle - \langle \sum_{k=1}^{\ell} \frac{\langle \vec{z}, \vec{x}_k \rangle}{|\vec{x}_k|^2} \vec{x}_k, \sum_{j=1}^{\ell} c_j \vec{x}_j \rangle \\
&= \sum_{j=1}^{\ell} c_j \langle \vec{z}, \vec{x}_j \rangle - \sum_{k=1}^{\ell} \frac{\langle \vec{z}, \vec{x}_k \rangle}{|\vec{x}_k|^2} \sum_{j=1}^{\ell} c_j \langle \vec{x}_k, \vec{x}_j \rangle \\
&= \sum_{j=1}^{\ell} c_j \langle \vec{z}, \vec{x}_j \rangle - \sum_{k=1}^{\ell} \frac{\langle \vec{z}, \vec{x}_k \rangle}{|\vec{x}_k|^2} c_k |\vec{x}_k|^2 = 0,
\end{aligned}$$

[13] I.e., $\langle \vec{x}_i, \vec{x}_j \rangle = 0$ for all $i \neq j$; this is always possible by Gram-Schmidt orthogonalization.

where in the step just before equating the relation to 0 we used the orthogonality. This gives the claim. □

Proof of the Fredholm Alternative. First, suppose we can solve $A\vec{x} = \vec{b}$ with some $\vec{x}_0 \in \mathbb{C}^n$. Let $\vec{v}$ be any vector in $\mathbb{C}^m$ satisfying $A^*\vec{v} = 0$. Then

$$\langle \vec{b}, \vec{v} \rangle = \langle A\vec{x}_0, \vec{v} \rangle = \langle \vec{x}_0, A^*\vec{v} \rangle = \langle \vec{x}_0, 0 \rangle = 0,$$

giving the forward implication.

On the other hand, suppose

$$A^*\vec{v} \implies \langle \vec{b}, \vec{v} \rangle = 0. \tag{2.7}$$

According to the Orthogonal Decomposition Theorem, we have

$$\mathbb{C}^m = \mathcal{R}(A) \oplus \mathcal{R}(A)^\perp,$$

which means that we can express $\vec{b}$ uniquely as

$$\vec{b} = \vec{b}_r + \vec{b}_0,$$

where $\vec{b}_r \in \mathcal{R}(A)$ and $\vec{b}_0 \in \mathcal{R}(A)^\perp$. Here, $\mathcal{R}(A)$ denotes the *range* of A, which is just every vector $\vec{b} \in \mathbb{C}^m$ for which there exists a vector $\vec{x} \in \mathbb{C}^n$ so that $A\vec{x} = \vec{b}$. Perhaps more simply, $\mathcal{R}(A)$ is precisely the set of vectors $\vec{b} \in \mathbb{C}^m$ for which we can solve the equation $A\vec{x} = \vec{b}$. We have then that for all $\vec{x} \in \mathbb{C}^n$,

$$0 = \langle \vec{b}_0, A\vec{x} \rangle = \langle A^*\vec{b}_0, \vec{x} \rangle,$$

where the zero on the left-hand side is because $A\vec{x} \in \mathcal{R}(A)$. Since $\vec{x}$ is arbitrary, we can take $\vec{x} = A^*\vec{b}_0$ to see that $A^*\vec{b}_0 = 0$. But then by (2.7) we must have $\langle \vec{b}, \vec{b}_0 \rangle = 0$. This allows us to compute

$$\begin{aligned} |\vec{b}_0|^2 = \langle \vec{b}_0, \vec{b}_0 \rangle &= \langle \vec{b}_0, \vec{b} - \vec{b}_r \rangle \\ &= \langle \vec{b}_0, \vec{b} \rangle - \langle \vec{b}_0, \vec{b}_r \rangle = 0, \end{aligned}$$

because each summand is 0. We see that $\vec{b}_0 = 0$, so that $\vec{b} = \vec{b}_r$, and since $\vec{b}_r \in \mathcal{R}(A)$, we can solve $A\vec{x} = \vec{b}_r$, completing the proof. □

In Fredholm's Alternative, the "alternative" is that either $A\vec{x} = \vec{b}$ has a solution or $A^*\vec{v} = 0$ has a solution so that $\langle \vec{b}, \vec{v} \rangle \neq 0$. One succinct way to express the theorem is to write

$$\mathcal{R}(A) = \mathcal{N}(A^*)^\perp,$$

where the set $\mathcal{N}(A^*)$ denotes the *null space* (or *kernel*) of A^*, which is just the collection of all vectors $\vec{v} \in \mathbb{C}^n$ so that $A^*\vec{v} = 0$. The orthogonal complement $\mathcal{N}(A)^\perp$ is precisely the set of vectors $\vec{b}$ described in the theorem:

$$\mathcal{N}(A^*)^\perp = \{\vec{b} \in \mathbb{C}^m : \langle \vec{b}, \vec{v} \rangle = 0 \text{ for all } \vec{v} \in \mathcal{N}(A^*)\},$$

so the Fredholm Alternative asserts that this equals the $\mathcal{R}(A)$, as stated. In addition, according to the Orthogonal Decomposition Theorem, we have

$$\mathbb{C}^m = \mathcal{N}(A^*) \oplus \mathcal{N}(A^*)^\perp,$$

giving the useful relation

$$\mathbb{C}^m = \mathcal{N}(A^*) \oplus \mathcal{R}(A).$$

Using Theorem 2.1 (though not yet Theorem 2.2), we can prove the following lemma about invertibility of F^TF.

Lemma 2.3. *For any matrix $F \in \mathbb{C}^{N\times m}$, the matrix $F^*F \in \mathbb{C}^{m\times m}$ is invertible if and only if the columns of F are linearly independent.*

Proof. According to Theorem 2.1, it suffices to show that $\vec{x} = 0$ is the only solution of $F\vec{x} = 0$ if and only if $\vec{x} = 0$ is the only solution of $F^*F\vec{x} = 0$. We will show

$$F\vec{x} = 0 \quad \iff \quad F^*F\vec{x} = 0,$$

which gives the claim. First, for ($\implies$), this is clear (just multiply by F^*). For ($\impliedby$), we suppose $F^*F\vec{x} = 0$ and compute

$$0 = \langle F^*F\vec{x}, \vec{x}\rangle = \langle F\vec{x}, F\vec{x}\rangle = |F\vec{x}|^2 = 0,$$

from which we conclude that $F\vec{x} = 0$. □

In addition, we can use the Fredholm Alternative to better understand how we should interpret the relation $F\vec{p} = \vec{y}$. These observations hold for $F \in \mathbb{C}^{N\times m}$, but in order to be consistent with our discussion of regression so far we will continue to take $F \in \mathbb{R}^{N\times m}$. The primary difference is that we will write F^T instead of F^*. To begin, we observe that if F^TF is invertible so that $\vec{p} = (F^TF)^{-1}F^T\vec{y}$, then we can express $F\vec{p} = \vec{y}$ as

$$F(F^TF)^{-1}F^T\vec{y} = \vec{y}.$$

This matrix $M := F(F^TF)^{-1}F^T$ is precisely the orthogonal projection onto the range of F, so while $F\vec{p}$ isn't generally precisely $\vec{y}$, it is in some sense as close to being $\vec{y}$ as possible. But what exactly do we mean when we say that M is an orthogonal projection onto the range of F? To understand this, let's first observe that based on our discussion above about the Fredholm Alternative, we can write

$$\mathbb{R}^N = \mathcal{R}(F) \oplus \mathcal{N}(F^T).$$

This means precisely that given any $\vec{y} \in \mathbb{R}^N$, there exists a unique pair of vectors $\vec{x} \in \mathcal{R}(F)$ and $\vec{z} \in \mathcal{N}(F^T)$ so that

$$\vec{y} = \vec{x} + \vec{z}.$$

The orthogonal projection of $\vec{y}$ onto the range of F (i.e., onto $\mathcal{R}(F)$) is the part of this decomposition in $\mathcal{R}(F)$, namely $\vec{x}$.[14] To see that M is indeed the orthogonal projection onto

[14]The projection is *orthogonal* because its range $\mathcal{R}(M)$ is orthogonal to its null space $\mathcal{N}(M)$. That is, if $\vec{y} \in \mathcal{R}(M)$, then there exists some $\vec{x} \in \mathbb{R}^N$ so that $M\vec{x} = \vec{y}$, and if $\vec{z} \in \mathcal{N}(M)$ it means that $M\vec{z} = 0$. But then $\langle \vec{y}, \vec{z}\rangle = \langle M\vec{x}, \vec{z}\rangle = \langle \vec{x}, M\vec{z}\rangle = \langle \vec{x}, 0\rangle = 0$.

$\mathcal{R}(F)$, we just compute

$$\begin{aligned} M\vec{y} &= M(\vec{x}+\vec{z}) = M\vec{x} + M\vec{z} \\ &= F(F^TF)^{-1}F^T\vec{x} + F(F^TF)^{-1}F^T\vec{z} \\ &= F(F^TF)^{-1}F^T\vec{x}, \end{aligned}$$

because $F^T\vec{z} = 0$. But also since $\vec{x} \in \mathcal{R}(F)$ there must exist some $\vec{v} \in \mathbb{R}^m$ so that $\vec{x} = F\vec{v}$, and this allows us to continue this calculation by writing

$$M\vec{y} = F(F^TF)^{-1}F^T\vec{x} = F(F^TF)^{-1}F^TF\vec{v} = F\vec{v} = \vec{x},$$

where in obtaining the final equality we observed that $(F^TF)^{-1}F^TF = I$ by the usual property of inverses.

2.3 Linear Regression for Systems

So far, our dependent variable y has been a scalar, but in many cases we want to carry out regression on a system. Suppose we have data $\{(\vec{x}_k, \vec{y}_k)\}_{k=1}^N$, which we can express as

$$\{(x_{k1}, x_{k2}, \ldots, x_{kl}, y_{k1}, y_{k2}, \ldots, y_{kn})\}_{k=1}^N.$$

Here, for each $k \in \{1, 2, \ldots, N\}$, we have $\vec{x}_k \in \mathbb{R}^l$ and $\vec{y}_k \in \mathbb{R}^n$. We will express our general linear regression relation as

$$\vec{y} = \vec{f}(\vec{x}; \vec{p}),$$

which can be expressed in component form as

$$\begin{aligned} y_1 &= f_1(\vec{x}; \vec{p}) = \sum_{j=1}^m p_j f_{1j}(\vec{x}) \\ y_2 &= f_2(\vec{x}; \vec{p}) = \sum_{j=1}^m p_j f_{2j}(\vec{x}) \\ &\vdots \\ y_n &= f_n(\vec{x}; \vec{p}) = \sum_{j=1}^m p_j f_{nj}(\vec{x}). \end{aligned}$$

Of course, we could analyze each component equation separately as a single equation, allowing us to proceed as before, but we would generally get different values for the parameters from each equation, and in addition some parameters may be omitted entirely from an equation.[15] In view of this, we generally treat the system cohesively by specifying the SSR

$$\begin{aligned} E(\vec{p}) &= \sum_{k=1}^N |\vec{y}_k - \vec{f}(\vec{x}_k; \vec{p})|^2 \\ &= \sum_{k=1}^N \sum_{i=1}^n (y_{ki} - f_i(\vec{x}_k; \vec{p}))^2, \end{aligned}$$

[15] For example, if $f_{21}(\vec{x}) \equiv 0$, then p_1 does not appear in the second equation.

where $|\vec{y}_k - \vec{f}(\vec{x}_k;\vec{p})|$ denotes the usual Euclidian norm of the vector $\vec{y}_k - \vec{f}(\vec{x}_k;\vec{p})$. In this case, we can think of the data as giving us n systems of N equations, namely

$$\begin{aligned}
i=1, \quad k=1: &\quad y_{11} = p_1 f_{11}(\vec{x}_1) + p_2 f_{12}(\vec{x}_1) + \cdots + p_m f_{1m}(\vec{x}_1) \\
k=2: &\quad y_{21} = p_1 f_{11}(\vec{x}_2) + p_2 f_{12}(\vec{x}_2) + \cdots + p_m f_{1m}(\vec{x}_2) \\
&\quad \vdots \quad \vdots \\
k=N: &\quad y_{N1} = p_1 f_{11}(\vec{x}_N) + p_2 f_{12}(\vec{x}_N) + \cdots + p_m f_{1m}(\vec{x}_N) \\
i=2, \quad k=1: &\quad y_{12} = p_1 f_{21}(\vec{x}_1) + p_2 f_{22}(\vec{x}_1) + \cdots + p_m f_{2m}(\vec{x}_1) \\
&\quad \vdots \quad \vdots \\
i=n, \quad k=N: &\quad y_{Nn} = p_1 f_{n1}(\vec{x}_N) + p_2 f_{n2}(\vec{x}_N) + \cdots + p_m f_{nm}(\vec{x}_N).
\end{aligned}$$

In this case, we will set

$$\vec{Y}_i := \begin{pmatrix} y_{1i} \\ y_{2i} \\ \vdots \\ y_{Ni} \end{pmatrix}, \quad F_i := \begin{pmatrix} f_{i1}(\vec{x}_1) & f_{i2}(\vec{x}_1) & \cdots & f_{im}(\vec{x}_1) \\ f_{i1}(\vec{x}_2) & f_{i2}(\vec{x}_2) & \ldots & f_{im}(\vec{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ f_{i1}(\vec{x}_N) & f_{i2}(\vec{x}_N) & \ldots & f_{im}(\vec{x}_N) \end{pmatrix}, \tag{2.8}$$

so that our equations are

$$F_i \vec{p} = \vec{Y}_i, \quad i = 1, 2, \ldots, n.$$

If we subsequently set

$$\vec{Y} := \begin{pmatrix} \vec{Y}_1 \\ \vec{Y}_2 \\ \vdots \\ \vec{Y}_n \end{pmatrix} \in \mathbb{R}^{nN}, \quad F = \begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ F_n \end{pmatrix} \in \mathbb{R}^{nN \times m},$$

then we can express our linear system of equations as

$$F\vec{p} = \vec{Y}.$$

Proceeding almost identically as in the case of a single equation, we can check that minimizers $\vec{p}$ of the SSR can again be computed as

$$\vec{p} = (F^T F)^{-1} F^T \vec{Y},$$

as long as $F^T F$ is invertible. We will work through a specific example below, but first we need to introduce a key new feature that becomes important in the case of systems.

2.3.1 Weighting the Dependent Variable

In practice, we often encounter the following complication when carrying out regression for systems: different components of $\vec{y}$ may take values on entirely different scales. For example, a common system encounted in ecology is a plant-herbivore interaction, in which

case y_1 might denote a measure of vegetation (measured by something like total mass, mass density, acreage covered etc.), while y_2 denotes a measure of herbivore presence (measured by something like total number of herbivores, herbivore density, etc.). Since the units of measurement aren't comparable, it may be the case that one set of values is much less than the others. For purposes of this discussion, let's suppose the values of y_1 are substantially smaller than those of y_2, and let's express the SSR as

$$\begin{aligned} E(\vec{p}) &= \sum_{k=1}^{N}(y_{k1} - f_1(\vec{x}_k;\vec{p}))^2 + \sum_{k=1}^{N}(y_{k2} - f_2(\vec{x}_k;\vec{p}))^2 \\ &=: E_1(\vec{p}) + E_2(\vec{p}). \end{aligned}$$

Generally, we will find that $E_1(\vec{p})$ is much smaller than $E_2(\vec{p})$, and so the parameter values will be determined almost entirely by the values of y_2. I.e., the model will fit the herbivore population well, but will not fit the plant population well. On an intuitive level, we can see this easily: if the values of y_1 and y_2 are treated equally, then the error on each will be the same, but this error will be a much higher percentage of the values of y_1 than of the values of y_2.

We typically address this situation by scaling the dependent variables with "weights" that capture in some sense the general sizes of the variables. We will denote these weights w_1, w_2, etc., so for a system of two equations we specify two weighted dependent variables

$$z_1 = \frac{y_1}{w_1}, \quad z_2 = \frac{y_2}{w_2}.$$

In this case,

$$\begin{aligned} z_1 &= \frac{y_1}{w_1} = \frac{1}{w_1} f_1(\vec{x};\vec{p}) =: g_1(\vec{x};\vec{p}) \\ z_2 &= \frac{y_2}{w_2} = \frac{1}{w_2} f_2(\vec{x};\vec{p}) =: g_2(\vec{x};\vec{p}). \end{aligned}$$

With scaled data $\{(\vec{x}_k, z_{k1}, z_{k2})\}_{k=1}^N = \{(\vec{x}_k, \frac{y_{k1}}{w_1}, \frac{y_{k2}}{w_2})\}_{k=1}^N$, we can work with the new weighted SSR

$$\begin{aligned} \tilde{E}(\vec{p}) &= \sum_{k=1}^{N}(z_{k1} - g_1(\vec{x}_k;\vec{p}))^2 + \sum_{k=1}^{N}(z_{k2} - g_2(\vec{x}_k;\vec{p}))^2 \\ &= \sum_{k=1}^{N}(\frac{y_{k1}}{w_1} - \frac{f_1(\vec{x}_k;\vec{p})}{w_1})^2 + \sum_{k=1}^{N}(\frac{y_{k2}}{w_1} - \frac{f_2(\vec{x}_k;\vec{p})}{w_2})^2 \\ &= \sum_{k=1}^{N}\Big\{(y_{k1} - f_1(\vec{x}_k;\vec{p}))^2/w_1^2 + (y_{k2} - f_2(\vec{x}_k;\vec{p}))^2/w_2^2\Big\}. \end{aligned}$$

More generally, the weighted SSR will be

$$\tilde{E}(\vec{p}) = \sum_{k=1}^{N}\sum_{i=1}^{n}(y_{ki} - f_i(\vec{x}_k;\vec{p}))^2/w_i^2,$$

and we will typically omit the tilde if we're not specifically comparing the weighted SSR to an unweighted SSR. For our matrix formulation, we let the vectors $\{\vec{Y}_i\}_{i=1}^n$ and $\{F_i\}_{i=1}^n$ be as in (2.8), and set

$$\vec{Y} := \begin{pmatrix} \vec{Y}_1/w_1 \\ \vec{Y}_2/w_2 \\ \vdots \\ \vec{Y}_n/w_n \end{pmatrix} \in \mathbb{R}^{nN}, \quad F = \begin{pmatrix} F_1/w_1 \\ F_2/w_2 \\ \vdots \\ F_n/w_n \end{pmatrix} \in \mathbb{R}^{nN \times m}.$$

Then we can find the regression values for $\vec{p}$ by thinking in the usual way about the system

$$F\vec{p} = \vec{Y}.$$

One caveat here is that if we could solve the system $F\vec{p} = \vec{Y}$ exactly, the weights would all cancel out and would turn out to be unnecessary, which is fair, since in that case we would have an exact fit for every component. As usual, the best-fit parameter values can generally be computed as $\vec{p} = (F^T F)^{-1} F^T \vec{Y}$.

Of course, in order to complete our discussion of using weights, we need a method for actually choosing the weights. For example, for our plant-herbivore system we might scale each variable by some maximum size so that y_1 and y_2 are both confined to the interval $[0, 1]$, and thereby presumably more comparable. While any number of such ad hoc choices can work in a sense, a more systematic approach, justified in part by statistical considerations, is to weight the dependent variables by their sample standard deviations. I.e., to weight y_i by

$$w_i = \sqrt{\frac{1}{N-1} \sum_{k=1}^{N} (y_{ki} - \mu_{y_i})^2}, \quad i = 1, 2, \ldots, n.$$

Working a bit loosely, we can understand how this is a reasonable choice by thinking in terms of the adjusted coefficient of determination for y_i, $\bar{R}_i^2$. First, the sample variance for the variable y_i is

$$w_i^2 = \frac{1}{N-1} \sum_{k=1}^{N} (y_{ki} - \mu_{y_i})^2 = \frac{T_i}{N-1},$$

where T_i is the total sum of squares for the component y_i as defined in (2.6). Likewise, the variance on predictions of y_i can reasonably be approximated by

$$s_i^2 = \frac{1}{N - m_i} E_i,$$

where m_i denotes the number of parameters that appear in the equation for y_i, and E_i denotes the SSR for the i^{th} equation, namely

$$E_i(\vec{p}) = \sum_{k=1}^{N} (y_{ki} - f_i(\vec{x}_k; \vec{p}))^2.$$

(Here, s_i^2 would be precisely the usual variance we use, if we were obtaining the values $\vec{p}$ only from the i^{th} equation; as it is, we are obtaining $\vec{p}$ from the full system, so this is only

an intuitive approximation.) If we add a weight w_i in our calculation of E_i, then s_i^2 will be replaced by s_i^2/w_i^2, and if w_i^2 is chosen to be sample variance, this will give

$$\frac{s_i^2}{w_i^2} = \frac{\frac{1}{N-m_i}E_i}{\frac{1}{N-1}T_i} = \frac{N-1}{N-m_i}\frac{E_i}{T_i} = 1 - \bar{R}_i^2. \tag{2.9}$$

The variance on predictions for the full fit is

$$s^2 = \frac{1}{Nn-m}\sum_{k=1}^{N}\sum_{i=1}^{n}(y_{ki} - f_i(\vec{x}_k;\vec{p}))^2/w_i^2,$$

and this is related to the individual variances by $s_i = sw_i$. I.e.,

$$\frac{y_i}{w_i} = \text{prediction } \pm s$$

so that

$$y_i = \text{prediction } * w_i \pm sw_i.$$

We see that the ratio $\frac{s_i}{w_i}$ is the same for all components, and if w_i is specified as above so that (2.9) holds, then it follows that $\bar{R}_i^2$ is the same for all components as well. In particular, with this choice of weights, the equation for y_1 is no more or less effective, as measured by the adjusted coefficient of determination, than the equation for y_2, and similarly for all other pairs of equations.

2.3.2 MATLAB Implementation

Suppose we have a system of two equations

$$\begin{aligned} y_1 &= p_1 + p_2x_1 + p_3x_2 \\ y_2 &= p_4 + p_3x_1 + p_5x_2, \end{aligned}$$

where p_3 appears in both equations. Data for this system is given below in the MATLAB script M-file *sysreg.m*, in which values for the parameters are obtained by weighted linear least squares regression. In particular, the dependent variables y_1 and y_2 are weighted by standard deviation as described above.

```
%SYSREG: MATLAB script M-file with an example of
%regression for a linear system.
%
%Define the data
[x1 x2]=meshgrid(linspace(-1,1,5),linspace(-1,1,5));
y1 = [-3.8656 -3.3269 -2.3375 -1.0512 0.1679;
-2.0415 -1.6084 0.2587 0.4690 1.1981;
-1.5647 0.0857 1.1814 2.3724 3.1793;
0.7155 2.3946 2.4842 3.8523 4.9076;
2.0797 3.6924 4.1787 5.3543 6.1222];
```

```
%
y2 = [-17.3183 -3.8368 -15.4453 -8.0703 -10.0780;
12.3868 9.6517 15.1628 9.0420 21.7404;
30.9294 40.3569 42.7626 37.7885 42.0379;
56.4325 62.3721 70.5031 61.1921 72.4431;
86.9658 94.0868 97.7211 103.2523 89.1758];
%
%Define dependent vectors; use [],1 to select a single column
%and any number of rows (for portability)
Y1 = reshape(y1,[],1);
Y2 = reshape(y2,[],1);
%Define columns of ones and columns of zeros
col1 = ones(size(reshape(y1,[],1)));
col0 = zeros(size(reshape(y1,[],1)));
%Use standard deviation as weights
w1=std(Y1);
w2=std(Y2);
%
%Define design matrix
F1=[col1 reshape(x1,[],1) reshape(x2,[],1) col0 col0];
F2=[col0 col0 reshape(x1,[],1) col1 reshape(x2,[],1)];
%
F = [F1/w1;F2/w2]; Y=[Y1/w1;Y2/w2];
p = F\Y
%Standard deviation
q = length(Y)-length(p);
ssr = norm(Y-F*p)^2
s = sqrt(ssr/q);
s1 = s*w1
s2 = s*w2
%Adjusted coefficient of determination
Ybar = mean(Y);
T = norm(Y-Ybar)^2
Rsq = 1-ssr/T
Rsqa = 1-(1-Rsq)*(length(Y)-1)/q
```

In the top lines of *sysreg.m*, the data is given in a grid format, and the *reshape* command is simply used to convert the values into a pair of column vectors Y1 and Y2, corresponding precisely with the vectors $\vec{Y}_1$ and $\vec{Y}_2$ in our general development. The output from *sysreg.m* is given below.

```
>>sysreg
p =
1.1559
2.0105
```

3.1930
40.0502
52.2744
ssr =
0.7014
s1 =
0.3421
s2 =
4.7592
T =
52.9417
Rsq =
0.9868
Rsqa =
0.9856

We see that the parameter values are estimated to be

$$\begin{aligned} p_1 &= 1.1559 \\ p_2 &= 2.0105 \\ p_3 &= 3.1930 \\ p_4 &= 40.0502 \\ p_5 &= 52.2744. \end{aligned}$$

The estimated standard deviations on predictions are

$$\begin{aligned} s_1 &= .3421 \\ s_2 &= 4.7592, \end{aligned}$$

effectively distinguising between the relatively small values associated with y_1 and the relatively large values associated with y_2.

3 Nonlinear Least Squares Regression

For most applications, we will model data $\{(\vec{x}_k, \vec{y}_k)\}_{k=1}^N$ with a relation

$$\vec{y} = \vec{f}(\vec{x}; \vec{p})$$

that is nonlinear in the parameters $\vec{p}$. For example, we've briefly discussed the relation

$$y = p_1 e^{p_2 x},$$

which arises commonly as a model of population growth, radioactive decay, and other processes in which the rate of change of the quantity under investigation (i.e., $\frac{dy}{dx}$) is proportional

to the value of the quantity itself[16]. Just as with linear least squares regression, we proceed by minimizing the SSR

$$E(\vec{p}) = \sum_{k=1}^{N} |\vec{y}_k - \vec{f}(\vec{x}_k; \vec{p})|^2,$$

or more generally a weighted version of this SSR. The main new difficulty we encounter with nonlinear regression is that $E(\vec{p})$ can now have multiple local minima.

Example 3.1. Consider the single-parameter nonlinear relation

$$y = \sin(px),$$

for which data $\{(x_k, y_k)\}_{k=1}^{N}$ will be randomly generated (see the MATLAB M-file *nlssr.m* below). The SSR for this relation and data is

$$E(p) = \sum_{k=1}^{N} (y_k - \sin(px_k))^2,$$

and E is plotted against p for a particular set of data in Figure 3.1. In this case, the data was generated with the parameter value $p = 2$, and we see that the global minimum of E is indeed near $p = 2$. However, it's clear that any numerical method for identifying this global minimum must be able to avoid becoming trapped near a local minimum. △

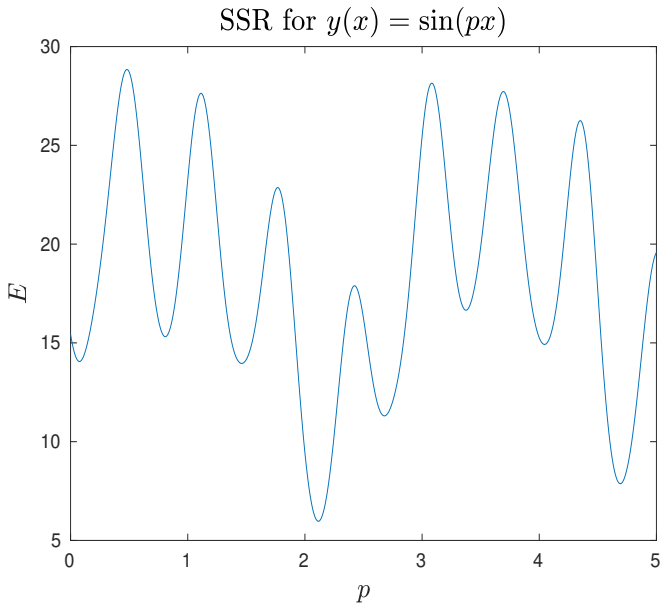


Figure 3.1: SSR for a relation nonlinear in a single parameter.

Example 3.2. Consider the two-parameter nonlinear relation

$$y = \sin(p_1 x) + \cos(p_2 x),$$

[16]Precisely, $y = p_1 e^{p_2 x}$ solves the differential equation $\frac{dy}{dx} = p_2 y$ subject to the initial condition $y(0) = p_1$.

for which data $\{(x_k, y_k)\}_{k=1}^N$ will be randomly generated (see the MATLAB M-file *nlssr.m* below). The SSR for this relation and data is

$$E(\vec{p}) = \sum_{k=1}^{N} (y_k - \sin(p_1 x_k) - \cos(p_2 x_k))^2,$$

and E is plotted against $\vec{p}$ for a particular set of data in Figure 3.2. In this case, by using the 3D-panning feature in MATLAB's figure window it's possible to check that the global minimizer is near the point $(p_1, p_2) = (2, 3)$, which was used in the generation of the data. For two parameters, the proliferation of local minima is already striking, and this effect generally becomes more pronounced as the number of parameters increases. △

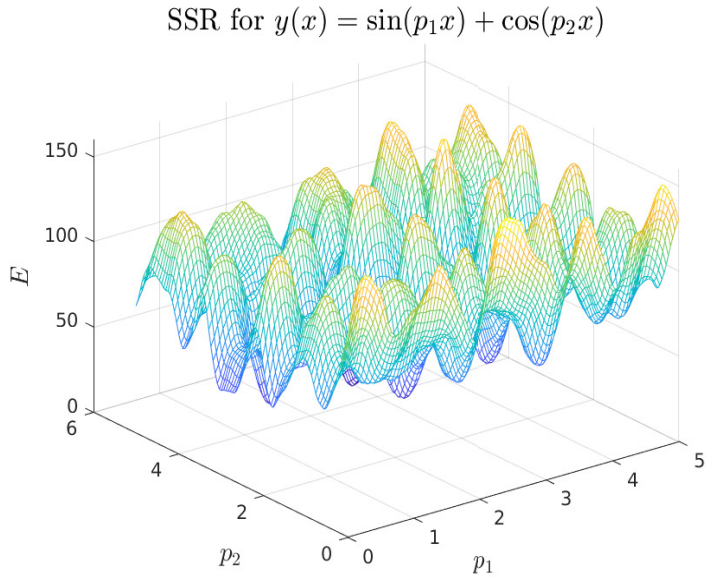


Figure 3.2: SSR for a relation nonlinear in two parameters.

The data and fits for Examples 3.1 and 3.2 were generated with the MATLAB script M-file *nlssr.m*, included below.

```
%NLSSR: MATLAB script M-file written to plot
%sums of squared residuals for some example
%nonlinear fits.
%
%Single parameter, y = sin(px)
%
%GENERATE RANDOM DATA
x = 5*randn(1,5)+2;
p = 2; %FOR GENERATING Y
y = sin(p*x)+randn(1,5);
avals = linspace(0,5,1000);
```

```
for k = 1:length(avals)
E1(k) = norm(y - sin(avals(k)*x))^2;
end
plot(avals, E1)
title('SSR for $y(x) = \sin (px)$','Interpreter','latex','Fontsize',16)
xlabel('$p$','Interpreter','latex','Fontsize',14)
ylabel('$E$','Interpreter','latex','Fontsize',14)
%
pause
%Two parameters, y = sin(p1*x)+cos(p2*x)
x = 5*randn(1,25)+2;
p1 = 2; p2 = 3; %FOR GENERATING Y
y = sin(p1*x)+cos(p2*x)+randn(1,25);
p1vals = linspace(0,5,100);
p2vals = linspace(0,5,100);
for k=1:length(p1vals)
for j = 1:length(p2vals)
E2(j,k) = norm(y - sin(p1vals(k)*x) - cos(p2vals(j)*x))^2;
end
end
[p1grid,p2grid]=meshgrid(p1vals,p2vals);
figure
mesh(p1grid,p2grid,E2);
title('SSR for $y(x) = \sin (p_1 x) + \cos(p_2 x)$','Interpreter','latex','Fontsize',16)
xlabel('$p_1$','Interpreter','latex','Fontsize',14)
ylabel('$p_2$','Interpreter','latex','Fontsize',14)
zlabel('$E$','Interpreter','latex','Fontsize',14)
```

For nonlinear regression, there is no general solution formula[17], so we proceed by minimizing the SSR numerically. Numerical methods for carrying out this minimization efficiently can be quite sophisticated, but most are centered on the simple idea of *gradient descent*. For this, we fix some initial approximation $\vec{p}_0$ of our parameter vector $\vec{p}$, and we evaluate the gradient of $E(\vec{p})$ at $\vec{p}_0$, namely

$$D_pE(\vec{p}_0) = (\frac{\partial E}{\partial p_1}(\vec{p}_0), \frac{\partial E}{\partial p_2}(\vec{p}_0), \ldots, \frac{\partial E}{\partial p_m}(\vec{p}_0)).$$

This vector indicates the direction (in $\mathbb{R}^m$) in which $\vec{p}$ could be moved to maximize the increase in $E(\vec{p})$, so we take a small step in the opposite direction. I.e., we update our initial approximation $\vec{p}_0$ to

$$\vec{p}_1 = \vec{p}_0 - \epsilon D_pE(\vec{p}_0),$$

where $\epsilon > 0$ is suitably small (and is generally taken to vary as the minimizer is approached). Repeating, we obtain the recursion relation

$$\vec{p}_{n+1} = \vec{p}_n - \epsilon_n D_pE(\vec{p}_n),$$

[17] I.e., no analogue to $\vec{p} = (F^TF)^{-1}F^T\vec{y}$ for the linear case.

where the index on ϵ indicates loosely that it will vary from step to step, setting aside the complicated question of how to optimize its values. It's clear from these considerations and the multiple local minimizers we see in Examples 3.1 and 3.2 that a key element of nonlinear regression is choosing an appropriate starting approximation $\vec{p}_0$. In general, this can be a difficult step, but in most of the examples relevant for us we will be able to take advantage of the following simple observation: *while solutions to ODE and PDE are generally nonlinear in the parameters, the ODE and PDE themselves are often linear in the parameters.* We will see how this plays out in the following important example.

Example 3.3. We will fit the *logistic* population model

$$\frac{dy}{dt} = ry(1 - \frac{y}{K}), \quad y(0) = y_0, \tag{3.1}$$

to US census data 1790-2020. Here, r is called the "growth rate" of the population, K is called the "carrying capacity," and y_0 denotes the initial number of individuals in the population. We observe that for this model the population is assumed to have a maximum possible number of individuals, K. We can see this by observing that if $y(0) < K$, then as $y(t)$ increases toward K, $\frac{dy}{dt}$ decreases to 0, so that the growth gets slower and slower. The population never actually reaches K, but rather approaches it asymptotically in large time. (This behavior can be seen directly from the explicit solution given just below.) On the other hand, if the population starts above the value K, then it will decline to K as t increases. Equation (3.1) can be solved by separation of variables and partial fractions, and we find

$$y(t; r, K, y_0) = \frac{y_0 K}{(K - y_0)e^{-rt} + y_0}. \tag{3.2}$$

We will take year 0 to be 1790, and we will assume the estimate that year was fairly crude and obtain a value of y_0 by fitting the entirety of the data. In this way, we have three parameters to contend with, r, K, and y_0, and it's clear that the relation (3.2) is nonlinear in each of these. The data we will use for this fit is given in *uspop.m*.

```
%USPOP: Defines decades and corresponding U.S. populations.
%Year 0 corresponds with the first census in 1790,
%and year 230 corresponds with the most recent census in 2020.
decades=0:10:230;
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...
91.97 105.71 122.78 131.67 151.33 179.32 203.21 226.5 249.63 281.42 308.75...
331.45];
```

Since the regression will be nonlinear, we will need appropriate initial approximations for the values r, K, and y_0. In order to identify these initial approximations, we observe that the logistic equation (3.1) can be expressed as

$$\frac{dy}{dt} = ry - \frac{r}{K}y^2,$$

which can be viewed as a relation between $\frac{dy}{dt}$ and two variables y and y^2, linear in the parameters r and $q_2 := -\frac{r}{K}$. Even better, if we divide by y we obtain the relation

$$\frac{1}{y}\frac{dy}{dt} = r - \frac{r}{K}y,$$

which can be viewed as a line relation between the variables $Y = \frac{1}{y}\frac{dy}{dt}$ and $X = y$, with slope $m = -\frac{r}{K}$ and intercept $b = r$. We will proceed by fitting Y as a function of X, but for this we need data-based values for the derivatives $\frac{dy}{dt}$, and these aren't generally known. We proceed by approximation.

3.1 Approximating Derivatives

In order to understand methods for approximating derivatives, we recall that the Taylor polynomial with remainder for a sufficiently differentiable function $f(x)$ near a value a can be expressed as

$$\begin{aligned} f(x) &= f(a) + f'(a)(x-a) + \frac{f''(a)}{2}(x-a)^2 + \cdots + \frac{f^{(n)}(a)}{n!}(x-a)^n + \frac{f^{(n+1)}(c)}{(n+1)!}(x-a)^{n+1} \\ &= \sum_{k=0}^{n} \frac{f^{(k)}(a)}{k!}(x-a)^k + \frac{f^{(n+1)}(c)}{(n+1)!}(x-a)^{n+1}, \end{aligned}$$

where c is some value between a and x (and so varies with both).[18]

3.1.1 Forward Difference Derivative Approximation

Given a sufficiently differentiable function $y(t)$ and a discrete set of times $\{t_k\}_{k=1}^N$, we can Taylor expand $y(t_{k+1})$ about $y(t_k)$ to see that

$$y(t_{k+1}) = y(t_k) + y'(t_k)(t_{k+1} - t_k) + \frac{y''(c)}{2}(t_{k+1} - t_k)^2,$$

where c is some value between t_k and t_{k+1}. If $y''(c)$ is bounded on the interval $[t_k, t_{k+1}]$, then there exists a constant C so that

$$\left|\frac{y''(c)}{2}(t_{k+1} - t_k)^2\right| \leq C(t_{k+1} - t_k)^2$$

for all $c \in [t_k, t_{k+1}]$. For brevity, we often introduce "big-O" notation, and write

$$\frac{y''(c)}{2}(t_{k+1} - t_k)^2 = \mathbf{O}((t_{k+1} - t_k)^2).$$

Definition 3.1. *For a real-valued function $f(x)$ defined in an open interval containing a value $x_0 \in \mathbb{R}$, we write $f = \mathbf{O}(|x - x_0|^k)$ if there exists a constant C so that*

$$|f(x)| \leq C|x - x_0|^k$$

for all x sufficiently close to x_0.

Here, Definition 3.1 is made for a function f taking real values to real values, but it immediately extends to functions taking complex values to complex values. With this notation, we can write

$$y(t_{k+1}) = y(t_k) + y'(t_k)(t_{k+1} - t_k) + \mathbf{O}((t_{k+1} - t_k)^2).$$

[18] See Theorem A.2 in the appendix for a precise statement of Taylor polynomials with remainder.

Solving for $y'(t_k)$, we find that

$$y'(t_k) = \frac{y(t_{k+1}) - y(t_k)}{t_{k+1} - t_k} - \frac{\mathbf{O}((t_{k+1} - t_k)^2)}{t_{k+1} - t_k}.$$

If follows from the specification of the big-O notation that we can write

$$-\frac{\mathbf{O}((t_{k+1} - t_k)^2)}{t_{k+1} - t_k} = \mathbf{O}(|t_{k+1} - t_k|),$$

leading to the convenient relation

$$y'(t_k) = \frac{y(t_{k+1}) - y(t_k)}{t_{k+1} - t_k} + \mathbf{O}(|t_{k+1} - t_k|).$$

We see that as long as the step size $|t_{k+1} - t_k|$ is sufficiently small, the derivative $y'(t_k)$ will be well approximated by the "forward difference" derivative approximation

$$y'(t_k) \cong \frac{y(t_{k+1}) - y(t_k)}{t_{k+1} - t_k}.$$

This suggests we can approximate data values $\{(\frac{dy}{dt})_k\}_{k=1}^{N-1}$ by computing

$$(\frac{dy}{dt})_k = \frac{y_{k+1} - y_k}{t_{k+1} - t_k}, \quad k = 1, 2, \ldots, N-1.$$

In practice, we can readily implement this approximation in MATLAB using the built-in *diff* command, which takes a vector $\vec{y} \in \mathbb{R}^N$ as input and returns a vector of differences

$$\text{diff}(\vec{y}) = (y_2 - y_1, y_3 - y_2, \ldots, y_N - y_{N-1}) \in \mathbb{R}^{N-1}.$$

This allows use to compute the vector of data values $\{(\frac{dy}{dt})_k\}_{k=1}^{N-1}$ with the command

$$\text{diff}(y)./\text{diff}(t)$$

In the MATLAB M-file *uspop1.m*, we use this approximation to fit values $Y_k = \frac{1}{y_k}(\frac{dy}{dt})_k$ against $X_k = y_k$, and we obtain values of r and K from the resulting slope and intercept. It's important to notice that in this case, the differences $t_{k+1} - t_k$ are not small (we have $t_{k+1} - t_k = 10$ for all $k = 1, 2, \ldots, N-1$), and yet the resulting fit is still fairly promising as an initial step prior to nonlinear regression.

```
%USPOP1: MATLAB script M-file that uses a
%forward difference derivative approximation
%to approximate parameter values for a logistic
%model to U.S. population growth.
%
%Define data
uspop;
%
```

```
dydt = diff(pops)./diff(decades);
plot(pops(1:end-1),dydt./pops(1:end-1),'o')
title({'Plot of per capita growth vs populations' 'forward differences'},...
'Interpreter','latex','FontSize',16)
xlabel('Populations','Interpreter','latex','FontSize',14)
ylabel('Per capital growth rate','Interpreter','latex','FontSize',14)
pause
p=polyfit(pops(1:end-1),dydt./pops(1:end-1),1);
r=p(2)
K= -r/p(1)
pause
%
%Plot regression line
plot(pops(1:end-1),dydt./pops(1:end-1),'o',pops(1:end-1),p(1)*pops(1:end-1)+p(2))
title({'Plot of per capita growth vs populations', 'forward differences, ...
with regression line'},'Interpreter','latex','FontSize',16)
xlabel('Populations','Interpreter','latex','FontSize',14)
ylabel('Per capital growth rate','Interpreter','latex','FontSize',14)
pause
%
%Plot ODE solution vs data
y0 = pops(1);
logistic = @(t) y0*K./(y0+(K-y0)*exp(-r*t));
modelpops = logistic(decades);
plot(decades, pops, 'o', decades, modelpops)
title({'Plot of data and model', 'parameters found by forward differences'},...
'Interpreter','latex','FontSize',16)
xlabel('Decades, 1790-2020','Interpreter','latex','FontSize',14)
ylabel('U. S. Populations','Interpreter','latex','FontSize',14)
```

The linear fit obtained by this code is given on the left-hand side of Figure 3.3, while the fit of the logistic model along with data is given on the right. The parameter values obtained are

$$\begin{aligned} r &= .0314 \\ K &= 335.3112. \end{aligned}$$

3.1.2 Central Difference Derivative Approximation

Suppose the independent variables are evenly spaced, so that

$$h = t_{k+1} - t_k, \quad \forall\, k = 1, 2, \ldots, N-1.$$

Proceeding similarly as with the forward-difference derivative approximation, we find that

$$y'(t_k) = \frac{y(t_k + h) - y(t_k - h)}{2h} + \mathbf{O}(h^2), \quad k = 2, 3, \ldots, N-1,$$

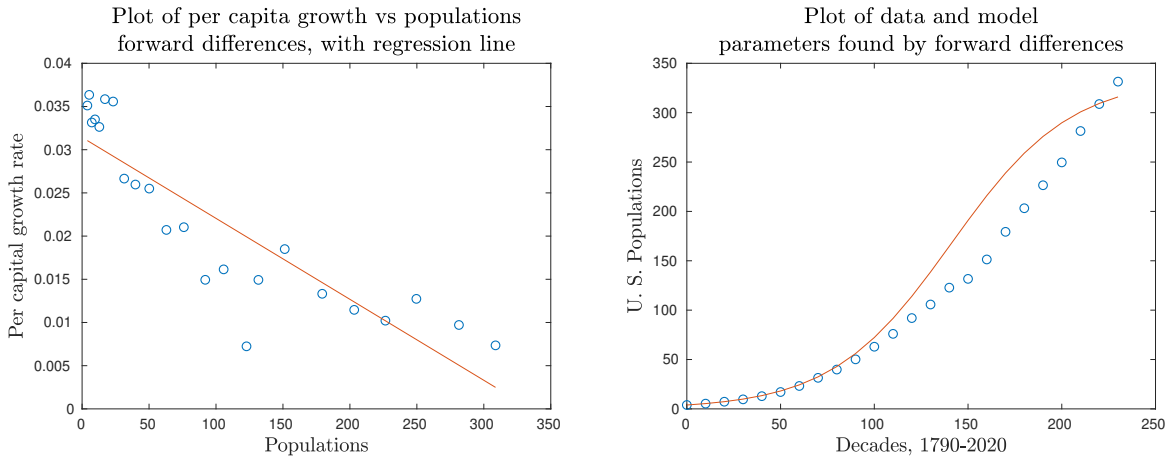


Figure 3.3: Figures associated with our fit of US population data to the logistic model, based on the forward difference derivative approximation.

which, for small values of h, is better than the forward difference approximation (for which the error would be $\mathbf{O}(h)$). This suggests that in some cases it can be useful to approximate values $\{(\frac{dy}{dt})_k\}_{k=2}^{N-1}$ with the "central difference" derivative approximation

$$(\frac{dy}{dt})_k = \frac{y_{k+1} - y_{k-1}}{2h}, \quad k = 2, 3, \ldots, N-1.$$

We carry this out with *uspop2.m*, from which we find

$$r = .0277$$
$$K = 362.3526.$$

```
%USPOP2: MATLAB script M-file that uses a
%central difference derivative approximation
%to approximate parameter values for a logistic
%model to U.S. population growth
%
%Define data
uspop;
%
dydt = (pops(3:end)-pops(1:end-2))/20;
plot(pops(2:end-1),dydt./pops(2:end-1),'o')
title({'Plot of per capita growth vs populations', 'central differences'},...
'Interpreter','latex','FontSize',16)
xlabel('Populations','Interpreter','latex','FontSize',14)
ylabel('Per capital growth rate','Interpreter','latex','FontSize',14)
pause
p=polyfit(pops(2:end-1),dydt./pops(2:end-1),1);
r=p(2)
K= -r/p(1)
```

```
pause
%
%Plot regression line
plot(pops(2:end-1),dydt./pops(2:end-1),'o',pops(2:end-1),p(1)*pops(2:end-1)+p(2))
title({'Plot of per capita growth vs populations',...
'central differences, with regression line'},...
'Interpreter','latex','FontSize',16)
xlabel('Populations','Interpreter','latex','FontSize',14)
ylabel('Per capital growth rate','Interpreter','latex','FontSize',14)
pause
%
%Plot ODE solution vs data
y0 = pops(1);
logistic = @(t) y0*K./(y0+(K-y0)*exp(-r*t));
modelpops = logistic(decades);
plot(decades, pops, 'o', decades, modelpops)
title({'Plot of data and model', 'parameters found by central differences'},...
'Interpreter','latex','FontSize',16)
xlabel('Decades, 1790-2020','Interpreter','latex','FontSize',14)
ylabel('U. S. Populations','Interpreter','latex','FontSize',14)
```

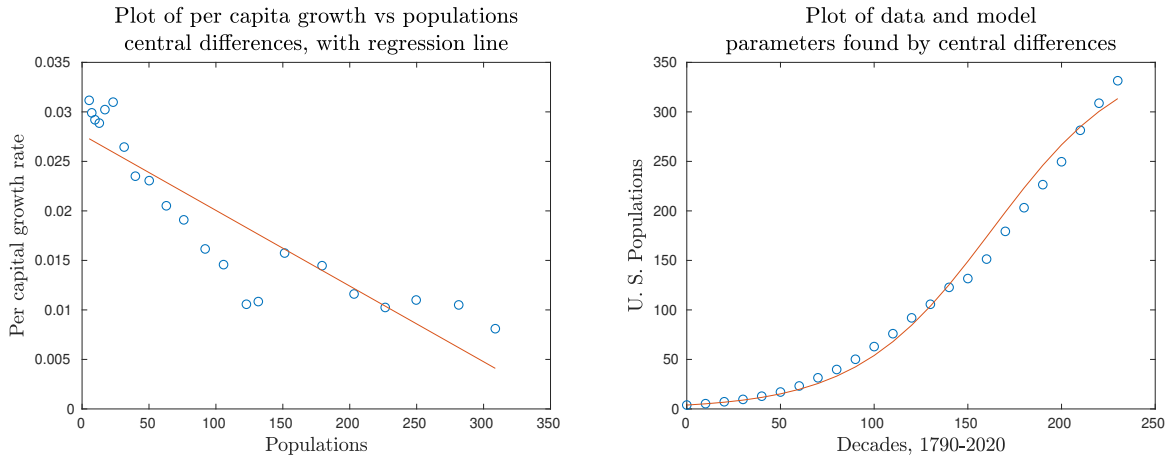


Figure 3.4: Figure associated with our fit of US population data to the logistic model, based on the central difference derivative approximation.

3.1.3 The Nonlinear Fit

We will carry out the nonlinear fit for Example 3.3 in two different ways, using MATLAB's built-in M-files *lsqcurvefit.m* and *fminsearch.m*. The program *lsqcurvefit.m* is designed especially for regression, but we will find that *fminsearch.m* is ultimately more general. Using

(3.2), we can express the nonlinear SSR for this example as

$$E(r, K, y_0) = \sum_{k=1}^{N} (y_k - \frac{y_0 K}{y_0 + (K - y_0)e^{-rt_k}})^2,$$

and our goal is to find the triple (r, K, y_0) that minimizes this function. First, for *lsqcurvefit.m*, the general syntax is as follows:

>>[p, ssr] = lsqcurvefit(function, i.p.v., xdata, ydata, l.b.p., u.b.p., options),

where

i.p.v. = initial parameter values
l.b.p. = lower bound on parameters
u.b.p. = upper bound on parameters,

with the entrees l.b.p., u.b.p., and options being optional. Nonlinear regression for Example 3.3 is carried out using *lsqcurvefit.m* in *uspop3.m*.

```
%USPOP3: MATLAB script M-file that uses MATLAB's built-in
%least squares regression function lsqcurvefit to find
%least squares values for r, K, and y0, for a logistic
%fit of U.S. population data
%
%define data
uspop;
%Define logistic solution
y = @(p,t) p(2)*p(3)./(p(3)+(p(2)-p(3))*exp(-p(1)*t));
%Initial estimate of parameters
%Forward differences
%p0 = [.0314 335.3112 pops(1)];
%Central differences
p0=[.0277 362.3526 pops(1)];
[p ssr]=lsqcurvefit(y,p0,decades,pops)
N=length(decades); m = length(p);
q = N-m;
sd=sqrt(ssr/q)
%Adjusted coefficient of determination: keep in mind that
%it's not entirely clear what this means for nonlinear regression.
popsbar = mean(pops);
T = norm(pops-popsbar)^2;
Rsq = 1-ssr/T
Rsqa = 1-(1-Rsq)*(length(decades)-1)/q
pause
%
%Plot model along with data
```

```
modelpops = y(p,decades);
plot(decades,pops,'o',decades,modelpops)
title({'Plot of data and model', 'parameters found by nonlinear regression'},...
'Interpreter','latex','FontSize',16)
xlabel('Decades, 1790-2020','Interpreter','latex','FontSize',14)
ylabel('U. S. Populations','Interpreter','latex','FontSize',14)
%Prediction for 2020
disp(['The prediction for 2030 is ' num2str(y(p,240)) ' +/- ' num2str(sd)])
```

MATLAB's output is as follows.

```
>>uspop3
p =
0.0205 500.6513 8.3979
ssr =
503.7926
sd =
4.8980
Rsq =
0.9980
Rsqa =
0.9978
The prediction for 2030 is 351.8384 +/- 4.898
```

In this setting, the coefficient of determination and adjusted coefficient of determination aren't particularly important, but the parameter values obtained are reasonable, and the ultimate fit is quite good (see Figure 3.5). The parameter values are

$$\begin{aligned} r &= .0205 \\ K &= 500.6513 \\ y_0 &= 8.3979. \end{aligned}$$

Next, the general syntax for *fminsearch.m* is as follows:

```
>>[p, ssr] = fminsearch(ssr function, initial parameter value, options)
```

We observe, in particular, that while *lsqcurvefit.m* takes as input the function to be fit, *fminsearch.m* takes as input the SSR to be minimized. Nonlinear regression for Example 3.3 is carried out using *fminsearch.m* in *uspop3a.m*.

```
%USPOP3A: MATLAB script M-file that uses MATLAB's built-in
%function fminsearch to to find least squares values for r,
%K, and y0, for a logistic fit of U.S. population data
%
%define data
uspop;
%Define logistic solution
```

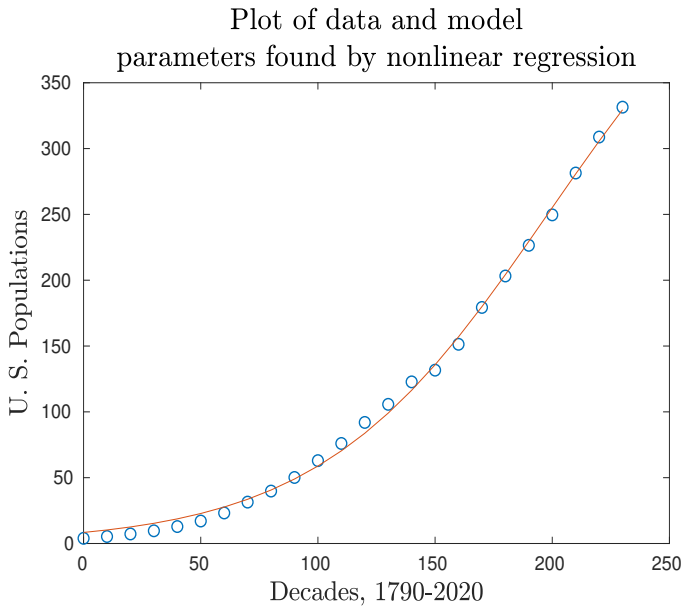


Figure 3.5: US population data along with nonlinear logistic fit.

```
y = @(p,t) p(2)*p(3)./(p(3)+(p(2)-p(3))*exp(-p(1)*t));
%Define the error
E = @(p) norm(pops-y(p,decades))^2;
%Initial estimate of parameters
%Forward differences
%p0 = [.0314 335.3112 pops(1)];
%Central differences
p0=[.0277 362.3526 pops(1)];
[p ssr]=fminsearch(E,p0)
N=length(decades); m = length(p);
q = N-m;
sd=sqrt(ssr/q)
pause
%
%Plot model along with data
modelpops = y(p,decades);
plot(decades,pops,'o',decades,modelpops)
title({'Plot of data and model', 'parameters found by nonlinear regression'},...
'Interpreter','latex','FontSize',16)
xlabel('Decades, 1790-2020','Interpreter','latex','FontSize',14)
ylabel('U. S. Populations','Interpreter','latex','FontSize',14)
pause
%Prediction for 2030
disp(['The prediction for 2030 is ' num2str(y(p,240)) ' +/- ' num2str(sd)])
```

The MATLAB output is as follows:

```
>>uspop3a
p =
0.0205 500.6525 8.3979
ssr =
503.7926
sd =
4.8980
The prediction for 2030 is 351.8385 +/- 4.898
```

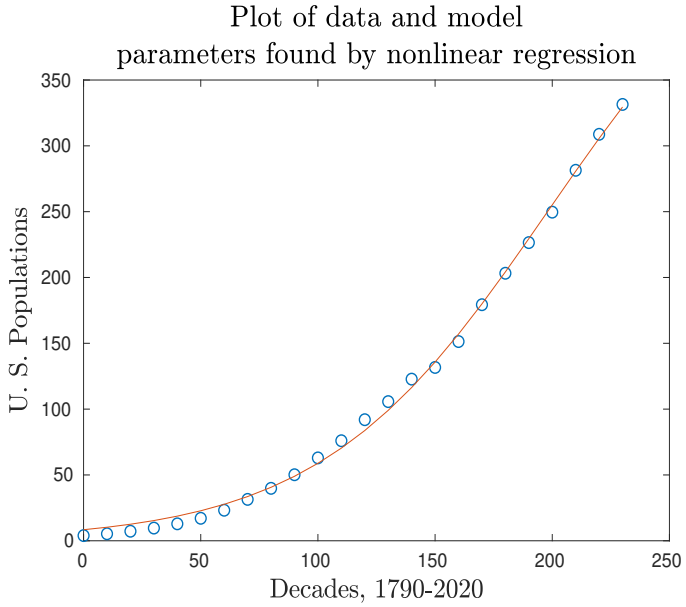


Figure 3.6: US population data along with nonlinear logistic fit.

3.2 Multiple Independent Variables

Suppose we have a relation

$$y = p_1 x_1^{p_2} e^{p_3 x_2},$$

along with data $\{(\vec{x}, y_k)\}_{k=1}^{N}$. We can again find least-squares regression values for the parameters with either *fminsearch.m* or *lsqcurvefit.m*. In either case, we begin by taking a natural logarithm of the relation to obtain

$$\ln y = \ln p_1 + p_2 \ln x_1 + p_3 x_2,$$

and obtaining approximate parameter values by fitting $Y = \ln y$ as a function of $\ln x_1$ and x_2, linear in the parameters $\ln p_1$, p_2, and p_3. We then carry out the nonlinear fit. This is done with *fminsearch.m* in *nlreg1.m* and by *lsqcurvefit.m* in *nlreg1a.m*. (In both cases, randomly generated data is stored in *nlregdata1.mat*.)

```
%NLREG1: Matlab script M-file for analyzing the
%data stored in nlregdata1
%Data has the form y, x1, x2
load nlregdata1
%First, carry out a linear fit of a transformed equation
%log y = log p1 + p2 log x1 + p3 x2
F = [ones(size(y)) log(x1) x2];
ptemp = F\log(y)
p0 = [exp(ptemp(1)) ptemp(2) ptemp(3)]
%
%Define the error
E = @(p) norm(y-p(1)*x1.^p(2).*exp(p(3)*x2))^2;
%Minimize the error with fminsearch
[p ssr]=fminsearch(E,p0)
q = length(y)-length(p);
sd = sqrt(ssr/q)
```

and

```
%NLREG1A: Matlab script M-file for analyzing the
%data stored in nlregdata1
%Data has the form y, x1, x2. Uses lsqcurvefit.
load nlregdata1
%First, carry out a linear fit of a transformed equation
%log y = log p1 + p2 log x1 + p3 x2
F = [ones(size(y)) log(x1) x2];
ptemp = F\log(y)
p0 = [exp(ptemp(1)) ptemp(2) ptemp(3)]
%
%Define the function (note the syntax)
yfunction=@(p,x) p(1)*x(:,1).^p(2).*exp(p(3)*x(:,2));
[p ssr] = lsqcurvefit(yfunction,p0,[x1 x2],y)
q = length(y)-length(p);
sd = sqrt(ssr/q)
```

The output is almost identical in the two cases, and we give it only for the *fminsearch.m* implementation:

```
>>nlreg1
ptemp =
1.0080
0.8401
0.0399
p0 =
2.7402 0.8401 0.0399
p =
```

```
2.8095 0.8241 0.0393
ssr =
230.5329
sd =
0.7255
```

3.3 Systems Nonlinear in their Parameters

In practice, we are often interested in using these techniques to analyze systems of ODE or PDE, but before we embarck on such cases, we consider a contrived example for which we won't require the additional step of solving the relevant ODE or PDE. Suppose we have the system relation

$$\begin{aligned} y_1 &= p_1 e^{-p_2 x_1} + p_3 e^{p_4 x_2} \\ y_2 &= p_5 e^{-p_2 x_1} + p_6 e^{p_4 x_2}, \end{aligned}$$

along with data $\{(\vec{x}_k, \vec{y}_k)\}_{k=1}^N$. For an ODE or PDE, we would typically use linearity of the equation to obtain approximate parameter values, but for this contrived case we will suppose such an approximation is known,

$$\vec{p}_0 = (.01, .1, .01, .1, 1, 1).$$

The built-in MATLAB M-file *lsqcurvefit.m* cannot accommodate systems, so in this case we will proceed only with *fminsearch.m*. We carry this out with *nlreg2g.m* for randomly generated data stored in *systemregressiondata.mat*.

```
%NLREG2g: Matlab script M-file for analyzing the
%data stored in systemregressiondata
load systemregressiondata
%Define the error
f1 = @(p) p(1)*exp(-p(2)*x1) + p(3)*exp(p(4)*x2);
f2 = @(p) p(5)*exp(-p(2)*x1) + p(6)*exp(p(4)*x2);
w1 = std(y1);
w2 = std(y2);
E = @(p) norm(y1-f1(p))^2/w1^2 + norm(y2-f2(p))^2/w2^2;
%Minimize the error with fminsearch
%For initial guess, get orders correct
p0 = [.01 .1 .01 .1 1 1];
[p ssr]=fminsearch(E,p0) %Using p instead of pstar
q = 2*length(y1)-length(p0);
s = sqrt(ssr/q)
%
s1 = w1*s
s2 = w2*s
```

3.4 Fitting Data to an ODE System

In this subsection, we assume we have data $\{(t_k, \vec{y}_k)\}_{k=1}^N$ that we would like to fit to the vector solution $\vec{y}(t;\vec{p})$ of a system or ordinary differential equations

$$\frac{d\vec{y}}{dt} = \vec{f}(\vec{y},\vec{p}), \quad \vec{y}(t_0) = \vec{y}_0.$$

As usual, our goal is to find a parameter vector $\vec{p}$ that minimizes the SSR

$$E(\vec{p}) = \sum_{k=1}^{N} |\vec{y}_k - \vec{y}(t_k;\vec{p})|^2,$$

or a weighted SSR

$$E(\vec{p}) = \sum_{k=1}^{N}\sum_{i=1}^{n} (y_{ki} - y_i(t_k;\vec{p}))^2/w_i^2,$$

but we now have the additional complication that most nonlinear ODE that arise in practical applications cannot be solved exactly. To handle such cases, we will need to numerically generate solutions to such equations as part of the regression calculations. We'll see how to do this in the next example.

Example 3.4. In the *News and Notes* section of the March 4, 1978 issue of the *British Medical Journal*, there is a brief article on p. 587 describing the progression of a strain of flu through a boys' boarding school in the north of England. As is all too common with articles along these lines, the data is never explicitly given, and is only available in the form of a plot provided in the article. By the method of magnifying and squinting, I've estimated the data to be approximately as given in Table 3.1.

Day	0	3	4	5	6	7	8	9	10	11	12	13	14
Susceptible	762	740	650	400	250	120	80	50	20	18	15	13	10
Infected	1	20	80	220	300	260	240	190	120	80	20	5	2

Table 3.1: Approximate data from the British Medical Journal, Mar. 4, 1978.

In this example, we will fit the data in Table 3.1 to the SIR epidemic model, which gets its name from dividing a population into the following three categories:

$$\begin{aligned} S(t) &= \text{number of susceptible individuals at time } t \\ I(t) &= \text{number of infected/infective individuals at time } t \\ R(t) &= \text{number of recovered/removed individuals at time } t. \end{aligned}$$

The model consists of the following system of ODEs:

$$\begin{aligned} \frac{dS}{dt} &= -aSI \\ \frac{dI}{dt} &= aSI - bI \\ \frac{dR}{dt} &= bI. \end{aligned}$$

The rationale for such a model is straightforward. We begin by assuming that the number of individuals infected per unit time by a single infected individual is proportional to the number of susceptible individuals in the population. I.e., there exists a proportionality constant a so that this number of infections is aS. If this is the number of individuals infected by each infective individual, then the total number of individuals infected per unit time by I infective individuals is aSI. This change is then precisely the rate at which individuals are leaving the susceptible population, giving the first equation $\frac{dS}{dt} = -aSI$. Next, each of these individuals leaving the susceptible population transitions to the infected/infective population, leading to the first summand on the right-hand side of the equation for $\frac{dI}{dt}$. In addition, it is assumed that the rate of recovery/removal is proportional to the number of infected/infective individuals in the population, giving the second summand on the right-hand side of the equation for $\frac{dI}{dt}$. Finally, each individual leaving the infected/infective population transitions to the recovered/removed population. For the SIR model, it is assumed that no individual in the recovered/removed population returns to the susceptible population.

We will proceed with nonlinear regression, in which case we first need to use linear regression to obtain initial approximations for the parameter values a and b. The easiest way to do this is with the second equation, which we can express as

$$\frac{1}{I}\frac{dI}{dt} = aS - b.$$

We can now fit $Y = \frac{1}{I}\frac{dI}{dt}$ as a function of S to obtain a as the slope and $-b$ as the intercept. We keep in mind here that we could proceed similarly with the first equation to estimate a and with the third equation to estimate b. This would give a different pair of values, but that pair would also be sufficient as initial approximations for the nonlinear regression. In this case, the time steps are not uniform, which suggests we might want to use the forward difference derivative approximation, but it turns out that the central difference derivative approximation works substantially better, so that's the one we'll use. The linear fit is carried out with *sirlinear1.m*. The linear fit is depicted in Figure 3.7, and the fit against data with the linearly obtained parameter values is depicted in Figure 3.8. The parameter values are computed to be

$$a = .0036$$
$$b = .9395.$$

```
%SIRLINEAR1: MATLAB script M-file in which SIR parameter
%values a and b are approximated by a linear fit based
%on the central difference derivative approximation and
%the infected/infective equation from the first-order
%SIR system.
%
%DATA FROM THE BRITISH MEDICAL JOURNAL, MAR. 4, 1978.
S=[762 740 650 400 250 120 80 50 20 18 15 13 10];
I=[1 20 80 220 300 260 240 190 120 80 20 5 2];
days=[0 3 4 5 6 7 8 9 10 11 12 13 14];
```

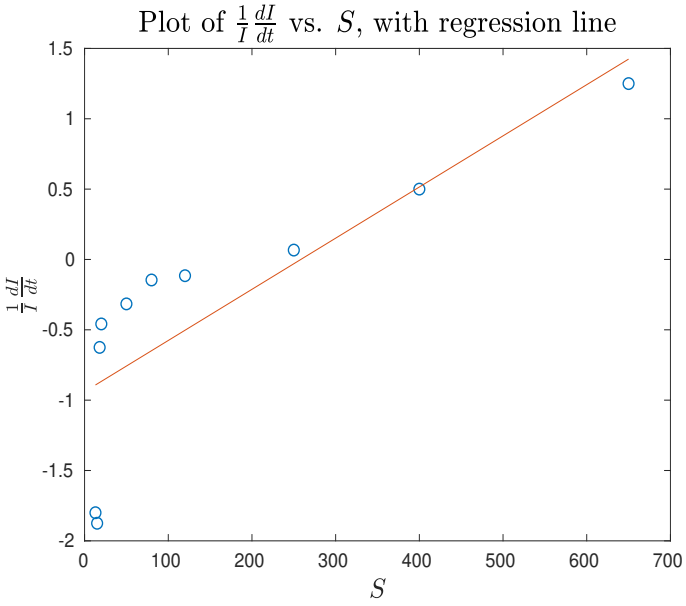


Figure 3.7: Linear fit for SIR example.

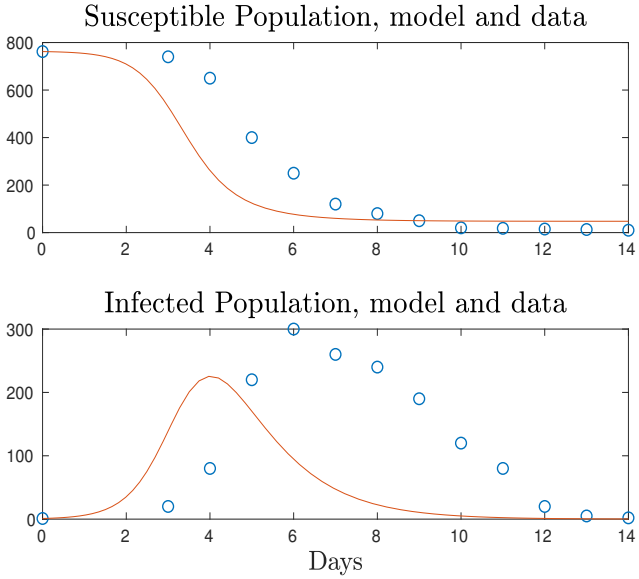


Figure 3.8: Fit of SIR model with data for linearly obtained parameters.

```
%CENTRAL DIFFERENCE FIT
dIdt = (I(4:end)-I(2:end-2))./2;
Y = dIdt./I(3:end-1);
X = S(3:end-1);
plot(X,Y,'o');
title('Plot of $\frac{1}{I} \frac{dI}{dt}$ vs. $S$','Interpreter','latex','FontSize',16)
xlabel('$S$','Interpreter','latex','FontSize',14)
ylabel('$\frac{1}{I} \frac{dI}{dt}$','Interpreter','latex','FontSize',14)
pause
c = polyfit(X,Y,1);
a = c(1)
b = -c(2)
plot(X,Y,'o',X,c(1)*X+c(2))
title('Plot of $\frac{1}{I} \frac{dI}{dt}$ vs. $S$, with regression line', ...
'Interpreter','latex','FontSize',16)
xlabel('$S$','Interpreter','latex','FontSize',14)
ylabel('$\frac{1}{I} \frac{dI}{dt}$','Interpreter','latex','FontSize',14)
%
pause
hold off
sirrhs = @(t,y) [-a*y(1)*y(2);a*y(1)*y(2)-b*y(2)];
[t,y]=ode45(sirrhs,[0 14],[762;1]);
subplot(2,1,1)
plot(days,S,'o',t,y(:,1))
title('Susceptible Population, model and data','Interpreter','latex','FontSize',16)
subplot(2,1,2)
plot(days,I,'o',t,y(:,2))
title('Infected Population, model and data','Interpreter','latex','FontSize',16)
xlabel('Days','Interpreter','latex','FontSize',14)
%
%To compute the standard deviation
[t,y]=ode45(sirrhs,days,[762;1]);
lE = norm(S'-y(:,1))^2+norm(I'-y(:,2))^2;
q = 2*(length(S)-1)-2;
ls = sqrt(lE/q)
```

Using the linearly-obtained values as initial approximations, we now carry out the nonlinear fit with *sirnonlinear1.m*. We obtain the parameter values

$$a = .0022$$
$$b = .4408,$$

and a fit of the SIR model along with data is depicted in Figure 3.9.

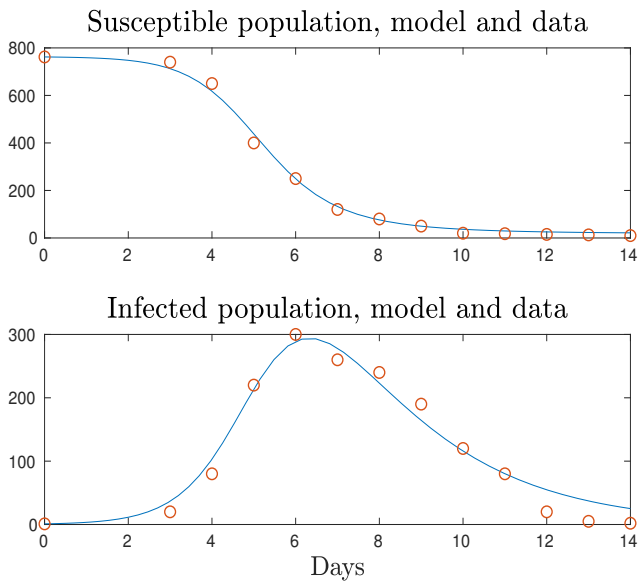


Figure 3.9: Fit of SIR model with data for nonlinearly obtained parameters.

```
function sirnonlinear1
%SIRNONLINEAR1: MATLAB function M-file that takes an initial
%approximation of parameter values and carries out nonlinear
%regression to obtain best-fit parameter values for the SIR
%system and the British Medical Journal influenza data.
global days S I w1 w2;
S=[762 740 650 400 250 120 80 50 20 18 15 13 10];
I=[1 20 80 220 300 260 240 190 120 80 20 5 2];
days=[0 3 4 5 6 7 8 9 10 11 12 13 14];
%Weights
w1 = std(S(2:end));
w2 = std(I(2:end));
guess = [.0036 .9395];
[p,error]=fminsearch(@sirerr, guess);
a = p(1)
b = p(2)
q = 2*(length(S)-1)-length(p);
s = sqrt(error/q)
s1 = w1*s
s2 = w2*s
%
[t,y]=ode45(@sirpe,[0,14],[S(1); I(1)],[],p);
subplot(2,1,1)
plot(t,y(:,1),days,S,'o')
title('Susceptible population, model and data','Interpreter','latex','FontSize',16)
subplot(2,1,2)
```

```
plot(t,y(:,2),days,I,'o')
title('Infected population, model and data','Interpreter','latex','FontSize',16)
xlabel('Days','Interpreter','latex','FontSize',14)
%
function error = sirerr(p)
%LVERR: Function defining error function for
%example with SIR equations.
global days S I w1 w2;
[t,y] = ode45(@sirpe,days,[S(1);I(1)],[],p); %Notice that we pass
%a parameter vector
error = norm(y(:,1)-S')^2/w1^2+norm(y(:,2)-I')^2/w2^2;
%
function value = sirpe(t,y,p)
%LVPE: ODE for example SIR paramter
%estimation example. p(1)=a, p(2) = b.
value=[-p(1)*y(1)*y(2);p(1)*y(1)*y(2)-p(2)*y(2)];
```

3.5 Neural Networks and Deep Learning

At the most basic level, deep learning with neural networks is an application of nonlinear regression, so we conclude our discussion of nonlinear regression with a brief introduction to this important topic. As background, starting in 1943, the American neurophysiologist Warren McCulloch (1898-1969) and American logician Walter Pitts (1923-1969) worked to create a mathematical model for computer implementation based on neural networks in the human brain. This work led to the development we'll discuss.

Our starting point will be the mathematical notion of a *neuron*, and in particular a type of neuron called a *perceptron*, named by the American psychologist Frank Rosenblatt (1928-1971) in the 1950s. The idea is straightforward: we have some number of binary (e.g., 0/1) inputs x_1, x_2, ..., x_n, and we would like to produce a single binary output (a neuron either fires or it doesn't). Schematically, with three inputs, we can view this as in Figure 3.10.

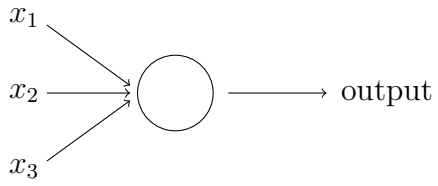


Figure 3.10: Schematic of a single neuron.

The inputs for a neuron may not be equally important for determining its output, so we generally associate a weight w_i with each input x_i. We can then think of the neuron's output as being determined by the weighted sum

$$\vec{w} \cdot \vec{x} = \sum_{i=1}^{n} w_i x_i$$

(i.e., the usual dot product). One way to do this is to set some threshold value v, and if $\vec{w} \cdot \vec{x} \leq v$, we set the output to 0, while if $\vec{w} \cdot \vec{x} > v$ we set the output to 1. We can express this as

$$\begin{aligned} \text{output} &= \begin{cases} 0 & \text{if } \vec{w} \cdot \vec{x} \leq v \\ 1 & \text{if } \vec{w} \cdot \vec{x} > v \end{cases} \\ &= \begin{cases} 0 & \text{if } \vec{w} \cdot \vec{x} - v \leq 0 \\ 1 & \text{if } \vec{w} \cdot \vec{x} - v > 0. \end{cases} \end{aligned}$$

In lieu of v, it's traditional in the context of neural networks to introduce a *bias* $b = -v$, so that

$$\text{output} = \begin{cases} 0 & \text{if } \vec{w} \cdot \vec{x} + b \leq 0 \\ 1 & \text{if } \vec{w} \cdot \vec{x} + b > 0. \end{cases}$$

If we set $z = \vec{w} \cdot \vec{x} + b$, then graphically the output is described by the Heaviside function[19]

$$H(z) = \begin{cases} 0 & z \leq 0 \\ 1 & z > 0, \end{cases}$$

as depicted in Figure 3.11.

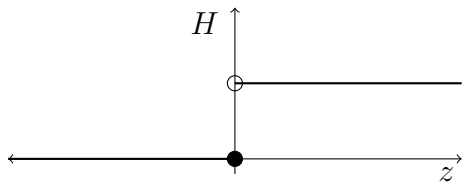


Figure 3.11: Heaviside function.

In fact, nothing about this discussion requires binary inputs, so we can just as well take x_1, x_2, ..., x_n to be any values we like. In practice, we often take each x_i to be some value on the interval $[0, 1]$. Likewise, we can also work with continuous output, and one way to do this is to approximate the Heaviside function $H(z)$ with the *sigmoid function*

$$\sigma(z) = \frac{1}{1 + e^{-z}},$$

depicted in Figure 3.12. In this case, the neuron's output is taken to be $\sigma(\vec{w} \cdot \vec{x} + b)$. We might make a decision by rounding this value up or down, or we might want continuous output. Whether we choose this function to be $H(z)$, $\sigma(z)$, or something else, we refer to it as the *activation function*. Another commonly used activation function is the "rectified linear unit" (ReLU)

$$r(z) = \begin{cases} 0 & z \leq 0 \\ z & z > 0. \end{cases}$$

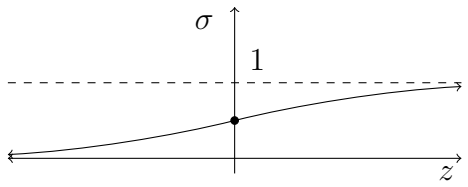


Figure 3.12: Sigmoid function.

A neural network consists of a combination of interconnected neurons. For example, we might have the neural network depicted in Figure 3.13. Each circle is a neuron or *node*, and each column of nodes is referred to as a *layer*. Every neural network has an input layer and an output layer, and any layers between these are called "hidden" layers. It's clear that the neural network depicted in Figure 3.13 has two hidden layers. By custom, we refer to a neural network with two or more hidden layers as a *deep neural network*.

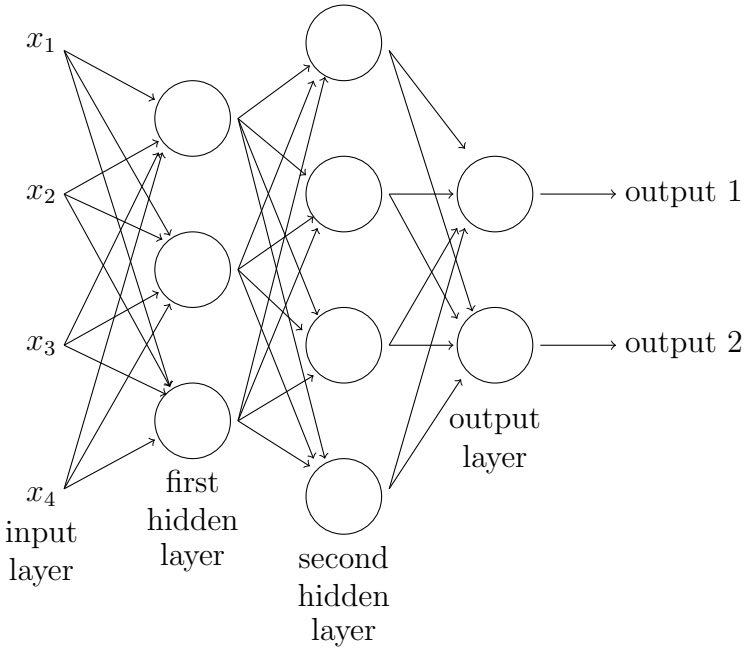


Figure 3.13: Neural network with two hidden layers.

As we will see in a specific example at the end of this section, neural networks can be quite large in practice, so it's important to have succinct notation for describing them. As a starting point, we will say the input layer is layer 0, and the subsequent layers (left to right) will be labeled layers 1, 2, 3 etc, up to the final layer L. (For the neural network in Figure 3.13, $L = 3$.) Within each layer, we will number the nodes top to bottom. We need to associate a weight vector and a scalar bias to each node, so we will use both subscripts and

[19]The Heaviside function is named for the English mathematician and physicist Oliver Heaviside (1850-1925).

superscripts, with the superscript denoting the layer and the subscript denoting the node. For example, for the second node from the top in the first hidden layer, we will denote the (4-component) weight vector $\vec{w}_2^1$ and the scalar bias b_2^1. Likewise, we will denote the output associated with this node

$$\alpha_2^1 = \sigma(\vec{w}_2^1 \cdot \vec{x} + b_2^1).$$

Here, we're using a superscript to leave room for yet another index, which will be a second subscript. In particular, we will view $\vec{w}_2^1$ as a row vector, and express it as

$$\vec{w}_2^1 = (w_{21}^1, w_{22}^1, w_{23}^1, w_{24}^1).$$

The first hidden layer in the neural network in Figure 3.13 has three nodes, so there will correspondingly be three weight vectors, $\vec{w}_1^1$, $\vec{w}_2^1$, and $\vec{w}_3^1$. We combine these into a weight matrix for level 1,

$$W^1 = \begin{pmatrix} \vec{w}_1^1 \\ \vec{w}_2^1 \\ \vec{w}_3^1 \end{pmatrix} = \begin{pmatrix} w_{11}^1 & w_{12}^1 & w_{13}^1 & w_{14}^1 \\ w_{21}^1 & w_{22}^1 & w_{23}^1 & w_{24}^1 \\ w_{31}^1 & w_{32}^1 & w_{33}^1 & w_{34}^1 \end{pmatrix}.$$

Similarly, we specify a bias vector

$$\vec{b}^1 = \begin{pmatrix} b_1^1 \\ b_2^1 \\ b_3^1 \end{pmatrix}.$$

If we now compute the 3-vector

$$W^1\vec{x} + \vec{b}^1,$$

we see that its first component is the quantity $\vec{w}_1^1 \cdot \vec{x} + b_1^1$ required for the top node in level 1, its second component is the quantity $\vec{w}_2^1 \cdot \vec{x} + b_2^1$ required for the middle node in level 1, and its third component is the quantity $\vec{w}_3^1 \cdot \vec{x} + b_3^1$ required for the bottom node in level 1. The output from the top node in level 1 is

$$\alpha_1^1 = \sigma(\vec{w}_1^1 \cdot \vec{x} + b_1^1),$$

the output from the middle node in level 1 is

$$\alpha_2^1 = \sigma(\vec{w}_2^1 \cdot \vec{x} + b_2^1),$$

and the output from the bottom node in level 1 is

$$\alpha_3^1 = \sigma(\vec{w}_3^1 \cdot \vec{x} + b_3^1).$$

We typically combine these expressions into

$$\vec{\alpha}^1 = \sigma(W^1\vec{x} + \vec{b}^1),$$

where we view σ as acting on one component at a time, returning the vector

$$\vec{\alpha}^1 = \begin{pmatrix} \alpha_1^1 \\ \alpha_2^1 \\ \alpha_3^1 \end{pmatrix}.$$

Moving to the next level, the three components of $\vec{\alpha}^1$ become the three outputs from level 1, playing now the roles of the components of $\vec{x}$ as the initial inputs. Level 2 has four nodes, and since there are three inputs, each node will have a corresponding 3-vector weight and scalar bias. Respectively, the corresponding weight matrix and bias vector are

$$W^2 = \begin{pmatrix} w_{11}^2 & w_{12}^2 & w_{13}^2 \\ w_{21}^2 & w_{22}^2 & w_{23}^2 \\ w_{31}^2 & w_{32}^2 & w_{33}^2 \\ w_{41}^2 & w_{42}^2 & w_{43}^2 \end{pmatrix} \quad \text{and} \quad \vec{b} = \begin{pmatrix} b_1^2 \\ b_2^2 \\ b_3^2 \\ b_4^2 \end{pmatrix}.$$

The output from level 2 can now be computed as

$$\vec{\alpha}^2 = \sigma(W^2\vec{\alpha}^1 + \vec{b}^2).$$

Notice particularly that this last expression is the same no matter how many nodes there are in levels 1 and 2 (that information is all contained in the objects W^2, $\vec{b}^2$, and $\vec{\alpha}^1$.)

Last, we arrive at the output layer $L = 3$, for which wave have

$$W^3 = \begin{pmatrix} w_{11}^3 & w_{12}^3 & w_{13}^3 & w_{14}^3 \\ w_{11}^3 & w_{12}^3 & w_{13}^3 & w_{14}^3 \end{pmatrix} \quad \text{and} \quad \vec{b}^3 = \begin{pmatrix} b_1^3 \\ b_2^3 \end{pmatrix}.$$

The final output of the neural network is seen to be

$$\text{output} = \vec{\alpha}^3 = \sigma(W^3\vec{\alpha}^2 + \vec{b}^3).$$

Combining these observations, we see that the full calculation can be viewed as an iterative process: for a neural network with L layers ($L + 1$ if the input layer 0 is counted), we have

$$\begin{aligned} \vec{\alpha}^\ell &= \sigma(W^\ell\vec{\alpha}^{\ell-1} + \vec{b}^\ell), \quad \ell = 1, 2, \ldots, L, \\ \vec{\alpha}^0 &= \vec{x}. \end{aligned} \tag{3.3}$$

The final output of the neural network is the vector $\vec{\alpha}^L$. We emphasize that with the notation that we've developed, the recursion system (3.3) describes a general neural network with any number of layers and nodes.

Certainly, the final vector $\vec{\alpha}^L$ depends on the initial input $\vec{x}$, and we observe that it also depends on all of the weights and biases, which even for this simple example are numerous: 12 weights and 3 biases for level 1, 12 weights and 4 biases for level 2, and 8 weights and 2 biases for level 3, for a grand total of 41 parameters. This differs markedly from our previous cases arising in this section on regression, in which the goal was generally to introduce the least number of parameters impossible. Indeed, one of the hallmarks of deep learning is the appearance (and, more important, the need for) a huge number of parameters.

Slightly abusing notation, we can express the final output of a general neural network with L levels as $\vec{\alpha}^L(\vec{x}; \vec{p})$, where as usual $\vec{x}$ denotes the input vector, and where $\vec{p}$ denotes a vector comprising all weights and biases in the network. E.g., for the neural network in Figure 3.13, $\vec{p}$ is a vector with 41 components.[20] We can now think of expressing the relation associated with our neural network in the standard form

$$\vec{y} = \vec{\alpha}^L(\vec{x}; \vec{p}),$$

[20] Ordered in any way we choose, as long as we do it consistently.

which is precisely the sort of relation we know how to handle with nonlinear least squares regression. Precisely, in order to "train" our neural network on a set of data $\{(\vec{x}_k, \vec{y}_k)\}_{k=1}^N$, we minimize the usual SSR

$$E(\vec{p}) = \sum_{k=1}^{N} |\vec{y}_k - \vec{\alpha}^L(\vec{x}_k; \vec{p})|^2,$$

or often, since these values get so large,

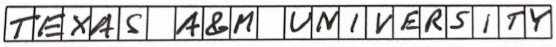
$$E(\vec{p}) = \frac{1}{N} \sum_{k=1}^{N} |\vec{y}_k - \vec{\alpha}^L(\vec{x}_k; \vec{p})|^2.$$

In many cases, this turns out to be an extremely effective way to handle applied problems, so much so, in fact, that one of the main questions researchers in the area of neural networks are trying to address is, Why is the method so effective? Among the numerous other aspects of deep learning under intense investigation are the following:

- As is always the case with nonlinear regression, it's important to have good initial approximations for the best-fit parameter values. How can these be obtained in the setting of neural networks?
- Given a specific application, what "architecture" should one choose for the neural network describing it; i.e., how many layers should the network have, and how many nodes should be in each layer. Also, what activation function should be chosen?
- With large numbers of parameters and enormous data sets, minimizing the SSR for a neural network can be computationally intensive. Several techniques have been introduced in order to make this process more efficient, and research into additional possibilities is ongoing.

3.5.1 Application to Image Recognition

One standard application of neural networks is to the science of image recognition, and in order to illustrate how neural networks can be implemented in practice, we will discuss a simple such example. To keep things tractable, we will suppose our images are the sorts of block letters that might be used in filling in a form by hand:



Our goal is to determine how we could apply the previous considerations to develop a neural network that will correctly identify the letter in a single box. Since these boxes are small, we can use a relatively small number of pixels for each image; in particular, let's suppose our images are 28×28 and grayscale. I.e., each letter is broken up into a square grid of $28 \times 28 = 784$ pixels, and each pixel corresponds with a numerical value on the interval $[0, 1]$, with 0 corresponding with white and 1 corresponding with black. This means that our inputs will be vectors of length 784.

The most direct way to contruct the output is as a vector with 26 components, each corresponding with a letter in the alphabet. If the output associated with a letter is 0, then that is not the letter recognized, while if the output associated with a letter is 1, then that is the letter recognized, so in practice we should always have 25 outputs near 0 and 1 output near 1. (The output for the training data will always comprise vectors with precisely 25 zeros and one 1.) At this point, we have decided to use 784 inputs and 26 outputs, but we still need to determine the full architecture of our network. There is no standard way of choosing either the number of hidden layers to use or the number of nodes in each layer, and in practice there is often a trial-and-error approach to this process. For our example, let's suppose we use a single hidden layer with 40 nodes, which would at least be a reasonable thing to try. This would get us to $40 \times 784 = 31,360$ weights in level 1, with also 40 biases, and $26 \times 40 = 1,040$ weights in the output layer, with also 26 biases. In total, we have $32,466$ parameters. In principle, the neural network can now be trained and implemented precisely as described above, though the practical implementation requires considerable care. We won't pursue it further in these notes.

4 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 M442 course notes *Modeling with ODE* and (separately) *Analysis of ODE Models*).

Example 4.1. 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 can be expressed in terms of velocity $v = y'$ as

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

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

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

△

5 Dimensional Analysis

Dimensional analysis refers to a collection of methods for understanding physical phenomena and relations based entirely on the dimensions of the quantities involved. As a start, we recall that a *dimension* is any quantity such as length, mass or time that can be measured. It's constructive to contrast this with a unit, which is a gauge we might use to measure a dimension. For example, the dimension length can be measured by several different units, including inches, feet, meters, light years etc. We refer to length as a *fundamental* dimension, and similarly for mass and time. Quantities such as velocity, acceleration, force etc. are considered to be *derived* dimensions. According to the International System of Units (SI),[21] there are seven fundamental dimensions, which we list in Table 5.1 along with their associated base units. (When discussing population dynamics, we will often refer to population as *biomass* and treat this is an eighth fundamental dimension, B.) Some typical physical quantities and their associated dimensions are listed in Table 5.2.

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

Table 5.1: Fundamental dimensions and their base units.

Quantity	Dimensions	Quantity	Dimensions
Length	L	Frequency	T^{-1}
Time	T	Density	ML^{-3}
Mass	M	Angular momentum	ML^2T^{-1}
Velocity	LT^{-1}	Viscosity	$ML^{-1}T^{-1}$
Acceleration	LT^{-2}	Pressure	$ML^{-1}T^{-2}$
Force	MLT^{-2}	Power	ML^2T^{-3}
Energy	ML^2T^{-2}	Entropy	$ML^2T^{-2}\Theta^{-1}$
Momentum	MLT^{-1}	Heat energy	ML^2T^{-2}
Work	ML^2T^{-2}	Voltage	$ML^2T^{-3}I^{-1}$

Table 5.2: Dimensions of common physical quantities.

Although the SI choices for the fundamental dimensions are sensible, they are by no means inevitable. For example, while velocity is viewed as a derived dimension, electric current (which, as charge per unit time, is analogous to velocity) is taken to be a fundamental

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

dimension, with charge then being considered a derived dimension. The method of dimensional analysis does not depend on these choices as long as: (1) some choice of fundamental dimensions is used consistently, and (2) the fundamental dimensions include all dimensions arising in the problem under consideration. It's also worth observing that considerable effort has gone into choosing standards for the units. Early units for length were based on individual physical attributes; for example, a *cubit* was the length of an individual's arm from the elbow to the tip of the 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. These notions of measure began to be standardized in about 1100 AD by Henry I, who decreed that a yard would be the distance between the tip of his nose and the end of his outstretched thumb. Later, a meter was taken to be the distance between two indicated positions on a platinum-iridium bar kept in a laborator in France, and is now[22] defined so that the speed of light in a vacuum is precisely 299,792,458 m/s.[23]

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. Under fixed atmospheric conditions, 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. The clear drawback of this approach is that the resulting unit depends on the choice of atmospheric conditions, and so isn't universal. For many years (1967–2019), the SI convention for temperature hinged on the observation that liquid water, solid ice, and water vapor can coexist at only one set of values for temperature and pressure, which could then be viewed as a canonical choice of atmospheric conditions. By international agreement in 1967, the triple point of water was taken to correspond with a temperature of 273.16 K (with corresponding pressure 611.2 Pascals). More succinctly, 1 Kelvin was precisely 1/273.16 of the (unique) temperature for the triple point of water. In 2019, this definition changed again, and currently a Kelvin is defined so that a value known as the Boltzmann constant[24] is precisely 1.380649×10^{-23} J/K.

When thinking about physical processes, it's often instructive simply to make sure we understand the dimensions involved. For example, we see from Table 5.2 that energy E and work W have the same dimensions, and this might remind us that the Work-Energy Principle asserts that the work done on a particle equals the change in kinetic energy of the

[22]Skipping a brief phase (1960-1983) in which it was defined in terms of a transition of energy levels in the Krypton-86 atom.

[23]This definition requires that the second be previously defined, and the second is defined based on a certain transition frequency associated with the cesium-133 atom.

[24]For an ideal gas, the product of pressure p and volume V is proportional to the product of temperature T and the number of molecules in the gas N (i.e., $pV \propto NT$), and the Boltzmann constant is precisely the proportionality constant. I.e., if we denote the Boltzmann constant by κ, then $pV = \kappa NT$. The constant is named for the Austrian physicist Ludwig Boltzmann (1844-1906).

particle,

$$W = \Delta K.$$

5.1 Finding Simple Relations

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

Example 5.1. Suppose an object is fired directly up from the earth's surface 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 will land.

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 can summarize this by writing $t = t(v, g)$. Next, we assume the relationship between t, v, and g has the simple form

$$t = kv^a g^b, \tag{5.1}$$

where k is a dimensionless constant and values for a and b will be found just below. Later, we'll see what happens if such a relationship doesn't in fact hold, but for now let's assume that it does, which turns out to be true for this example. At this point, we assume that for physical processes the dimensions on each side of an equation must agree (we say the equation is "dimensionally consistent"). For this to be true, we must have the following relationship, obtained by equating the dimensions on each side of (5.1):

$$T = L^a T^{-a} L^b T^{-2b}.$$

In order for this equation to hold, the exponents of the individual dimensions must match, leading 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 $t = k\frac{v}{g}$, where it's important to note that we have not found an exact form for t, only proportionality. *This is as far as dimensional analysis will take us.* At this point, we should check our expression to ensure it makes sense physically. According to our expression, the larger v is, the longer it will be before the object lands, 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 find that $k = 2$), we generally obtain values for the undetermined constants that arise in dimensional analysis by collecting data (in this case, launching the object with different velocities) and using regression $\triangle$

Example 5.2. Use dimensional analysis to 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 moon or planet, ρ, volume of the meteorite, V, 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 (for the exponents of m, ρ, V, v, and g). In order to apply the method outlined in Example 5.1, 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 from our list. Next, let's try to combine quantities. Noticing that m and v can be combined into kinetic energy ($E = \frac{1}{2}mv^2$), we can drop them and consider the new quantity of dependence E. Finally, we are prepared to begin our analysis. We have,

$$R = R(E, \rho, g) = kE^a\rho^b g^c \implies L = M^a L^{2a} T^{-2a} M^b L^{-3b} L^c T^{-2c},$$

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

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

Again, we observe that the basic dependences make sense: higher energies create larger craters, while planets with greater density or gravitational pull end up with smaller craters. $\triangle$

One useful application of this method is that it can help remind us of the form a known relation must have. The next two examples illustrate this.

Example 5.3. 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 5.1). 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,

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

from which we obtain the dimensions equations,

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

Solving, we find that $a = 1$, $c = 2$, and $b = -1$, so that

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

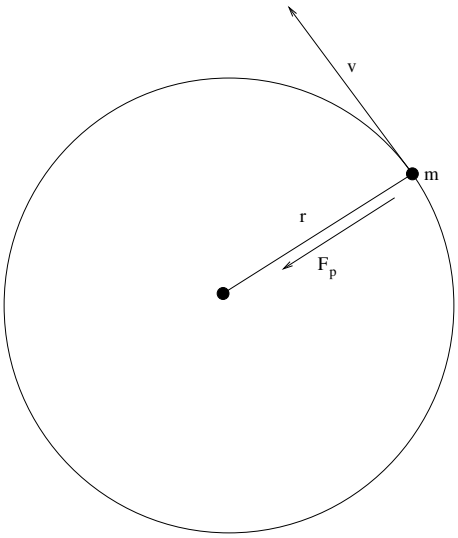


Figure 5.1: Centripetal force on a rotating object.

This calculation may serve to remind us that we can in fact use Newtonian mechanics to show that $F_p = \frac{mv^2}{r}$ (i.e., in this case, $k = 1$). △

Example 5.4. 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.[25] 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

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

In this case, we leave off the usual proportionality constant k, which we can view as being absorbed into G. 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:

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

[25]Following standard notation, we are denoting dimension by square brackets $[\cdot]$, so if t is a time, we will write $[t] = T$, and likewise if v is a velocity we will write $[v] = LT^{-1}$ and so on.

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

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

△

5.2 More General Dimensional Analysis

Example 5.1a. Let's consider the following slight variation to Example 5.1 (we'll refer to it as Example 5.1a): suppose that instead of launching the object from the earth's surface, we launch it from a height h above the earth's surface. If we still want to determine the time t at which the object will return to the earth's surface, then we must assume that this time depends on three quantities, v, g, and h. Proceeding similarly as before, we now have

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

from which we obtain the dimensions equations,

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

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 difficulty in Example 5.2 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. According to Newton's second law of motion, the height of our object at time t is given by

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

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

$$t = \frac{-v \pm \sqrt{v^2 + 2gh}}{-g}. \tag{5.2}$$

Of course the time we're looking for is positive[26], so

$$t = \frac{v + \sqrt{v^2 + 2gh}}{g},$$

and we see that this relation does not have the simple form $t = kv^a g^b h^c$. In light of this, we see that our initial approach can't possibly work for this example, and so we will need to proceed in a more general way. To this end, we will introduce the notion of *dimensionless products*.

[26]The corresponding negative time is precisely the time at which the object would have to be fired from the ground to achieve the height h and velocity v at time $t = 0$.

5.2.1 Dimensionless Products

Precisely as the name suggests, a dimensionless product is a multiplicative combination of variables that has no dimension. For example, recall from our Example 5.1a the equation

$$-\frac{1}{2}gt^2 + vt + h = 0.$$

If we divide this equation by h, we obtain

$$-\frac{1}{2}\frac{gt^2}{h} + \frac{vt}{h} + 1 = 0, \tag{5.3}$$

and each summand in this new equation must be dimensionless. In particular, the multiplicative combinations $\frac{gt^2}{h}$ and $\frac{vt}{h}$ are dimensionless, and so serve as our first examples of dimensionless products.

One advantage of dimensionless products is that if we change our units of measurement, their values don't change. For example, suppose we evaluate the dimensionless product $\frac{vt}{h}$ with the units feet for length and seconds for time, and likewise we evaluate the same dimensionless product with the units inches for length and milliseconds for time. If we let v_1, t_1, and h_1 denote the measurements in the first units, and likewise we let v_2, t_2, and h_2 denote the measurements in the second units, then we have

$$\begin{aligned} t_2 &= 1000t_1 \\ h_2 &= 12h_1 \\ v_2 &= \frac{12}{1000}v_1, \end{aligned}$$

so that

$$\frac{v_2t_2}{h_2} = \frac{\frac{12}{1000}v_1 \cdot 1000t_1}{12h_1} = \frac{v_1t_1}{h_1}.$$

In most cases, we don't have an equation such as (5.3) to work with, so we need a more general approach for identifying dimensionless products. First, we typically denote a dimensionless product by π (for product).[27] For Example 5.1a, the variables we're working with are v, g, h, and t, and so we look for multiplicative combinations of these variables

$$\pi = \pi(v, g, h, t) = v^a g^b h^c t^d. \tag{5.4}$$

In this case, we don't include a proportionality constant, because for the moment we're only interested in how the variables combine. If we equate the dimensions on either side of (5.4), we obtain the relation

$$1 = L^aT^{-a}L^bT^{-2b}L^cT^d,$$

where the 1 on the left-hand side records that π is dimensionless. From this last relation, we obtain the dimensions equations

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

[27] Dimensionless products also typically have subscripts, so there isn't much danger of confusing them with the standard constant π.

In the usual way, we can express this as a matrix system

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ -1 & -2 & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \tag{5.5}$$

where for convenient reference we will designate the matrix on the left-hand side by A. Importantly, we see that the exponents we're trying to identify will correspond precisely with vectors in the null space (a.k.a., the kernel) of A: i.e., the vectors $\vec{v}$ so that $A\vec{v} = 0$. For such calculations, it's convenient to put A in row reduced echelon form, which can be accomplished by hand (at least for relatively small matrices), and more generally with MATLAB's built-in *rref* command. The RRE form of our system is

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

where (again for convenient reference) we will designate the matrix on the left-hand side by $\tilde{A}$.

At this point, let's recall that the *rank* of any matrix A is the dimension of its range, which is equal to the number of linearly independent columns it has, and is also equal to the number of linearly independent rows it has. In particular, for a matrix in RRE form, the rank is just the number of non-zero rows that remain. Here, the rank of $\tilde{A}$ is clearly 2, and rank does not change under row operations, so the rank of A must be 2 as well.[28]

In identifying dimensionless products, we're more interested in the the *nullity* of a matrix (i.e., the dimension of its null space), and this is related to the rank via the Rank-Nullity Theorem.[29]

Theorem 5.1 (Rank-Nullity Theorem). *For any matrix $A \in \mathbb{C}^{m\times n}$,*

$$\operatorname{rank} A + \operatorname{nullity} A = n.$$

For Example 5.1a, we have $n = 4$ and $\operatorname{rank} A = 2$, so $\operatorname{nullity} A = 2$. This means:

1. We will be able to find two linearly independent solutions of the system $A\vec{v} = 0$, and these solutions will comprise a basis for the null space of A;

2. The linearly independent vectors described in Item 1 will correspond with a *complete set of dimensionless products* for this example in the following sense: any additional dimensionless products can be expressed as a multiplicative combination of these two.

[28] See Section 5.3 for rigorous statements and proofs of the results from linear algebra being used here.
[29] A proof of the Rank-Nullity Theorem is given in Section 5.3.

We can express the system $\tilde{A}\vec{v} = 0$ as

$$\begin{aligned} a &= -2c - d \\ b &= c + d, \end{aligned} \tag{5.6}$$

in which we can think of c and d as values to be chosen and a and b as values that will subsequently be determined.

Choosing π_1. For these notes, our convention will be to exclude the dependent variable from the first dimensionless product π_1. Recalling (5.4), we can accomplish this by choosing $d = 0$. If we choose $c = 0$ as well, then a, b, c, and d will all be 0, which isn't useful, so we need to take $c \neq 0$. Most simply, we can take $c = 1$. Having selected $d = 0$ and $c = 1$, we find immediately $a = -2$ and $b = 1$, giving us

$$\pi_1 = \frac{gh}{v^2}.$$

Choosing π_2. For the final (in this case second) dimensionless product, we will ensure that the dependent variable appears in a simple way. For this, we choose $d = 1$ and $c = 0$, and we compute $a = -1$ and $b = 1$, giving us

$$\pi_2 = \frac{gt}{v}.$$

Below, we will continue working with π_1 and π_2, but first let's recall the standard approach from introductory courses in linear algebra for identifying the null space of a matrix. Using the relations (5.6), we can express any element $\vec{v}$ of the null space of A in the form

$$\vec{v} = \begin{pmatrix} -2c - d \\ c + d \\ c \\ d \end{pmatrix} = c \begin{pmatrix} -2 \\ 1 \\ 1 \\ 0 \end{pmatrix} + d \begin{pmatrix} -1 \\ 1 \\ 0 \\ 1 \end{pmatrix}.$$

In this way, we readily see that the null space of A is a linear combination of the vectors

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

Recalling the connection

$$\vec{v} = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \longleftrightarrow \pi = v^a g^b h^c t^d,$$

we see that the vectors $\vec{v}_1$ and $\vec{v}_2$ correspond respectively with the dimensionless products $\pi_1 = \frac{gh}{v^2}$ and $\pi_2 = \frac{gt}{v}$.

Suppose we would like to check directly that π_1 and π_2 comprise a complete set of dimensionless products for Example 5.1a. By virtue of (5.4) any third dimensionless product π can be expressed as

$$\pi = v^{-2c-d} g^{c+d} h^c t^d = \pi_1^c \pi_2^d.$$

This corresponds precisely with the linear combination $\vec{v} = c\vec{v}_1 + d\vec{v}_2$. In summary, the fact that π_1 and π_2 forms a complete set of dimensionless products for this example is equivalent to the observation that $\vec{v}_1$ and $\vec{v}_2$ comprise a basis for the null space of A, and we have the useful correspondence

$$\vec{v} = c\vec{v}_1 + d\vec{v}_2 \longleftrightarrow \pi = \pi_1^c \pi_2^d.$$

5.2.2 Buckingham's Theorem

At this point, we can state the main theorem associated with dimensional analysis, Buckingham's Theorem.[30]

Theorem 5.2 (Buckingham's Theorem). *Suppose an algebraic equation is dimensionless (i.e., each summand is dimensionless), and additionally that $\{\pi_j\}_{j=1}^n$ comprises a complete set of dimensionless products for the variables in the equation. Then there exists a function f so that the equation can be expressed as*

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

We'll prove Buckingham's Theorem in Section 5.4, but for now let's be clear about what it's asserting. For this, we'll return to Example 5.1a, keeping in mind that we typically wouldn't use Buckingham's Theorem in cases for which we can identify f by other means. For the example, the algebraic equation referenced by Buckingham's Theorem is

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

except it's assumed that we're working with a dimensionless form such as

$$-\frac{1}{2} \frac{g t^2}{h} + \frac{v t}{h} + 1 = 0.$$

Buckingham's Theorem asserts that we can write this equation entirely in terms of π_1 and π_2, and in this case, we can do it explicitly

$$-\frac{1}{2} \frac{\pi_2^2}{\pi_1} + \frac{\pi_2}{\pi_1} + 1 = 0.$$

[30] Named for the US physicist Edgar Buckingham (1867-1940), Buckingham's Theorem was first observed by the French mathematical physicist Aimé Vaschy (1857-1899). This is yet another example of Stigler's Law of Eponymy, which states that no scientific discovery is named for its original discoverer. Stigler's Law is named for the US statistician Stephen M. Stigler (b. 1941), though is properly due to the US sociologist Robert Merten (1910-2003).

The left-hand side of this relation is an example of the function f guaranteed by Buckingham's Theorem to exist, as is (upon multiplying by π_1)

$$f(\pi_1, \pi_2) = -\frac{1}{2}\pi_2^2 + \pi_2 + \pi_1.$$

(Notice that Buckingham's Theorem doesn't claim anything about uniqueness.) Having come this far, we can now solve $f(\pi_1, \pi_2) = 0$ with the quadratic formula to obtain

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

We want the positive solution

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

and in the original variables this is

$$\frac{gt}{v} = 1 + \sqrt{1 + 2\frac{gh}{v^2}} \implies t = \frac{v}{g} + \frac{v}{g}\sqrt{1 + 2\frac{gh}{v^2}}, \tag{5.7}$$

which is the same time we found before.

More generally, suppose we're analyzing a more complicated phenomenon, and we have a complete set of dimensionless products $\{\pi_j\}_{j=1}^n$ for the variables in the problem. Then Buckingham's Theorem asserts that the equation we're looking for can be expressed as

$$f(\pi_1, \pi_2, \ldots, \pi_n) = 0. \tag{5.8}$$

At this point, let's recall a standard theorem from analysis known as the Implicit Function Theorem. In the current setting, this theorem asserts that if f is continuously differentiable and there is a vector $\vec{\pi}^*$ so that $f(\vec{\pi}^*) = 0$ and $\frac{\partial f}{\partial \pi_n}(\vec{\pi}^*) \neq 0$, then at least near $\vec{\pi}^*$ we can solve (5.8) with

$$\pi_n = \phi(\pi_1, \ldots, \pi_{n-1})$$

for some function ϕ.[31] For example, we see from our calculations above that for Example 5.1a we have

$$\phi(\pi_1) = 1 + \sqrt{1 + 2\pi_1}.$$

To be clear, just as we don't generally have an explicit form for f, we don't generally have an explicit form for ϕ, and in addition to this, since we don't have an explicit form for f we can't generally even check that the Implicit Function Theorem applies. Nonetheless, in applications of the general method of dimensional analysis, we typically assume such a ϕ exists.

At this point, there are two standard ways to take advantage of the method: (1) regression and (2) structured experiments. We will start with the former.

[31] See the appendix for a full statement of the Implicit Function Theorem.

5.2.3 Dimensional Analysis with Regression

Returning yet again to Example 5.1a, suppose we have data

$$\{(v_k, g_k, h_k, t_k)\}_{k=1}^N,$$

and as usual we would like to find t as a function of v, g, and h. We can use this data to compute *dimensionless data* $\{(\pi_{k1}, \pi_{k2})\}_{k=1}^N$, where

$$\pi_{k1} = \frac{g_k h_k}{v_k^2}, \quad \pi_{k2} = \frac{g_k t_k}{v_k}, \quad k = 1, 2, \ldots, N,$$

and we can then use this data and regression to identify the function ϕ that arises from the combination of Buckingham's Theorem and the Implicit Function Theorem. It's important to observe at this point that a critical simplification has occurred. Originally, we wanted to find $t = t(v, g, h)$, that is, t as a function of three independent variables. In principle, we could do this directly from our data with regression, but we would have a quite difficult regression problem, impossible to visualize since it would be in four-dimensional space. On the other hand, now that we've reduced the relationship to one between only two values, π_1 and π_2, we have a two-dimensional regression problem, easy to visualize. In general, each fundamental, dimension in an application will allow us to reduce the number of independent variables by one. This is because the number of variables is the number of columns of A and the number of fundamental dimensions is the number of rows, which is generally (though not always) the rank of A. So, according to the Rank-Nullity Theorem, the number of dimensionless products is the number of variables minus the number of dimensions. For the example at hand, there are two fundamental dimensions, L and T, so we expect the number of independent variables to reduce by two, as observed.

Turning now to the full analysis, we emphasize that our goal is to use our dimensionless data $\{(\pi_{k1}, \pi_{k2})\}_{k=1}^N$ and regression to find ϕ so that

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

We will carry this out with the M-file *da1.m*, which accomplishes the following:

- plots the dimensionless data;
- fits the dimensionless data to a regression line;
- compares model times with precise times (computed from the solution formula), with times viewed as a function of height, with v and g held fixed.

```
%DA1: MATLAB script M-file that carries out a dimensional
%analysis calculation for the example in which an object is
%fired up from a height h with velocity v.
%
%Data
```

```
g = 9.81; v = 8.61;
h = 0:.2:2;
t = [1.76 1.78 1.80 1.82 1.84 1.86 1.89 1.91 1.92 1.94 1.96];
%
pi1 = g*h/v^2;
pi2 = g*t/v;
%
plot(pi1, pi2, 'o')
axis equal
title('Dimensionless Products for the Data','Interpreter','latex','FontSize',16)
xlabel('$\pi_1$','Interpreter','latex','FontSize',14)
ylabel('$\pi_2$','Interpreter','latex','FontSize',14)
set(get(gca,'YLabel'),'Rotation',0.0)
pause
p = polyfit(pi1,pi2,1)
plot(pi1,pi2,'o',pi1,p(1)*pi1+p(2))
axis equal
title('Dimensionless Products, Data and Fit','Interpreter','latex','FontSize',16)
xlabel('$\pi_1$','Interpreter','latex','FontSize',14)
ylabel('$\pi_2$','Interpreter','latex','FontSize',14)
set(get(gca,'YLabel'),'Rotation',0.0)
pause
%
%Compare model with exact solution
modelt = p(1)*h/v+p(2)*v/g;
exactt = (v/g)+sqrt(v^2+2*g*h)/g;
plot(h,modelt,'o',h,exactt,'*')
legend('Model Times','Calculated Times','location','Northwest','Interpreter','latex')
title('Model Time Predictions with Calculated Values ','Interpreter','latex','FontSize',16)
xlabel('Height $h$','Interpreter','latex','FontSize',14)
ylabel('Time $t$','Interpreter','latex','FontSize',14)
pause
%
h=0:.2:5;
modelt = p(1)*h/v+p(2)*v/g;
exactt = (v/g)+sqrt(v^2+2*g*h)/g;
plot(h,modelt,'o',h,exactt,'*')
legend('Model Times','Calculated Times','location','Northwest','Interpreter','latex')
title('Model Time Predictions with Calculated Values ','Interpreter','latex','FontSize',16)
xlabel('Height $h$','Interpreter','latex','FontSize',14)
ylabel('Time $t$','Interpreter','latex','FontSize',14)
pause
h=0:1:50;
modelt = p(1)*h/v+p(2)*v/g;
exactt = (v/g)+sqrt(v^2+2*g*h)/g;
```

```
plot(h,modelt,'o',h,exactt,'*')
legend('Model Times','Calculated Times','location','Northwest','Interpreter','latex')
title('Model Time Predictions with Calculated Values ','Interpreter','latex','FontSize',16)
xlabel('Height $h$','Interpreter','latex','FontSize',14)
ylabel('Time $t$','Interpreter','latex','FontSize',14)
```

First, when working with dimensionless products, since scaling issues have typically been eliminated, it's often convenient to set the axes equal on plots, as done in *da1.m*. The first step is to identify $\phi(\pi_1)$, and we see from Figure 5.2 that in this case a linear fit is natural.

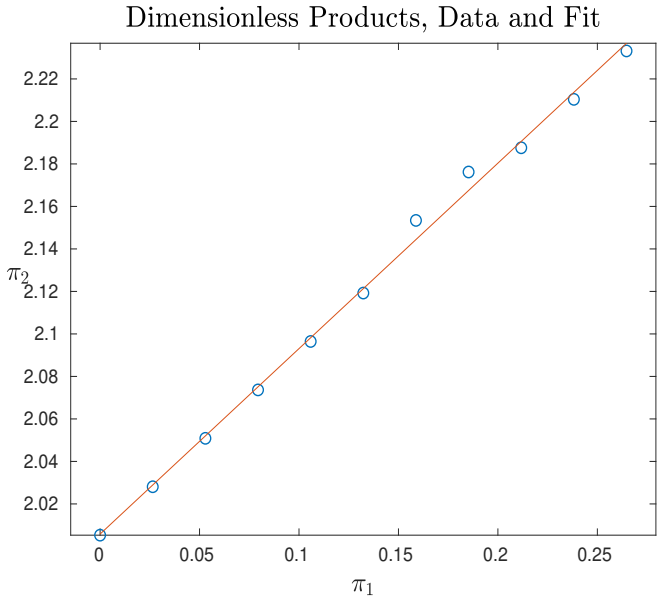


Figure 5.2: Fit for $\pi_2 = \phi(\pi_1)$ for Example 5.1a.

We find $m = .8727$ and $b = 2.0058$, so that

$$\phi(\pi_1) = .8727\pi_1 + 2.0058,$$

giving us the model

$$t = \frac{v}{g}(.8727\frac{gh}{v^2} + 2.0058). \tag{5.9}$$

We can now compute landing times t in two different ways, the first based on our exact solution (5.7), and the second based on (5.9). For comparison, we evaluate t as h varies, first for small values of h and then for larger values. See Figure 5.3.

From the figure, we find that for small values of h (and so small values of π_1), our model is quite good, but as h increases it degrades substantially. In order to understand why this happens, let's recall that the exact form of ϕ in this case is $\phi(\pi_1) = 1 + \sqrt{1 + 2\pi_1}$. If we compute a Taylor polynomial approximation of ϕ about $\pi_1 = 0$, we find

$$\phi(\pi_1) = 2 + \pi_1 + \boldsymbol{O}(\pi_1^2).$$

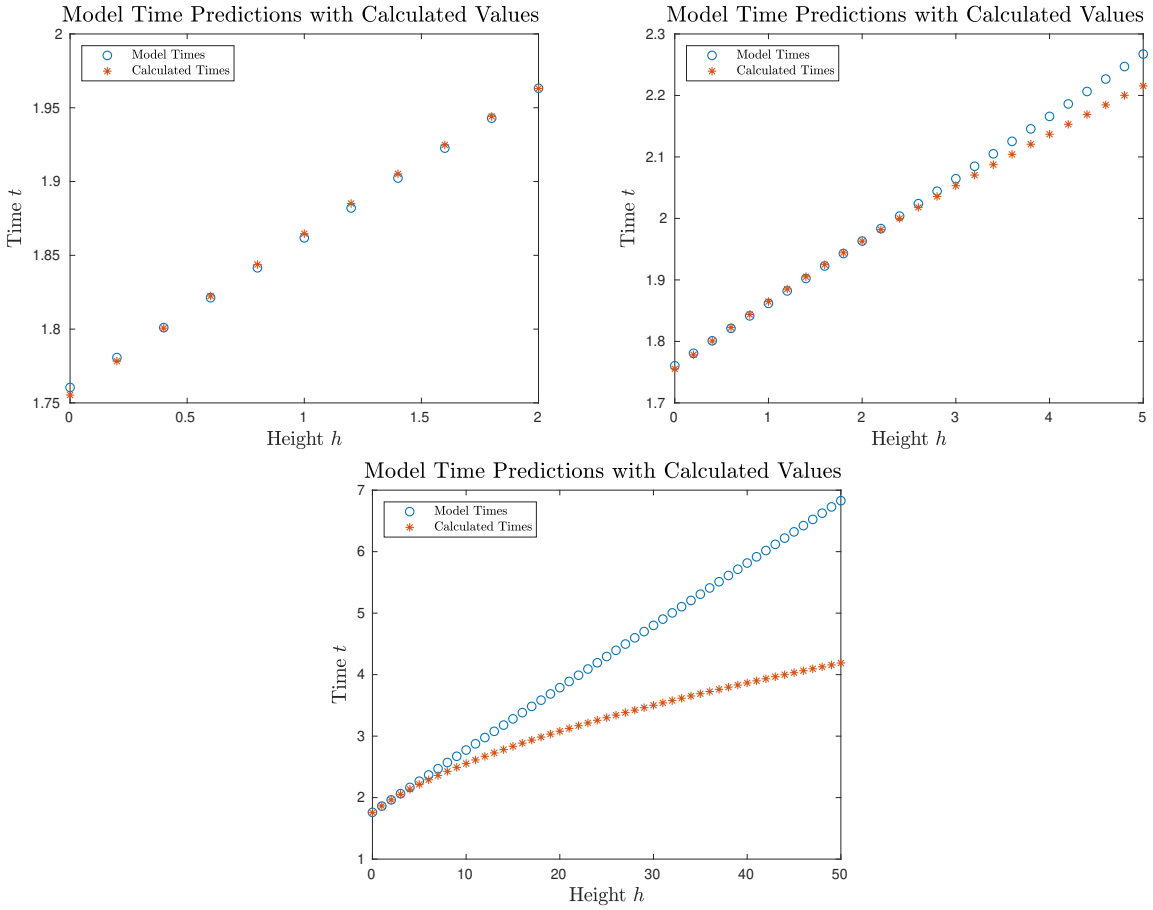


Figure 5.3: Comparison of model values and exact values for Example 5.1a.

Since our values of π_1 are fairly small, this is the behavior our model is capturing: the intercept is almost exact, and the slope isn't all that far off. But since $\phi(\pi_1)$ isn't a linear function, there is no chance that with the data we have we will be able to find an appropriate form of ϕ for all values of the variables. Generally, given a sufficiently small range of data, the function ϕ will be linear; for a slightly larger range of data it will be quadratic (second order Taylor expansion) and so on. This observation is especially important in the case of multiple parameters for which the fit becomes difficult to visualize.

By the way, the choice $\pi_1 = 0$ is convenient for the intuitive discussion above, but it's not quite right conceptually. It makes more sense to view the line as an approximation in the middle of the data, so we might Taylor expand ϕ about the average $\bar{\pi}_1$ of the values of π_1 used for the fit. In this case, $\bar{\pi}_1 = .1323$, and we find $\phi(\bar{\pi}_1) = 2.1246$ and $\phi'(\bar{\pi}_1) = .8892$ so that

$$\begin{aligned}\phi(\pi_1) &\cong \phi(\bar{\pi}_1) + \phi'(\bar{\pi}_1)(\pi_1 - \bar{\pi}_1)\\ &= 2.1246 + .8892(\pi_1 - .1323)\\ &= .8892\pi_1 + 2.0070,\end{aligned}$$

closer to the line obtained from our regression fit.

Before considering a more involved example of this method, we review the key steps.

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_n.
3. Apply Buckingham's Theorem to obtain the existence of a (typically unknown) function f, for which the (also unknown) equation relating the variables in our problem can be expressed as
$$f(\pi_1, \pi_2, ..., \pi_n) = 0.$$
4. Apply the Implicit Function Theorem to obtain the existence of a (typically unknown) function ϕ presumably satisfying
$$\pi_n = \phi(\pi_1, \pi_2, ..., \pi_{n-1}).$$
5. Use experimental data to determine the form of ϕ from Step 4.
6. Replace the dimensionless products with the multiplicative variable combinations they denote, and solve the equation from Step 4 for the dependent variable.

Example 5.5. The first atomic bomb, code-named Trinity, was detonated in the New Mexico desert on July 16, 1945. The energy released was classified, but in 1947 photographs of the explosion were declassified. Based on these images, the British mathematician and physicist Geoffrey Ingram (G.I.) Taylor (1886-1975) was able to estimate the energy released.

His analysis was published in 1950 in Proceedings of the Royal Society of London.[32] The Trinity explosion tested a plutonian-based design (like the "fat man" bomb detonated over Nagasaki), the date chosen because President Truman would be meeting with Churchill and Stalin on July 17, and Truman thought it would strengthen his position. In this example, we will use dimensional analysis to obtain most of Taylor's results, with the exception of one calculation requiring partial differential equations, which he did in a companion paper (also listed in Footnote 32).

To begin, let's suppose we would like to find an expression for the radius R of the shock wave associated with the explosion, based on the following variables (with ambient terms prior to explosion):

$$\begin{aligned} \text{time since explosion } & t \\ \text{energy released in exlosion } & E \\ \text{ambient air density } & \rho \\ \text{ambient air pressure } & p \end{aligned}$$

As usual, we first identify the dimensionless products for the problem, writing

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

Equating dimensions on the two sides of this relation, we obtain

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

leading to the dimensions equations

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

We can express this system in matrix form,

$$\begin{pmatrix} 0 & 2 & -3 & -1 & 1 \\ 0 & 1 & 1 & 1 & 0 \\ 1 & -2 & 0 & -2 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

For convenient reference, we will denote the matrix in this last relation A. Reducing A to RRE form, we obtain

$$\begin{pmatrix} 1 & 0 & 0 & -\frac{6}{5} & \frac{2}{5} \\ 0 & 1 & 0 & \frac{2}{5} & \frac{1}{5} \\ 0 & 0 & 1 & \frac{3}{5} & -\frac{1}{5} \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

[32] G.I. Taylor, *The formation of a blast wave by a very intense explostion II: the atomic explosion of 1945*, Proceedings of the Royal Society of London, Series A, vol. 201 (1950), no. 1065, pp. 175-186. We will also refer to Taylor's earlier paper, *The formation of a blast wave by a very intense explosion I: theoretical discussion*, Proceedings of the Royal Society of London, Series A, vol. 201 (1950), no. 1065, pp. 159-174.

We see from this expression that rank $A = 3$ so that nullity $A = 2$, and we expect to find two dimensionless products (forming a complete set). We can express the RRE form of our system as

$$\begin{aligned} a &= \frac{6}{5}d - \frac{2}{5}e \\ b &= -\frac{2}{5}d - \frac{1}{5}e \\ c &= -\frac{3}{5}d + \frac{1}{5}e. \end{aligned}$$

For π_1, we choose $e = 0$ and $d = 1$ so that $a = \frac{6}{5}$, $b = -\frac{2}{5}$, and $c = -\frac{3}{5}$, giving

$$\pi_1 = \frac{pt^{6/5}}{E^{2/5}\rho^{3/5}}.$$

For π_2, we choose $e = 1$ and $d = 0$ so that $a = -\frac{2}{5}$, $b = -\frac{1}{5}$, and $c = \frac{1}{5}$, giving

$$\pi_2 = \frac{R\rho^{1/5}}{t^{2/5}E^{1/5}}.$$

At this point, Buckingham's Theorem asserts that there exists a function f so that we can express the relation between our variables that we're searching for as

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

and the Implicit Function Theorem *suggests* that we can solve this by finding a function ϕ so that

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

For the standard (regression-based) approach, we would collect data at this point of the form

$$\{(t_k, E_k, \rho_k, p_k, R_k)\}_{k=1}^N,$$

and use it to compute dimensionless data $\{(\pi_{k1}, \pi_{k2})\}_{k=1}^N$. For this application, however, the value we're trying to identify is E (which is the same for all k), so no such data is available. In order to get around this difficulty, let's first observe that since E is presumably quite large, we can assert, at least for small times, that $\pi_1 \cong 0$. In this case, the relation $\pi_2 = \phi(\pi_1)$ reduces to approximately $\pi_2 \cong \phi(0)$, which we an express as

$$\frac{R\rho^{1/5}}{t^{2/5}E^{1/5}} \cong \phi(0) \implies R(t) \cong (\frac{E^{1/5}\phi(0)}{\rho^{1/5}})t^{2/5}.$$

Notice that we can view $\phi(0)$ as a zeroth order Taylor polynomial for ϕ, so it would be in the spirit of our approach to obtain a value for $\phi(0)$ by carrying out experiments with light explosives, for which π_1 can be kept small by taking measurements at small times. In lieu of this, in the second paper listed in Footnote 32 G.I. Taylor used an analysis based on partial differential equations to show that $\phi(0) \cong 1.0316$.[33]

[33]In fact, Taylor had written this paper in 1941, but it was classified by the British government, and wasn't published until 1950.

Rather than the full set of data mentioned above, the data Taylor had to work with, taken from the images declassified in 1947, had the form $\{(t_k, R_k)\}_{k=1}^N$. Observing that the quantity

$$c = \frac{E^{1/5}\phi(0)}{\rho^{1/5}}$$

is constant, we can use this data to find a best-fit value for c from the relation $R = ct^{2/5}$. Once c is identified, we can use Taylor's value for $\phi(0)$ along with the measured ambient air density $\rho = 1.25\ kg/m^3$ to compute

$$E = (\frac{c\rho^{1/5}}{\phi(0)})^5.$$

This will be the energy released by the detonation.

Turning to the specific calculations, we will use the MATLAB M-file *taylordatafit.m*, given below. First, we check the basic form of our model by writing

$$\ln R = \ln c + \frac{2}{5}\ln t,$$

and checking that if we fit $\ln R$ as a function of $\ln t$, we indeed obtain a slope of $\frac{2}{5}$. Next, we carry out the fit for $R = ct^{2/5}$ in our usual way, which involves (since we only have one parameter) the design matrix

$$F = \begin{pmatrix} t_1^{2/5} \\ t_2^{2/5} \\ \vdots \\ t_N^{2/5} \end{pmatrix}.$$

(We observe that we cannot simply fit R as a function of $t^{2/5}$ and take c to be the slope, because that would introduce a non-zero intercept.) Last, we will repeat Taylor's approach, which was to write

$$\frac{5}{2}\log_{10} R - \log_{10} t = \frac{5}{2}\log_{10} c,$$

and to think of fitting $Y := \frac{5}{2}\log_{10} R - \log_{10} t$ with only an intercept.

The data associated with these calculation is provided in the MATLAB M-file *taylordata.m*.

```
%TAYLORDATA: MATLAB script M-file that defines data
%for the first atomic explosion in New Mexico (code-named Trinity,
%July 16, 1945). This is taken from p. 176 of Taylor's
%paper "The formation of a blast wave by a very intense explosion II:
%the atomic explosion of 1945," in Proceedings of the Royal Society
%of London, Series A, vol. 201 (1950), no. 1065, pp. 175-186.
%The companion paper is "The formation of a blast wave by a
%very intense explosion I: theoretical discussion," in
%in Proceedings of the Royal Society of London, Series A,
%vol. 201 (1950), no. 1065, pp. 159-174.
```

```
t = 1e-3*[.1 .24 .38 .52 .66 .80 .94 1.08 1.22 1.36 1.50 1.65 1.79 1.93 3.26 3.53
3.80 4.07 4.34 4.61 15.0 25.0 34.0 53.0 62.0]; %in seconds
R = [11.1 19.9 25.4 28.8 31.9 34.2 36.3 38.9 41.0 42.8 44.4 46.0 46.9 48.7 59.0
61.1 62.9 64.3 65.6 67.3 106.5 130.0 145.0 175.0 185.0]; %in meters
rho = 1.25; %kg/m^3
phi0 = 1.0316; %dimensionless
```

The described calculations are now carried out with the MATLAB M-file *taylordatafit.m*.

```
%TAYLORDATAFIT: MATLAB script M-file that analyzes the data stored
%in taylordata.m
%
%define and fit the data
taylordata;
%
%VERIFICATION OF THE 2/5 POWER LAW
plot(log(t),log(R),'o')
title('Data Plot: ln R Against ln t','Interpreter','latex','FontSize',16)
xlabel('$\ln t$','Interpreter','latex','FontSize',14)
ylabel('$\ln R$','Interpreter','latex','FontSize',14)
pause
p=polyfit(log(t),log(R),1);
plot(log(t),log(R),'o',log(t),p(1)*log(t)+p(2))
title('Data and Fit: ln R Against ln t','Interpreter','latex','FontSize',16)
xlabel('$\ln t$','Interpreter','latex','FontSize',14)
ylabel('$\ln R$','Interpreter','latex','FontSize',14)
legend('Data','Fit','location','Northwest','Interpreter','latex')
fprintf('Slope = %5.4f; Should be .4000.\n',p(1));
pause
%
%METHOD 1 FOR COMPUTING C: DIRECT FIT
c = t'.^(2/5)\R' %i.e., F = t'.^(2/5)
%
%Compute E
E = (c*rho^(1/5)/phi0)^5 %in Joules (kg m^2/s^2)
pause
%One metric ton (1,000 kg or 2,204.62 lbs) of TNT corresponds, by definition,
to
%an energy of 4.184e9 Joules, but the definition was different in 1950,
%and Taylor used 4.25e9.
Etons = E/4.25e9;
fprintf('This corresponds with %7.2f metric tons of TNT\n',Etons);
pause
%
%METHOD 2 FOR COMPUTING C: TAYLOR'S APPROACH
```

```
%NOTE: Taylor used cm, and log10
Y = (5/2)*log10(100*R)-log10(t);
F = ones(size(R))';
ctemp = F\Y'
fprintf('Taylor reported 11.915; see p. 176. Probably round-off error (on his part).\n')
pause
our_taylorc = 10^((2/5)*ctemp)
our_taylorc_our_units = our_taylorc/100 %Return from cm to m.
ourEtaylor = (our_taylorc_our_units*rho^(1/5)/phi0)^5 %in Joules (kg m^2/s^2)
pause
ourEtaylortons = ourEtaylor/4.25e9;
fprintf('This corresponds with %7.2f metric tons of TNT\n',ourEtaylortons);
pause
fprintf('Here is the calculation Taylor actually did.\n')
taylorc = 10^((2/5)*11.915)
fprintf('This corresponds with c^5 = %7.2e \n',taylorc^5)
fprintf('The value he computed with was c^5 = 6.67 x 10^23 cm^5 s^-2. More round-off error? \n')
pause
%There seems to be a typesetting error on p. 177 of Taylor's second paper.
%The value of c^5 is reported as c^5 = 6.67 x 10^2, suggesting the 3 was
%dropped off by the typesetter. In our units, c^5 = 6.67 x 10^13.
Etaylor = (6.67e13*rho/phi0^5)/4.25e9
fprintf('This corresponds with %7.2f metric tons of TNT\n',Etaylor);
fprintf('Taylor reported 16,800 metric tons. In most references the energy is\n')
fprintf('now reported as 20,000 metric tons.\n')
```

The full output from *taylordatafit.m* is included below.

```
>>taylordatafit
Slope = 0.4058; Should be .4000.
c =
570.5925
E =
6.4712e+13
This corresponds with 15226.31 metric tons of TNT
ctemp =
11.9038
Taylor reported 11.915; see p. 176. Probably round-off error (on his part).
our_taylorc =
5.7748e+04
our_taylorc_our_units =
577.4770
```

ourEtaylor =
6.8711e+13
This corresponds with 16167.31 metric tons of TNT
Here is the calculation Taylor actually did.
taylorc =
5.8345e+04
This corresponds with c^5 = 6.76e+23
The value he computed with was c^5 = 6.67 x 10^23 cm^5 s^-2. More round-off error?
Etaylor =
1.6792e+04
This corresponds with 16791.53 metric tons of TNT
Taylor reported 16,800 metric tons. In most references the energy is now reported as 20,000 metric tons.

The first thing we observe from the output is that in the fit of $\ln R$ as a function of $\ln t$ we obtain a slope of .4058, very close to the expected $\frac{2}{5}$ (see Figure 5.4). Next, by finding c with out standard technique (i.e., using the design matrix F given above), we find that the energy released in the detonation was approximately $E = 6.4712 \times 10^{13}$ Joules, which translates to about 15,226.31 metric tons of TNT.[34] If we find the value of c by Taylor's method, but with our computational technolgy (i.e., MATLAB), we obtain an energy of 16,167.31 metric tons of TNT, and finally if we use Taylor's value of c we obtain 16,791.53 metric tons of TNT. The value Taylor actually reported was 16,800 metric tons of TNT. In most references, the energy released in the Trinity explosion is now reported to be about 20,000 metric tons of TNT. For a quick comparison, the uranium-based "little boy" bomb detonated over Hiroshima on Aug. 6, 1945 is typically reported to have released about 12,00 metric tons of TNT, and the plutonium-based "fat man" bomb detonated over Nagasaki is typically reported to have released about 20,000 metric tons. Hydrogen bombs, fortunately never detonated over any city, are reported to release about 25 million metric tons of TNT.[35]

5.2.4 Dimensional Analysis with Structured Experiments

Once again let's start with Example 5.1a before considering a more realistic application. We recall (though, at this point, how could we forget?) that for that example, we have the dimensionless products $\pi_1 = \frac{gh}{v^2}$ and $\pi_2 = \frac{gt}{v}$, related by the equation $\pi_2 = \phi(\pi_1)$. Suppose we're interested in computing the landing time t for some particular values v, g, and h, but that for some reason it would be difficult to arrange an experiment with those values. For example, v or h could be prohibitively large, or we could be trying to determine t on the

[34]The energy released in large explosions is often measured in metric tons of TNT, i.e., in the number of metric tons of TNT that would be required to achieve that energy. By the current definition, one metric ton of TNT is equivalent to exactly 4.184×10^9 Joules of energy, though at the time that Taylor was working this definition was slightly different, and one metric ton of TNT was defined to be exactly 4.25×10^9 Joules of energy.

[35]The first design for an atomic bomb was designated "thin man," but researchers soon found that it was too simplistic and would release low energy over a long period of time, rather than the blast they were trying to achieve. "Thin man" was plutonium-based.

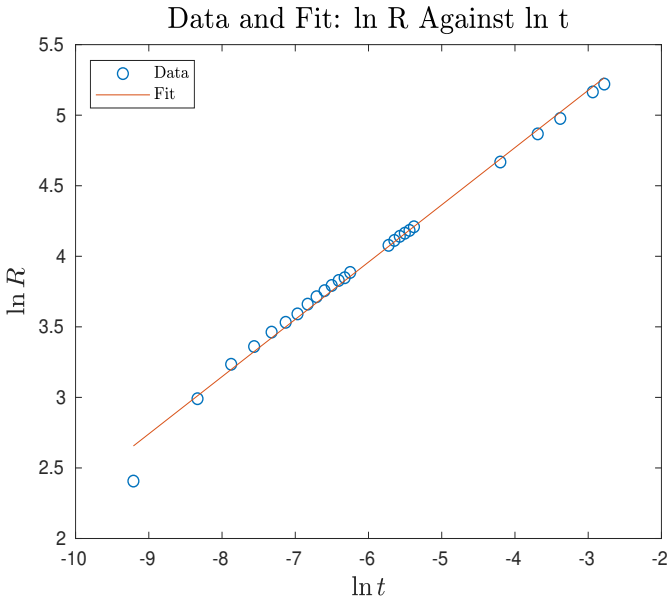


Figure 5.4: Fit of $\ln R$ as a function of $\ln t$ for the Trinity detonation.

moon or Mars via an experiment on Earth. In this case, we can view v, g, and h as given values, and we can design (i.e., *structure*) an experiment with values v_e, g_e, and h_e so that $\pi_1 = \pi_{e1}$, where $\pi_{e1} = \frac{g_e h_e}{v_e^2}$. For example, if v is large, we might take v_e smaller, but also take h_e smaller to achieve the equality. With these choices of v_e, g_e, and h_e, we carry out an experiment and *measure* a value t_e. This allows us to compute $\pi_{e2} = \frac{g_e t_e}{v_e}$, so that π_1, π_{e1}, and π_{e2} are fully specified numerical values. From these values, we can obtain the value of π_2 by computing

$$\pi_2 = \phi(\pi_1) = \phi(\pi_{e1}) = \pi_{e2},$$

which we often summarize by writing

$$\pi_1 = \pi_{e1} \implies \pi_2 = \pi_{e2}.$$

Having identified π_2, we can solve for the time t that we're looking for,

$$t = \frac{v}{g}\pi_2 = \frac{v}{g}\pi_{e2}.$$

Example 5.6. Suppose we want to compute the landing time t for $g = 1.63$ m/s^2 (i.e., on the moon), with $v = 2$ m/s and $h = 5$ m.

First, we compute

$$\pi_1 = \frac{gh}{v^2} = 2.0375.$$

This means that we need to arrange an experiment so that

$$\pi_{e1} = \frac{g_e h_e}{v_e^2} = 2.0375.$$

Notice that we have considerable flexibility in how we go about this: we can choose any two of the values of g_e, h_e, and v_e and solve for the third. For this example, let's take $g_e = 9.81$ m/s^2 and $v_e = 8.61$ m/s, the latter of which is the velocity with which darts fire from a particular dart gun that will be used to carry out the experiment. Having chosen values for g_e and v_e, we can compute

$$h_e = 2.0375\frac{8.61^2}{9.81} = 15.3970 \text{ m}.$$

At this point, we carry out an experiment, and in this case the value

$$t_e = 2.85 \text{ s}$$

was measured. This allows us to compute

$$\pi_{e2} = \frac{g_e t_e}{v_e} = \frac{9.81 \cdot 2.85}{8.61} = 3.2472.$$

Finally, we can find the time t we're looking for by computing

$$t = \frac{v}{g}\pi_{e2} = \frac{2}{1.63}3.2472 = 3.9843 \text{ s}.$$

△

Example 5.7. When designing airplanes, it's impractical to carry out experiments on full-sized planes, so one approach engineers take is to use dimensional analysis to design wind tunnel experiments. In this example, we'll see how this process works.

Very roughly, an airplane wing in flight (the shape of which is called an *airfoil*) can be characterized by the following variables:

angle of inclination (or "attack") θ
length in the direction of flight ("chord") r
length from fuselage to wingtiop s.

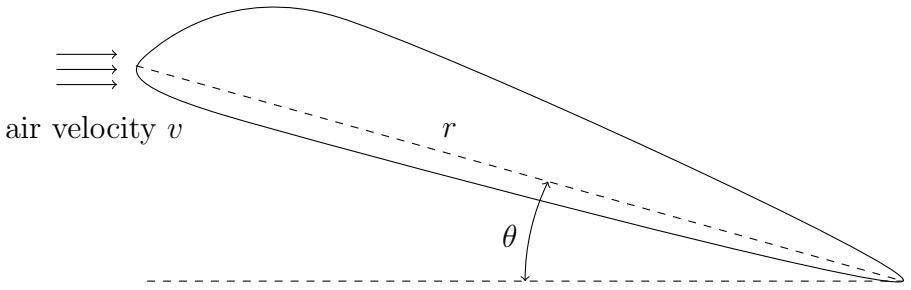


Figure 5.5: Airfoil for Example 5.7.

It's customary to depict an airfoil in the configuration of Figure 5.5, because we typically analyze it by thinking of the airfoil as remaining still while air flows in from the left. In

addition to the values θ, r, and s, the lift force F_l and the drag force F_d depend on the following:

$$\begin{aligned} \text{air density} &\quad \rho \\ \text{air viscosity} &\quad \mu \\ \text{velocity of the plane (or air flow)} &\quad v \\ \text{Mach number} &\quad \mathcal{M} = \frac{v}{v_s}, \end{aligned}$$

where v_s denotes the speed of sound in the medium (about 340.3 m/s (761.2 mph) in air).[36]

The lift force F_l and drag force F_d can be analyzed separately, and the analysis of each is precisely the same up to the point of actually conducting experiments, so we will focus on only one, lift force. At this point, we have eight variables to work with, so we look for dimensionless products of the form

$$\pi = \pi(\theta, r, s, \rho, \mu, v, \mathcal{M}, F_l) = \theta^a r^b s^c \rho^d \mu^e v^f \mathcal{M}^g F_l^h.$$

Equating the dimensions on either side of this relation, we have (noting that θ and $\mathcal{M}$ are dimensionless)

$$1 = L^b L^c M^d L^{-3d} M^e L^{-e} T^{-e} L^f T^{-f} M^h L^h T^{-2h},$$

from which we obtain the dimensions equations

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

In matrix form, we can express this system as

$$\begin{pmatrix} 0 & 1 & 1 & -3 & -1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & -1 & 0 & -2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

If we put the matrix in RRE form, we get

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

[36]The Mach number is named for the Austrian physicist and philosopher Ernst Mach (1838-1916). It's important in the determination of whether a fluid can be treated as incompressible; typically, small Mach numbers ($\mathcal{M} < .2$) correspond with incompressibility.

We see that the rank of the matrix is 3, and since it has eight columns this means the nullity is 5, from which we know to identify five dimensionless products. Keying on the embedded identity matrix associated with c, d, and e, we can express solutions of this system as

$$\begin{aligned} c &= -b + f \\ d &= f + h \\ e &= -f - 2h. \end{aligned}$$

We have the freedom to choose five values a, b, f, g, and h, from which the values c, d, and e will be determined. We will do this in a systematic way, choosing each of the free values to be 1 with the others all zero: for π_1, $a = 1$, so $c = 0$, $d = 0$, and $e = 0$, giving $\pi_1 = \theta$. For π_2, $b = 1$, so $c = -1$, $d = 0$, and $e = 0$, giving $\pi_2 = \frac{r}{s}$. For π_3, $f = 0$, so $c = 1$, $d = 1$, and $e = -1$, giving $\pi_3 = \frac{s\rho v}{\mu}$.[37] For π_4, $g = 1$, so $c = 0$, $d = 0$, and $e = 0$, giving $\pi_4 = \mathcal{M}$. For π_5, $h = 1$, so $c = 0$, $d = 1$, and $e = -2$, giving $\pi_5 = \frac{\rho F_l}{\mu^2}$.

According to Buckingham's Theorem and the Implicit Function Theorem, we expect to have the relation

$$\pi_5 = \phi(\pi_1, \pi_2, \pi_3, \pi_4)$$

for some appropriate function ϕ. Now, suppose we want to compute F_l for some values θ, r, s, ρ, μ, v, and $\mathcal{M}$ for which it would be difficult to carry out an experiment in practice. We can design a wind tunnel experiment with values θ_e, r_e, s_e, ρ_e, μ_e, v_e, and $\mathcal{M}_e$ chosen so that

$$\begin{aligned} \theta_e &= \theta \\ \frac{r_e}{s_e} &= \frac{r}{s} \\ \frac{s_e \rho_e v_e}{\mu_e} &= \frac{s\rho v}{\mu} \\ \mathcal{M}_e &= \mathcal{M}. \end{aligned}$$

It follows that

$$\frac{F_l \rho}{\mu^2} = \frac{F_{le} \rho_e}{\mu_e^2}.$$

In this way, we can measure F_{le} from our wind tunnel experiment and find F_l by computing

$$F_l = \frac{\mu^2}{\rho} \frac{F_{le} \rho_e}{\mu_e^2}.$$

[37] The combination $\frac{s\rho v}{\mu}$ is an important quantity in fluid mechanics known as Reynolds number (in recognition of the Irish engineer Osbourne Reynolds (1842-1912)). If we replace s with any other length such as r we continue to call the result a Reynolds number. The Reynolds number can often be used to predict the onset of turbulent flow; in particular, turbulent flow typically corresponds with large values of the Reynolds number. The threshold varies according to the fluid and flow configuration (e.g., pipe, channel, etc.), but is typically in the range of about 1,000 to 5,000.

5.2.5 Aside on Viscosity

As we saw in Example 5.7, viscosity has an important role in the study of fluid dynamics, so it's worth taking some time to make sure we understand how it works. First, to get some intuition, let's imagine that we're walking across the Texas A&M campus, trying to get to class etc. Density would be analogous to the number of students out walking around, and so high density would certainly impede our progress. Viscosity, on the other hand, is analogous to the number of people we actually know and might stop and talk to. It also impedes our progress, but in a different way. Likewise, for fluids, viscosity is a measure of the tendency of molecules to stick together.

For a precise definition of viscosity, we suppose a thin plate with area A is moved through a viscous fluid with velocity v, above a fixed flat surface, as depicted in Figure 5.6. By definition, the fluid is referred to as *Newtonian* if the force required to keep the plate moving with velocity v is proportional to $\frac{Av}{h}$; i.e., if there exists a constant μ so that this force F can be expressed as

$$F = \mu \frac{Av}{h}.$$

This constant μ is typically the value we give as a measure of viscosity.[38] In order to understand the dimensions of viscosity, we can solve for μ, writing then

$$[\mu] = \frac{[F][h]}{[A][v]} = \frac{(MLT^{-2})L}{L^2(LT^{-1})} = ML^{-1}T^{-1}.$$

The standard unit for viscosity is the Pascal-second, which has units kg $\mathrm{m}^{-1}\mathrm{s}^{-1}$. Common Newtonian fluids include air, water, and thin motor oil. At 20^o C (68^o F), these have respective viscosities 1.983×10^{-5} Pa $\cdot$ s, $.001 \times 10^{-5}$ Pa $\cdot$ s, and about $.250 \times 10^{-5}$ Pa $\cdot$ s. Common non-Newtonian fluids include thicker motor oil, blood, ketchup, honey, and of course oobleck.[39] The viscosity of a non-Newtonian fluid varies depending on the force applied.

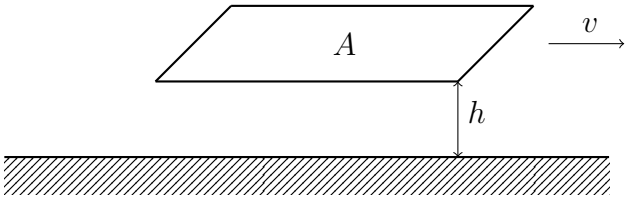


Figure 5.6: Thin plate moving through a viscous fluid.

5.3 More Matrix Theory

In this section, we include details about the matrix theory that arose in our discussion of the general method of dimensional analysis. We start with the following definition.

[38] The value μ is sometimes referred to as *dynamic viscosity*, as opposed to *kinematic viscosity*. Kinetmatic viscosity is $\nu = \frac{\mu}{\rho}$, where ρ is the density of the fluid.

[39] One part water, two parts corn starch.

Definition 5.1. *For any matrix $A \in \mathbb{C}^{m\times n}$, the column rank is the dimension of the linear space spanned by the columns of the matrix, and the row rank is the dimension of the linear space spanned by the rows of the matrix.*

Next, we introduce a lemma that we often use without explicit reference.

Lemma 5.1. *For any matrix $A \in C^{m\times n}$, the column rank is conserved under both column and row operations, and likewise the row rank is conserved under both column and row operations.*

Proof. We'll prove this for the case of row operations; the proof in the case of column operations is similar. First, since row operations on a matrix simply correspond with linear combinations of the rows in the matrix, the statement about row rank is fairly clear. Nonetheless, let's write out a short calculation to drive the point home. Precisely, a single row operation on a matrix (that isn't just a row exchange) corresponds with replacing one of the rows in the matrix with a linear combination of that row and another. For example, if the rows of the original matrix A are expressed as row vectors $\{\vec{r}_j\}_{j=1}^m$, and a multiple of the second row is added to the first row, the resulting matrix has first row $\vec{r}\,'_1 = \vec{r}_1 + c\vec{r}_2$ and remaining rows unchanged (i.e., $\{\vec{r}_j\}_{j=2}^m$). We now easily check that any vector $\vec{v} \in \mathbb{C}^m$ can be obtained as a linear combination of the rows of A if and only if it can be obtained as a linear combination of the rows of the reduced matrix. First, if

$$\vec{v} = \sum_{j=1}^{m} c_j \vec{r}_j$$

then

$$\vec{v} = c_1\vec{r}\,'_1 + (c_2 - cc_1)\vec{r}_2 + \sum_{j=2}^{m} c_j\vec{r}_j,$$

and second if

$$\vec{v} = c_1\vec{r}\,'_1 + \sum_{j=2}^{m} c_j\vec{r}_j$$

then

$$\vec{v} = c_1\vec{r}_1 + (c_2 + cc_2)\vec{r}_2 + \sum_{j=2}^{m} c_j\vec{r}_j.$$

A similar calculation can be carried out for any row operation, giving the statement about row ranks.

Turning to the case of column ranks, we recall from introductory courses in linear algebra that row operations can be carried out by multiplying the matrix A on the left by an appropriate elementary matrix, and that all elementary matrices are invertible with inverses being elementary matrices of the same type.[40] We let E denote some elementary matrix, and we express the columns of A as vectors $\{\vec{a}_j\}_{j=1}^n$ and the columns of EA as vectors $\{\vec{b}_j\}_{j=1}^n$. I.e., we can view A and EA as

$$A = \begin{pmatrix} \vec{a}_1 & \vec{a}_2 & \cdots & \vec{a}_n \end{pmatrix}, \quad EA = \begin{pmatrix} \vec{b}_1 & \vec{b}_2 & \cdots & \vec{b}_n \end{pmatrix}.$$

[40] See, for example, Section 1.5 in *Linear algebra with applications*, 9th Ed., by Steven J. Leon.

Now, suppose the matrix A has column rank $r \in \{0, 1, \ldots, n-1\}$ so that any $r+1$ columns of A must be linearly dependent. For ease of notation, let's denote these columns $\{\vec{a}_j\}_{j=1}^{r+1}$, though of course they do not need to be the first $r+1$ columns of A. Since the vectors $\{\vec{a}_j\}_{j=1}^{r+1}$ are linearly dependent, it must be the case that there exists a linear combination $\sum_{j=1}^{r+1} c_j \vec{a}_j = 0$, where the coefficients $\{c_j\}_{j=1}^{n}$ are not all 0. In this case, we can write

$$0 = E(\sum_{j=1}^{r+1} c_j \vec{a}_j) = \sum_{j=1}^{r+1} c_j E\vec{a}_j = \sum_{j=1}^{r+1} c_j \vec{b}_j,$$

so the corresponding columns of EA are also linearly dependent. Since this is true for all collections of $r+1$ columns of EA, we can conclude that

$$\text{column rank } (EA) \leq r = \text{column rank } (A). \tag{5.10}$$

In the event that $r = n$ (the one case left out above), this inequality holds trivially with $r = n$ since the column rank of the $m \times n$ matrix EA cannot exceed n. Since elementary matrices are invertible, if we set $\tilde{A} = EA$, then we will have $A = E^{-1}\tilde{A}$, from which the above argument, applied to $\tilde{A}$, with elementary matrix E^{-1}, yields

$$\text{column rank } (E^{-1}\tilde{A}) \leq \text{column rank } (\tilde{A}),$$

which is precisely the converse of (5.10). We conclude that

$$\text{column rank } (EA) = \text{column rank } (A),$$

as claimed. □

Theorem 5.3. *For any matrix $A \in \mathbb{C}^{m\times n}$, the rank of A, the column rank of A, and the row rank of A are equivalent.*

Proof. First, we recall that the rank of a matrix is the dimension of its range, where its range is the collection of all vectors that can be obtained by acting with the matrix on some vector. Precisely, the range $\mathcal{R}(A)$ is defined to be

$$\mathcal{R}(A) := \{\vec{b} \in \mathbb{C}^m : A\vec{v} = \vec{b} \text{ for some } \vec{v} \in \mathbb{C}^n\}.$$

Next, let's notice that the action of a matrix on a vector (i.e., the product $A\vec{v}$) can be viewed as the evaluation of a linear combination of the columns of A. That is, if we denote the n columns of A as vectors $\{\vec{a}_j\}_{j=1}^{n}$ so that A can be viewed as

$$A = \begin{pmatrix} \vec{a}_1 & \vec{a}_2 & \cdots & \vec{a}_n \end{pmatrix},$$

then

$$A\vec{v} = v_1\vec{a}_1 + v_2\vec{a}_2 + \cdots + v_n\vec{a}_n,$$

where as usual the values $\{v_j\}_{j=1}^{n}$ denote the components of the vector $\vec{v}$.

Turning now to the statements of the theorem, we see that the claim that the rank is equivalent to the column rank follows immediately from our two preliminary observations,

because $\mathcal{R}(A)$ is seen to be precisely the collection of vectors $\vec{b}$ that can be obtained as linear combinations of the columns of A.

For the claim that the row rank is equivalent to the column rank (and so also to the rank, by the first part), we let A' denote the RRE form of A (which, by Lemma 5.1 has the same row rank and column rank as A), and we note that if A has row rank s then in RRE form $\tilde{A}$ will necessarily have s rows with leading 1's (and all other rows 0).[41] At this point, we can perform column operations on $\tilde{A}$ (i.e., take linear combinations of the columns) to reduce the number of non-zero columns to at most s. (If this number was greater than s, we would have more than s linearly independent vectors of length s, and this is a contradiction.) Since the number of non-zero columns must be the column rank (and so the rank) of A, we can conclude that $r \leq s$. I.e., we see that

$$\text{column rank } A \leq \text{row rank } A.$$

On the other hand,

$$\begin{aligned} \text{row rank } A &= \text{column rank } A^T \\ &\leq \text{row rank } A^T \\ &= \text{column rank } A, \end{aligned}$$

allowing us to conclude

$$\text{column rank } A = \text{row rank } A.$$

This completes the proof. □

We end this section with a short proof of the Rank-Nullity Theorem.

Proof of the Rank-Nullity Theorem. First, suppose $\text{nullity } A = 0$, in which case $\vec{x} = 0$ is the only solution of $A\vec{x} = 0$. According to our Uniqueness Theorem for Matrices, we can conclude that the n columns of A are linearly independent, so that the column rank of A is n. But we saw in Theorem 5.3 that the column rank of A is equal to the rank of A, allowing us to conclude that $\text{rank } A = n$, so that in this case

$$\text{rank } A + \text{nullity } A = n + 0 = n,$$

as claimed.

Next, suppose $\text{nullity } A = \ell \geq 1$, and let $\{\vec{v}_j\}_{j=1}^{\ell}$ denote a basis for the null space of A. We can extend this to a full basis of $\mathbb{R}^n$, $\{\vec{v}_j\}_{j=1}^{n}$, in which case the collection of vectors $\{A\vec{v}_j\}_{j=1}^{n}$ must span $\mathcal{R}(A)$. But we know that $A\vec{v}_j = 0$ for all $j = 1, 2, \ldots, \ell$, so in fact $\{A\vec{v}_j\}_{j=\ell+1}^{n}$ must span $\mathcal{R}(A)$. In addition, we can check that the set $\{A\vec{v}_j\}_{j=\ell+1}^{n}$ is linearly independent. To this end, suppose we have any linear combination from this set satisfying

$$\sum_{j=\ell+1}^{n} c_j A\vec{v}_j = 0.$$

[41] Here, we need to set aside the fact that we "know" that this is actually the rank of A, because that's effectively what we're proving.

This means $A(\sum_{j=\ell+1}^{n} c_j \vec{v}_j) = 0$ so that the linear combination $\sum_{j=\ell+1}^{n} c_j \vec{v}_j$ is in the null space of A. But this contradicts our assumption that the vectors $\{\vec{v}_j\}_{j=1}^{n}$ form a linearly independent set. We can conclude that the vectors $\{A\vec{v}_j\}_{j=\ell+1}^{n}$ form a basis for $\mathcal{R}(A)$, and so $\text{rank}(A) = n - \ell$. But then

$$\text{rank}\, A + \text{nullity}\, A = (n - \ell) + \ell = n,$$

as claimed. □

5.4 Proof of Buckingham's Theorem

Before proving Buckingham's Theorem in the general case, we'll work through what the proof looks like for Example 5.1a.

5.4.1 Proof of Buckingham's Theorem: Special Case

Let's recall once again that for Example 5.1a, we express our dimensionless products as

$$\pi = v^a g^b h^c t^d,$$

and we have a complete set of dimensionless products

$$\pi_1 = \frac{gh}{v^2}, \quad \pi_2 = \frac{gt}{v}.$$

These correspond respectively with the linearly independent vectors

$$\vec{v}_1 = \begin{pmatrix} -2 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \quad \vec{v}_2 = \begin{pmatrix} -1 \\ 1 \\ 0 \\ 1 \end{pmatrix},$$

which comprise a basis for the null space of the matrix A for this problem (the matrix in (5.5)). Let's also recall that we have the correspondence between multiplicative combinations of the dimensionless products

$$\pi = \pi_1^{\alpha_1} \pi_2^{\alpha_2}, \quad \text{with} \quad \pi = v^a g^b h^c t^d$$

and linear combinations of the associated vector of coefficients

$$\vec{v} = \alpha_1 \vec{v}_1 + \alpha_2 \vec{v}_2, \quad \text{with} \quad \vec{v} = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}.$$

Starting with $\vec{v}_1$ and $\vec{v}_2$, we can complete a basis for $\mathbb{R}^4$ by adding any two additional vectors $\vec{v}_3$ and $\vec{v}_4$ so that the full collection $\{\vec{v}_j\}_{j=1}^{4}$ is linearly independent. For our purposes, a convenient choice will be

$$\vec{v}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \quad \vec{v}_4 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}. \tag{5.11}$$

We can readily check that the vectors $\{\vec{v}_j\}_{j=1}^4$ are linearly independent by verifying that $\det(\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{v}_4) \neq 0$; in this case, we find $\det(\vec{v}_1, \vec{v}_2, \vec{v}_3, \vec{v}_4) = 1$.

Remark 5.1. *According to the Fredholm Alternative, we have the relation*

$$\mathbb{R}^4 = \mathcal{N}(A) \oplus \mathcal{R}(A^T).$$

Since $\vec{v}_1, \vec{v}_2 \in \mathcal{N}(A)$, we could take $\vec{v}_3$ and $\vec{v}_4$ to be any two linearly independent vectors in $\mathcal{R}(A^T)$. As discussed in the proof of Theorem 5.3, the range of A^T is just its column space, which of course includes both its columns, so we can just use those. E.g., in this case we have

$$A = \begin{pmatrix} 1 & 1 & 1 & 0 \\ -1 & -2 & 0 & 1 \end{pmatrix} \implies A^T = \begin{pmatrix} 1 & -1 \\ 1 & -2 \\ 1 & 0 \\ 0 & 1 \end{pmatrix},$$

so we could take

$$\vec{v}_3 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \quad \vec{v}_4 = \begin{pmatrix} -1 \\ -2 \\ 0 \\ 1 \end{pmatrix}.$$

We could also use the RRE form of A to obtain yet another viable pair. Nonetheless, we'll stick with (5.11), because this choice clarifies an important part of the proof.

These new vectors $\vec{v}_3$ and $\vec{v}_4$ (from (5.11)) correspond with *dimensioned products* $\pi_3 = h$ and $\pi_4 = t$. Since $\{\vec{v}_j\}_{j=1}^4$ is a basis for $\mathbb{R}^4$, we can express any vector in $\mathbb{R}^4$ as a linear combination of these vectors. Correspondingly, we can express any multiplicative combination of the variables in our problem as multiplicative combinations of the dimensionless and dimensioned products $\{\pi_j\}_{j=1}^4$. In particular, we can express each individual variable v, g, h, and t as a multiplicative combination of the products $\{\pi_j\}_{j=1}^4$. This guarantees that we can express the equation we're looking for as

$$\psi(\pi_1, \pi_2, \pi_3, \pi_4) = 0 \tag{5.12}$$

for some function ψ, because we can simply replace each variable with precisely the multiplicative combination of the products $\{\pi_j\}_{j=1}^4$ to which it is equal. Of course, what Buckingham's Theorem claims is that we can express our equation entirely in terms of π_1 and π_2, so we still have some work to do.

At this point, let's be careful to observe that the relation (5.12) describes the relationship between the four quantities $\{\pi_j\}_{j=1}^4$, but the function ψ itself is not identically zero. To be clear about this, let's recall that for this example we know that we can use $f(\pi_1, \pi_2) = -\frac{1}{2}\pi_2^2 + \pi_2 + \pi_1$ to express the equation we're searching for as

$$f(\pi_1, \pi_2) = -\frac{1}{2}\pi_2^2 + \pi_2 + \pi_1 = 0.$$

But of course the function f itself is not identically zero; e.g., $f(1, 1) = \frac{3}{2}$. It's just that $(\pi_1, \pi_2) = (1, 1)$ is not a solution. In particular, if we vary either π_1 or π_2, we must also vary the other to ensure that $f(\pi_1, \pi_2) = 0$ continues to hold.

Returning to ψ, for the specific choices of π_3, and π_4 in the current case, we have

$$\psi(\pi_1, \pi_2, h, t) = 0. \tag{5.13}$$

Here's the key: we claim that ψ is actually independent of h and t. To see this, let's recall that under any change of units the values π_1 and π_2 will remain fixed. On the contrary, by changing our unit of length we can change the value of h, and by changing our unit of time we can change the value of t. Focusing on h, this means that by changing the unit of length we can vary h without varying any of the other variables that ψ depends on. (We emphasize here the contrast with our discussion of f in the previous paragraph.). Since each summand in ψ is dimensionless, even if ψ is not 0, its value will not vary as the units change, and so it will not vary in h; i.e., it will be constant in h as claimed.[42] Exactly the same reasoning can be applied in the case of t, and we can conclude that ψ does not depend on t either. It follows that the function guaranteed to exist by Buckingham's Theorem can be taken to be

$$f(\pi_1, \pi_2) = \psi(\pi_1, \pi_2, h, t).$$

5.4.2 Proof of Buckingham's Theorem: General Case

For the general proof of Buckingham's Theorem, we will let d denote the number of dimensions involved (i.e., L, M, T etc.), and we will denote by p the number of physical variables involved (i.e., v, g, h etc.) We let $\{\pi_j\}_{j=1}^n$ denote the complete set of dimensionless products assumed in the theorem statement, and we let $\{\vec{v}_j\}_{j=1}^n$ denote the associated vectors of exponents.

In this context, the matrix A that we obtain from our usual dimensions equations will be size $d \times p$, corresponding with one row for each dimension and one column for each physical variable. We know that nullity $A = n$ (the number of dimensionless products), so by the Rank-Nullity Theorem

$$\operatorname{rank} A = p - n.$$

First, let's consider the special case $p = n$ for which $\operatorname{rank} A = 0$, implying that A is identically 0 (i.e., 0 in every entry). This can only happen if every variable in the equation is dimensionless (and so a possible dimensionless product), but if every variable in the equation is dimensionless, then each can be written as a multiplicative combination of the $\{\pi_j\}_{j=1}^n$ (by definition of a complete set of dimensionless products). If we can write each variable in terms of the $\{\pi_j\}_{j=1}^n$, then we can certainly write the full equation in terms of this set, from which we can conclude that the theorem holds for $p = n$.

If $n < p$, we complete a basis for $\mathbb{R}^p$, writing the full basis as $\{\vec{v}_j\}_{j=1}^p$, where the additional vectors $\{\vec{v}_j\}_{j=n+1}^p$ are not in the null space of A, and so correspond with dimensioned products $\{\pi_j\}_{j=n+1}^p$. Next, since every element of $\mathbb{R}^p$ can be expressed as a linear combination of the vectors $\{\vec{v}_j\}_{j=1}^p$, it must be the case that every variable in the problem can be written as a multiplicative combination of the dimensionless and dimensioned products $\{\pi_j\}_{j=1}^p$. This means that we can certainly express the equation we're searching for as

$$\psi(\pi_1, \pi_2, \ldots, \pi_p) = 0$$

[42] We note that our assumption that the equation is in dimensionless form rules out a case such as $\psi(\pi_1, \pi_2, h, t) = h(-\frac{1}{2}\pi_2^2 + \pi_2 + \pi_1)$, which is a perfectly valid form of ψ, except with every term having dimension length. Of course, we just divide by h to get the assumed dimensionless form.

for some function ψ. At this point, we make an important claim regarding an issue that got strategically swept under the rug in our special case.

Claim 5.1. *The dimensioned products $\{\pi_j\}_{j=n+1}^p$ can be chosen so that each contains at least one dimension that does not appear in any of the others.*

Let's first check that this claim allows us to finish the proof, and then we'll go back at the end and prove it. If the claim is true, then by changing units selectively, we can vary the value of each of the $\{\pi_j\}_{j=n+1}^p$ without varying the value of any other product, dimensionless or dimensioned. This means that ψ must be independent of all of the dimensioned products, and we can set

$$f(\pi_1, \pi_2, \ldots, \pi_n) = \psi(\pi_1, \pi_2, \ldots, \pi_p).$$

This completes the proof, except for the verification of Claim 5.1, which we do now. As a start, we observe that for each dimensioned product π_j, we can associate a new vector $\vec{u}_j \in \mathbb{R}^d$ whose components are the dimension exponents of π_j. For example, if the only dimensions are L, M, and T, and $\pi_j = v$ (i.e., velocity) then $\vec{u}_j = (1, 0, -1)$. Now, the vectors $\{\vec{u}_j\}_{j=n+1}^p$ must be linearly independent, because otherwise we could find a non-trivial linear combination of them that would be 0 (i.e., $\sum_{j=n+1}^p \alpha_j \vec{u}_j = 0$), and this would necessarily correspond with a multiplicative combination of the $\{\pi_j\}_{j=n+1}^p$ with no dimension (i.e., $\Pi_{j=n+1}^p \pi_j^{\alpha_j}$), which is a contradiction.

At this point, let's arrange the vectors $\{\vec{u}_j\}_{j=n+1}^p$ as the rows of a $(p-n) \times d$ matrix

$$\begin{pmatrix} — \ \vec{u}_{n+1} \ — \\ — \ \vec{u}_{n+2} \ — \\ — \ \vdots \ — \\ — \ \vec{u}_p \ — \end{pmatrix},$$

so that each row corresponds with a dimensioned product and each column corresponds with a dimension. In particular, a dimension appears in a dimensioned product if and only if the entry where the row corresponding with the dimensioned product and the column corresponding with the dimension intersect has a non-zero entry. We now think of performing row operations on this matrix, which correspond with linear combinations of the vectors $\{\vec{u}_j\}_{j=n+1}^p$, which in turn correspond with multiplicative combinations of the associated dimensioned products $\{\pi_j\}_{j=n+1}^p$. If we put this matrix in RRE form, we ensure that each row (corresponding with each dimensioned product) contains at least one non-zero entry in a column (corresponding with a dimension) that is 0 in all other rows. This new set of rows corresponds with an alternative choice of dimensioned products for which each dimensioned product includes at least one dimension that is not included in the others. This completes the proof of the claim, which completes the proof of Buckingham's Theorem. □

5.5 Non-dimensionalizing Equations

It's often useful to non-dimensionalize an ODE or PDE before studying its qualitative properties (e.g., whether or not it has periodic solutions, the long-time behavior of solutions, chaotic behavior etc.) This process produces a simpler form of the equation and identifies dimensionless combinations of parameters that determine qualitative behavior.

Example 5.8. The equation for the height $y(t)$ of an object falling from far above the earth's surface is

$$y'' = -\frac{GM}{(R+y)^2} - b|y'|y',$$

where G denotes Newton's gravitational constant, M denotes the mass of the earth, R denotes the radius of the earch, and $b > 0$ is a coefficient of air resistance. Here, $[y''] = LT^{-2}$, so each summand in the equation must have the dimensions of acceleration. We can use this observation to determine the dimensions of G and b as

$$[G] = \frac{[y''][(R+y)^2]}{[M]} = \frac{(LT^{-2})L^2}{M} = L^3M^{-1}T^{-2}$$
$$[b] = \frac{[y'']}{[y']^2} = \frac{LT^{-2}}{L^2T^{-2}} = L^{-1}.$$

The idea of non-dimensionalization is to replace each independent and dependent variable in the equation with a corresponding dimensionless variable. For this, we'll set

$$\tau = \frac{t}{A}, \quad Y(\tau) = \frac{y(t)}{B},$$

where A will be chosen as a constant with dimension T (so that τ is dimensionless) and B will be chosen as a constant with dimension L (so that Y will be dimensionless). Before making these choices, we observe that

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

and likewise $y''(t) = \frac{B}{A^2}Y''(\tau)$. This allows us to express our original equation in the form

$$\frac{B}{A^2}Y'' = -\frac{GM}{(R+BY)^2} - b|\frac{B}{A}Y'|\frac{B}{A}Y'.$$

We will take A and B to be positive constants, so we can express this as

$$Y'' = -\frac{GM}{(R+BY)^2}\frac{A^2}{B} - bB|Y'|Y'.$$

It's natural to choose $B = b^{-1}$ (which has the right dimensions), and in this case the first summand on the right-hand side becomes

$$-\frac{GM}{(R+\frac{1}{b}Y)^2}bA^2 = -\frac{GMb}{(1+\frac{1}{bR}Y)^2}\frac{A^2}{R^2}.$$

To simply things, let's choose A so that

$$\frac{GMbA^2}{R^2} = 1.$$

I.e., we take

$$A = \frac{R}{\sqrt{GMb}}.$$

Just to be sure we have our dimensions right, we can check

$$[A] = \frac{[R]}{[\sqrt{GMb}]} = \frac{L}{L^{3/2}M^{-1/2}T^{-1}M^{1/2}L^{-1/2}} = T,$$

as expected. We can now express our equation in the non-dimensionalized form

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

This suggest that the qualitative behavior of our projectile is primarily determined by the (often small) parameter $\epsilon := \frac{1}{bR}$. △

Example 5.9. Consider the Lotka–Volterra system for a population of prey $y_1(t)$ and a population of predators $y_2(t)$,

$$\begin{aligned}\frac{dy_1}{dt} &= ay_1 - by_1y_2\\ \frac{dy_2}{dt} &= -ry_2 + cy_1y_2,\end{aligned}$$

where a, b, c, and r are taken to be positive constants. Here, $[y_1] = [y_2] = B$ (i.e., biomass), and we can check that $[a] = [r] = T^{-1}$ and $[b] = [c] = B^{-1}T^{-1}$. We set

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

where we're skipping B as a constant since it's the symbol for one of our dimensions. Similarly as with the previous example, we find that $y_1' = \frac{C}{A}Y_1'$ and $y_2' = \frac{D}{A}Y_2'$, so that the Lotka-Volterra system can be expressed as

$$\begin{aligned}\frac{C}{A}Y_1' &= aCY_1 - bCDY_1Y_2\\ \frac{D}{A}Y_2' &= -rDY_2 + cCDY_1Y_2.\end{aligned}$$

Upon multiplying the first of these equations by $\frac{A}{C}$ and the second by $\frac{A}{D}$, we arrive at the system

$$\begin{aligned}Y_1' &= aAY_1 - bADY_1Y_2\\ Y_2' &= -rAY_2 + cACY_1Y_2.\end{aligned}$$

We make the choices $A = \frac{1}{a}$, $C = \frac{a}{c}$, and $D = \frac{a}{b}$, to arrive at the system

$$\begin{aligned}Y_1' &= Y_1 - Y_1Y_2\\ Y_2' &= -\frac{r}{a}Y_2 + Y_1Y_2.\end{aligned}$$

In this case, we see that the qualitative behavior is primarily determined by the single parameter $\delta = \frac{r}{a}$. △

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 in a neighborhood of $\vec{p}$, and its derivatives with respect to each variable are also continuous in a neighborhood of $\vec{p}$). Suppose additionally that*

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

and

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

Then there exists a neighborhood N_p of $(p_1, p_2, ..., p_{n-1})$ and a function $\phi : N_p \to \mathbb{R}$ so that

$$p_n = \phi(p_1, p_2, ..., p_{n-1}),$$

and for every $x \in N_p$,

$$f(x_1, x_2, ..., x_{n-1}, \phi(x_1, x_2, ..., x_{n-1})) = 0.$$

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 polynomial 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*

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

References

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

[2] J. Ellenberg, *How Not to Be Wrong: The Power of Mathematical Thinking*, Penguin Press 2014.

[3] F. Galton, *Regression toward mediocrity in hereditary stature*, Anthropological miscellanea, 1886, 246-263.

[4] J. G. Kalbfleisch, *Probability and Statistical Inference, Volume 2: Statistical Inference*, Springer-Verlag 1985.

[5] T. O. Kvålseth, *Cautionary Note about R^2*, The American Statistician **39** (1985) 279-285.

[6] S. J. Leon, Linear algebra with applications, 9th Ed., Pearson 2015.

[7] G.I. Taylor, *The formation of a blast wave by a very intense explostion II: the atomic explosion of 1945*, Proceedings of the Royal Society of London, Series A, vol. 201 (1950), no. 1065, pp. 175-186.

[8] G.I. Taylor, *The formation of a blast wave by a very intense explostion I: theoretical discussion*, Proceedings of the Royal Society of London, Series A, vol. 201 (1950), no. 1065, pp. 159-174.

Index