ideas of societal development raised by theorists such as Nisbet, Rostow, Organski, Ingelhart, and Pye. Graph algebra need not be limited to those realms in which one wants to resolve a parameter's value. Indeed, graph algebra can be used narrowly to specify a model for a specific question or problem or broadly to theorize within intellectual realms of significant expanse, or anywhere in between.

2. GRAPH ALGEBRA BASICS

The use of graph algebra can yield marked benefits to theory building in the social sciences, and it is useful to view these benefits when considering the linear regression model. Arguably, the most common model used in the social sciences is the linear regression model. While many approaches to parameter estimation exist for linear models, the ultimate result is typically a table with a list of independent variables and their associated parameter estimates and standard errors. From this perspective, the list of variables in the table *is* the model. Specification concerns usually revolve around the question of whether or not a researcher has omitted one or more important variables from the analysis, although sometimes the issue of functional form also is involved.

While graph algebra does not reduce a researcher's need to be aware of potential omitted variable specification problems, it does allow the researcher much greater flexibility with respect to designing innovative and intellectually appealing functional forms. As an absolute minimum, graph algebra allows us to develop more sophisticated model specifications such that the algebraic form of the model becomes as important as the variables that exist within that form. Thus, systems theory as it is expressed through graph algebra offers a means of developing algebraic formulations that correspond with social and political theories that are more complex and sophisticated than the ubiquitous linear form. Thus, as a movement away from the linear model, the use of graph algebra encourages the development of increasingly interesting scientific theories. Moreover, as will become clear by the end of this book, such theories find their origin in the thinking of the theorist, not in the graph algebra itself.

A researcher gains the benefits of graph algebra by mastering its functionality as a language. Graph algebra is the language that we use to describe a system's structure and functioning. With graph algebra we identify the parts of the system's structure, and then we connect those parts in a process that identifies the structure's functioning. Thus, the system's structure leads to an understanding of its functioning. Once we have identified a system's structure and functioning using graph algebra, we then turn to an analysis of the response of the system with respect to variations in inputs. Ultimately, we use graph algebra to describe a social process and to identify causal inference with a system's perspective. Most graph algebra models can be fully estimated with respect to a body of data, and researchers will in general want to do this. However, one can also use graph algebra to develop models that are used only analytically. Subsequent analysis of the model (estimated or otherwise) can take many forms, including prediction or forecasting, various analytics, and simulation.

In terms of mechanics, graph algebra uses elements to transform inputs into outputs. The elements are the parts of the system's structure that go between the inputs and the outputs. The elements contain operators that describe how those elements work to transform one state of the system into another state of the system. One can think of a ''state of the system'' as a measuring point during the process of transforming inputs into outputs. As inputs are changed into outputs, they experience various intermediate conditions. These intermediate conditions are the states of the system. An element works on a state of the system to transform it into another state of the system. All of this happens before the entire system eventually gives birth to a final output.

In graph algebra, elements are normally represented by boxes. Operators go inside the boxes. The collection of connected elements constitutes the system's structure. Typically, inputs go to the left of this structure, and outputs are placed to the right of this structure.

Inputs, Outputs, and the Forward Path

All systems require inputs and outputs. In many situations often associated with single-equation models, the inputs are typically considered the independent variables of the model, whereas the output is the dependent variable of the model. I write ''in many situations'' because it is possible for independent variables not to be explicit inputs, such as with the variable *time* in many of the continuous-time dynamic models presented later in this book. Also, sometimes inputs are not really ''independent,'' as in the multiple regression sense of the word. Indeed, the idea that one variable is ''independent'' and another variable is ''dependent'' normally implies that the independent variable causes change in the dependent variable and nothing of significant importance is causing change to the independent variable. That is, the causality is one-way only, from independent to dependent. But in many interesting systems, the idea that a variable is truly independent can be a misnomer. Indeed, the essence of systems analysis in the first place is that everything affects everything else, or a least some other things. So it is possible for systems to exist that do not really have any authentically independent variables. For this reason, we normally use the word *input* rather than *independent variable* when talking about systems. But this choice of terminology will vary depending on a particular theoretical context and a researcher's preferences.

There is a similar issue with the use of the words *dependent variable* to describe a system's output. We can use these words so long as we recognize that their use may not be entirely parallel to the way they are used in statistical analyses, such as with multiple regression, since dependency can reside in many places within a system. The more common usage when talking about systems is to call the dependent variable the output. Again, this choice of terminology will depend on the theoretical context and a researcher's preferences. Also, multiple-equation models may have more than one output or dependent variable. More intuitively, the inputs are what go into the system to make it work, and the outputs are the result of the system's processing of the inputs. In mechanical terms, one can think of gasoline as an input, the engine as the system, and forward movement of the car as the output.

It is best to describe graph algebra with a simple heuristic example. Using an example drawn from human behavior, let us say that workers in a political campaign are doing door-to-door canvassing for potential voters to support their candidate. Let us also say that a certain proportion of these interactions result in a successful mobilization of voters. This is a simple system, and for the purpose of keeping this example system simple, so that we can initially focus only on the graph algebra that describes its structure, I am purposely avoiding ideas that might make this system more realistic and thus more complicated. (For example, one might ask if some of those contacted by the campaign workers might have voted regardless of whether or not the workers knocked on their door.) The input to this system is the canvassing activity done by the campaign workers, and the output to this system is the mobilization of voters. This is represented using graph algebra in Figure 2.1.

In Figure 2.1, C_t is the input to the system, and it represents the number of people who are contacted by the campaign workers during the canvassing activity at time *t.* The box in the figure is an element of the system, and it contains the parameter *p.* Parameters are one form of operator, and they are called "parameters of proportional transformation" in the language of graph algebra. What this means in terms of the model in Figure 2.1 is that a certain proportion (*p*) of the people canvassed by the campaign workers [C_t] will become mobilized to vote. The output of the system is V_t, and this

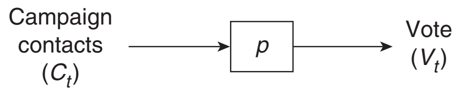

Figure 2.1 A Simple System of Voter Mobilization Using Graph Algebra

represents the number of people who are mobilized to vote. The path from C_t to V_t is called the "forward path," since the activity of the system moves "forward" from input to output along this path. The convention is that forward paths typically flow from left to right.

Graph algebra always translates into an algebraic statement. In the case of Figure 2.1, the algebraic statement for this graph algebra model is $V_t = pC_t$. This results from the most basic rule of graph algebra:

> **Graph Algebra Rule #1:** *Things that flow along the same path are multiplied.*

Thus, we begin with the input and multiply it by anything that is located along the forward path. We then set that equal to the output of the system. This simple model says that some proportion (p) of the campaign contacts [C_t] is transformed into voters [V_t], which means that we multiply p by C_t to obtain V_t.

It is now easy to use graph algebra to represent a simple linear regression equation of the sort commonly used in many statistical analyses. This will also allow us to introduce the second most basic rule of graph algebra. This linear regression model is presented in Figure 2.2 with a model having four independent variables, an intercept, and one dependent variable. The error term is omitted here for simplicity.

In Figure 2.2, the graph algebra model describes a person's probability of voting as a function of the person's income, level of education, self-described partisan identification, and a contextual measure of the partisan composition of the neighborhood within which he or she lives. Note that there are five forward paths in this figure. Four of the forward paths are a result of the inputs from the four independent variables described above, and I describe the fifth forward path below. The algebraic equation that results from the graph in Figure 2.2 is shown as Equation 2.1:

$$\text{Vote} = \beta_0 + \beta_1(\text{Income}) + \beta_2(\text{Education}) + \beta_3(\text{PID}) + \beta_4(\text{Context}), \quad [2.1]$$

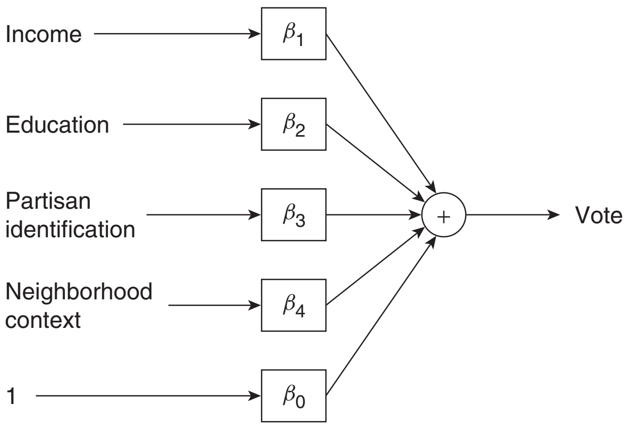

Figure 2.2 A Simple Regression Model With Four Independent Variables, an Intercept, and One Dependent Variable

where PID stands for partisan identification and context stands for neighborhood context. Note that all the forward paths in Figure 2.2 get added together in Equation 2.1. This introduces the second most basic rule of graph algebra:

> **Graph Algebra Rule #2:** *Add paths that merge together at an intersection.*

Note also that in Figure 2.2 there is an additional forward path that contains an input of 1 and the parameter β_0. This reflects how many computer programs calculate an intercept for an estimated regression line, in the sense that a column of 1s is typically added to a data matrix as a new variable. Thus, in computational terms, an intercept is actually nothing more than a slope times the "variable" 1, which in practice simply leaves us with an additive constant. Finally, note that the model in Figure 2.2 is static, in the sense that time plays no role in structuring the relationship between these variables. This is simply a model that specifies how the values of the four independent variables, which reflect qualities of selected individuals, affect the value of the dependent variable. In this instance, there is no ambiguity in using the terms *independent variables* and *dependent variable* instead of *inputs* and *output,* respectively, since graph algebra is being used to describe a multiple regression setting in which the former terms are totally appropriate and the issue of causality is clear. Also, readers should note that the model is purposely simple, and to keep it useful as a heuristic vehicle to

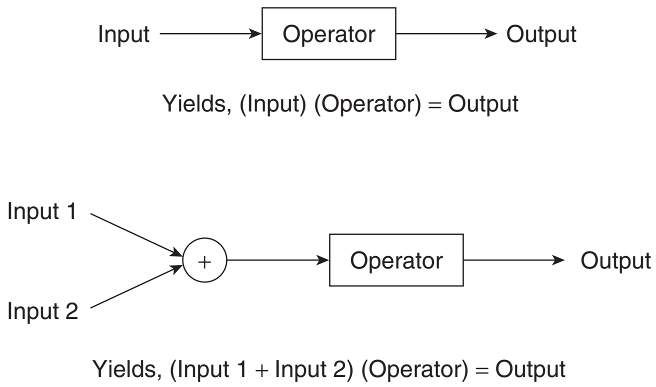

Figure 2.3 The General Application of Rules #1 and #2 of Graph Algebra

demonstrate the mechanics of graph algebra, other factors that might make it more realistic (and thus more complicated) are avoided.

In general, the application of the first two rules of graph algebra is summarized in Figure 2.3. Note the use of the term *operator* in the element boxes in Figure 2.3. A parameter of proportional transformation is one type of operator that can be used in an element. Other operators are discussed later in this book. Note also that in the example given in the lower part of Figure 2.3 two separate inputs are added together before both of them are "sent" through the same operator on the forward path. These are just heuristic examples of a limitless arrangement of inputs, operators, and outputs. Also, in situations in which there are two inputs, both the inputs do not have to enter the model on the far left. It is possible to enter an input or to place an operator nearly anywhere in graph algebra as long as it all makes sense from a social theory perspective.

Feedback Loops and Mason's Rule

Regulation and control are primary strengths of modeling using graph algebra. Feedback loops are typically used to accomplish regulation and control. A feedback loop is like an input, but its origin is from within the system itself, not from outside of the system. In many systems, the output reenters the system as another input. As mentioned previously, this is exactly what happens with a microphone and speakers when the sound from the speakers feeds back into the microphone, often causing a loud squeal.

Let us return to the earlier example in which campaign workers were canvassing for potential voters. After campaign workers talk to potential voters, some of these people will be mobilized to support the candidate or issue that is discussed. Some of these mobilized people may subsequently begin to talk to their neighbors, friends, and coworkers trying to convince them also to support the cause. Thus, the newly mobilized people are reentering the vote-mobilization system as a new input. They are not the original campaign workers, so they cannot be included as part of that original input. Rather, they are separate. But they too will interact with people just as do the original campaign workers, which means that a proportion of these new interactions will result in additional support for the campaign. This is the same as it was with the original campaign workers.

The general depiction of a feedback loop using graph algebra is shown in Figure 2.4. This figure demonstrates a positive feedback loop, in the sense that the output of the system feeds back into the system as a positive input. The two examples described above, one involving feedback with a microphone and speakers and the other involving campaign interactions, are both instances of positive feedback. But the microphone example is dynamic, in the sense that the squeal from the speakers (and thus the degree of feedback) changes over time. However, note that in Figure 2.4 there is no indication of how time would be involved in the process, since neither the input nor the output is subscripted with *t*. This kind of system is called a "static system."

Static systems are conceptually different from a "simultaneous system," which is described below, since a static system is independent of time entirely. Linear regression models that relate one dependent variable to a list of independent variables are common examples of static systems as long as none of the variables are subscripted by time. For example, relating income to education via a correlation analysis is a static comparison, as opposed to relating income to education at a particular point in time, which assumes that the relation could be different at a different point in time. This distinction will become more important to us later when we begin using time-based operators with graph algebra.

One might naturally ask how a campaign feedback process can be static (or simultaneous) since it must take place over time. This depends on how we conceptualize the feedback process. Later in this book, time operators are introduced that allow one to specify exactly when the feedback process occurs relative to the other parts of the system. But such time operators do not appear in Figure 2.4. With feedback loops, a fraction of the output reenters the system to eventually show up again as an output, and a fraction of that output reenters the system yet again through the feedback loop . . . and

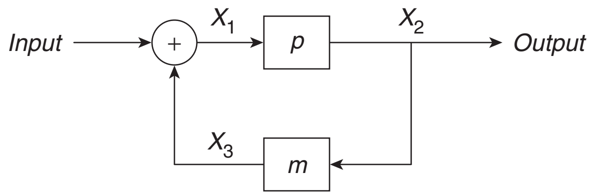

Figure 2.4 A Positive Feedback Loop

on and on it goes. With static systems, the feedback parameter (in this case *m*) represents the summation of these ongoing feedback cycles, as if the process is in equilibrium, or perhaps at the conclusion of a conceptually bounded time span such as an election campaign.

It is also important to note that when an output reenters the system through a feedback loop, the reentry does not operate as a subtraction from the output (thus diminishing the output). Thus, a feedback process does not remove the output to reuse it. For example, the microphone does not take sound away from the speakers when it reenters some of the output back into the system's amplification process. The output is still the output; it is simply fed back into the system.

Note the variables X_1, X_2, and X_3 in Figure 2.4. These are states of the system, which are values of the system at various points within the system. With graph algebra, one never leaves the states of the system in the final algebra of the model. The states of the system are used only as algebraic conveniences to help us determine the model's final form. For example, in this system we have

$$X_1 = Input + X_3,$$
$$X_2 = pX_1,$$
$$X_3 = mX_2.$$

Note that X_2 also equals the output of the system. We can substitute and eliminate the states of the system, thereby restating the model:

$$X_2 = p(Input + X_3)$$
$$X_2 = p(Input + mX_2)$$

And since $X_2 = Output,$ we have

$$Output = p(Input + mOutput),$$

or, after rearranging,

$$Output = Input[p/(1 - pm)]. \qquad [2.2]$$

This derives Mason's Rule, named after its author (see Cortés et al., 1974, p. 104). Mason's Rule is a shortcut for finding the function of a single feedback loop. In words, Mason's Rule can be stated as the forward path divided by the quantity 1 minus the product of the forward path and the feedback path. Restated, Mason's Rule for determining the function of a single feedback loop is as follows:

Mason's Rule: *Forward path*/[1 − (*Forward path*)(*Feedback path*)].

This formula gets multiplied by the system's input to equal the output. The states of the system can always be used to determine the algebraic equation for any graph algebra representation. But sometimes Mason's Rule is quite handy and is introduced here as a convenience for graph algebra that a modeler may or may not wish to use.

An Example From Economics: The Keynesian Multiplier

The Keynesian multiplier is a good example from economics of the use of a positive feedback loop. Consider a system in which a nation's total income is a function of investment and consumption. Consumption acts as an additional input into the system since consumers spend their money on products, thereby reinserting their income into the functioning of the economy. Using graph algebra, we can depict such a simple economy, as in Figure 2.5.

Note that the forward path of Figure 2.5 has an element that contains the number 1. This is an "invariant transformation," which is simply an identity operation. In this model, all investments are transformed into economic output, in the sense that nothing is lost. Note also that investment and economic output are subscripted with respect to time and that the time subscript is the same for both. This means that the system is a time-dependent *simultaneous system,* in the sense that all investments are immediately counted as (or transformed into) economic output. The equation that is produced with this graph algebra representation is shown as Equation 2.3, and this can easily be obtained by applying Mason's Rule to Figure 2.5:

$$Economic\ output_t = Investment_t[1/(1 - c)], \qquad [2.3]$$

where $1/(1 - c)$ is the familiar Keynesian multiplier.

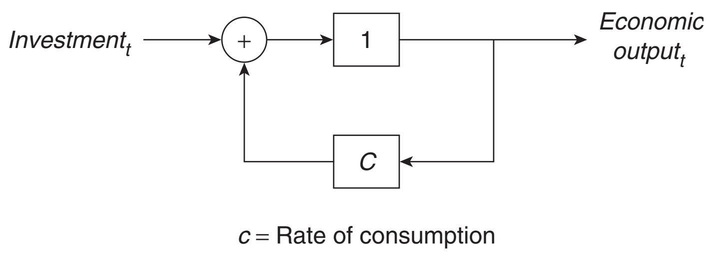

Figure 2.5 The Keynesian Multiplier: Economic Output as a Function of Investment and Consumption

Equation 2.3 would normally be estimated using the statistical form of Equation 2.4, where the parameter β_0 is an intercept for the statistical model and parameter β_1 is the slope:

$$Economic\ output_t = \beta_0 + \beta_1(Investment_t) \quad [2.4]$$

Note that an intercept could have been added to the graph algebra diagram in Figure 2.5 just as it is included in Figure 2.2. In this book, intercepts are often omitted from the graph algebra diagrams to obtain a more tidy presentation, and it is left to the reader to reinsert those intercepts in cases where desired. From Equation 2.4 we can see that $\beta_1 = 1/(1 - c)$. Thus, if we estimate Equation 2.4 and obtain the parameter β_1, then we also need to calculate the value of the parameter c, which is embedded inside β_1.

This simple example helps to emphasize a useful feature of graph algebra. Our real interest is not in finding the value of β_1, but rather the value of c. It is the graph algebra that helps us see this. If we began with the statistical model shown as Equation 2.4 (i.e., in the absence of the graph algebra representation of the role of consumption), then we might not realize that the relationship between investment and economic output is complicated by the feedback component of consumption. Thus, graph algebra assists in clarifying the specification of many such models in which the parameters of interest are embedded inside the estimated statistical parameters. In this case, the Keynesian multiplier is so well understood that one might say that the graph algebraic representation of the model is not needed. While this may be the case in this instance, there will be many other models in which the specifications are more complex and not well-known, and it is in those situations that the use of graph algebra is particularly valuable.

This example is useful in again pointing out an aspect of graph algebra that can sometimes be misunderstood. In Figure 2.5, note that an arrow leaves *Economic output*$_t$ and flows through the parameter c before reentering the system at the front end of the feedback loop where the circle with the plus sign is located. This does not mean that something is being subtracted from the value of *Economic output*$_t$ for it to be reentered at the plus sign. Thus, the arrow leaving *Economic output*$_t$ and pointing to the parameter c is not reducing the value of *Economic output*$_t$. Rather, a measure of *Economic output*$_t$ is being taken where the feedback loop begins, and some proportion (c) of this is being reinvested in the economy. Again, restated differently, the beginning of a feedback loop does not "pull" something out of the forward path. It merely takes a measure of the value of the forward path at that point in the system so that part of that measure can be reentered elsewhere in the system.

Also note that the parameter c instantaneously summarizes a set of diminishing feedback cycles, as is the case with all static and simultaneous systems. This is the same as was described previously with respect to Figure 2.4 and parameter m. In Chapter 3, we learn how to structure the feedback process using time operators, thereby keeping track of when an output actually feeds back into the system relative to other parts of the system.

3. GRAPH ALGEBRA AND DISCRETE-TIME LINEAR OPERATORS

So far time has not played a significant role in our discussions. Structuring the relationships between the variables with respect to time within the context of a system is one of the great strengths of graph algebra. All the models that are discussed throughout the remainder of this book use graph algebra to do this. This discussion begins with explaining how graph algebra is integrated with discrete-time applications. Discrete time implies the use of difference equations, and difference equations are often appropriate for the social sciences since a great deal of social scientific data are collected in discrete intervals. Examples of this are census data, economic data, election data, and polling data (which often correspond with an electoral calendar). Differential equations are used to model continuous-time processes and are discussed later. Models can also be built using graph algebra that have both continuous and discrete parts. These are called "metered" differential equations, and they are also discussed later in this book in the context of differential equations with embedded time lags.

All the operators used in this book are linear operators (see especially Allen 1963, p. 725; see also Goldberg, 1958). This is true of the discrete-time operators as well as the continuous-time operators. What do we mean by