Int. J. General Systems
1978, Vol. 4, pp. 163–177

DISCRETE AND CONTINUOUS MODELS

ANDREW G. BARTO
School of Advanced Technology, State University of New York, Binghamton, New York, U.S.A.

(*Received November* 30, 1976; *in final form April* 25, 1977)

The growing influence of digital computing in systems modelling and simulation is leading to an increase in the use of discrete mathematical structures for describing models. While it is generally recognized that discrete methods and classical continuous methods both provide valuable tools for modelling, strong biases exist which depend on the modelling techniques that are traditional within specific disciplines. The choice of a modelling approach sometimes reflects the background of the model builder more strongly than it reflects the character of the problem to be solved. Since continuous methods have played the dominant role in scientific education, there are aspects of discrete modelling techniques and their relationship to continuous methods that are not widely recognized. The purpose of this article is to discuss some of these issues in order to dispel common criticisms of discrete techniques which are the result of unfamiliarity with discrete styles of mathematical thinking and a tendency to underestimate the degree of abstraction used in continuous approaches.

INDEX TERMS Modelling, simulation, modelling traditions, discrete models, numerical analysis.

INTRODUCTION

M. E. Van Valkenburg writes in the foreword to Steiglitz's *Introduction to Discrete Systems*[1], that "Given the widespread availability of computers, there seems little doubt that the teaching of electrical engineering should undergo an evolution ... In the emerging pedagogical approach, equations should be written in discrete form as difference equations, instead of in continuous form as differential equations. Indeed, equations should seldom be used, since principles should be stated directly in algorithmic form." Approaches having this character are increasingly being proposed not only in electrical engineering but in other fields where continuous mathematical methods have been traditional, and not only are pedagogical changes being suggested. Digital computing and discrete models are influencing our conception of real world systems and the role classical mathematical methods are to play in modelling them.

Discussions of this emerging approach occur frequently and continue to reveal strong biases which correspond to the technical backgrounds of the participants. Those whose experience is in the classical physical sciences or in traditional engineering disciplines where differential equations have been so remarkably successful, understandably have strongly developed intuitions in which continua and rates of change are powerful conceptual primitives. Those whose intuition has developed more directly under the influence of digital computing, on the other hand, find it very natural to think in terms of such concepts as algorithms, data structures, and automata. In many applications of these discrete concepts the availability of theoretical results is replaced by the computational power of digital computers.

Due to the relative isolation of these methodological traditions, both historically and in the educational process, there are aspects of continuous and discrete modelling techniques, and aspects of the relationship between these techniques, that are not generally recognized. Although the subject of numerical analysis focuses on the relationship between continuous and discrete methods, the perspective it provides lies thoroughly within the continuous tradition. While intimately concerned with digital computation, numerical analysis concentrates on going from a continuous model, e.g. a differential equation, to some discrete means of deducing that model's behavior. The issues that are important in numerical analysis are substantially different from those which arise from attempts to model *directly* in discrete form, bypassing continuous formulations entirely. In this direct approach the emphasis is on the relationship between some perceived real system and a discrete model, rather than between a continuous model and a discrete approximation of it. This approach has been convincingly put forward by Donald Greenspan and his associates in a series of articles[2,3,4,5] in which discrete models are proposed for the mechanical systems which physicists have classically modelled using differential equations. Not only are the discrete models easily simulated by computer, but they also preserve some of the theoretical attractiveness of the classical models.

Although Greenspan's efforts have primarily been toward formulating (and simulating) discrete models of systems which are usually modelled by continuous methods, he points out, as do Zeigler and Barto[6], that computational methods permit the exploration of entirely new classes of models.

The purpose of this article is to examine some of these issues from the point of view provided by system theory. One of the goals of the system sciences, as stated by Sutherland[7], is to permit the properties of a particular problem rather than *a priori* methodological biases, to determine the analytical approach to be used, and going a step further, to prevent such biases from determining which problems are considered for analysis. These goals seem especially appropriate to the use of discrete and continuous models. Which approach is taken is usually determined by the background of a researcher or by the tradition prevalent within a discipline, rather than by which set of conceptual tools can provide the most expressive means of formalizing a model. Although the modelling formalism which most often comes to mind is the differential equation, system theory provides a framework for rigorously defining a much more general class of dynamical models. Since continuous methods have traditionally played the more visible role in mathematics, some of the material to follow, being well known in some quarters, is tutorial in nature. We hope our discussion dispels some of the apparent misapprehensions about the potential of discrete modelling techniques so that future discussions can focus on substantive criticisms of discrete modelling as it is now practiced.

The terms discrete and continuous apply to a wide range of structures and techniques, but for the purposes of this article they will refer to discrete and continuous representations of time as used in models of processes whose behavior unfolds over time. Models of temporal processes take the form of discrete-time or continuous-time dynamical systems. By the term discrete model, then, we shall mean a model formulated as a dynamical system having a countable time base (usually the integers). This kind of model might be formalized as a set of difference equations or as an automaton. The term continuous model will refer to a system whose time base is the uncountably infinite set of real numbers. Continuous models are usually, although not exclusively, formalized as differential equations. According to this terminology a discrete model can still involve uncountable sets, such as the real or complex numbers, as ranges of some or all of its descriptive variables. In other words, a discrete model (as we shall use the term) might be a discrete-time but continuous-state model. Distinctions between countable or finite sets and uncountable sets are clearly important when referring to other parts of a model besides the time base, especially since actual digital computers can manipulate only finite sets. However, the countable-uncountable distinction drawn between time bases classifies models in a way that most closely parallels the classification arising in practice which distinguishes those who use differential equations from those who do not. We shall therefore focus on the use of different time bases, but much of what will be said is also applicable to the differences between discrete and continuous methods in general.

MODELLING TRADITIONS

On the surface, the distinction between countable and uncountable representations of time does not appear to present problems that are not already adequately treated by existing mathematical theory. Modern functional analysis subsumes both the discrete and continuous cases and makes explicit the algebraic and topological differences between spaces of countable and uncountable dimensionality. Whatever the usefulness of these results for modelling, the problems to which this article is addressed are not those of discrete and continuous *mathematics*, and we shall not touch on the logical status of the continuum nor, as we point out in the next section, on the direct empirical justification, or lack of it, for the use of continuous methods. Rather we address problems which have their roots in the isolation between discrete and continuous *traditions of model building*. The evolution of these traditions is attributable to the success of particular modelling formalisms for expressing hypotheses about observed regularities and to the successful development of techniques for deducing the consequences of these hypotheses. Discrete-time systems concepts evolved primarily as design tools for the digital technology emerging in the 1950's and as abstract models of computation. It became possible to develop methodologies for digital computing, both at hardware and software levels, that required no knowledge of the classical continuous models used so successfully in the physical sciences. The subsequent isolation is currently reflected in the very different characters of discrete-time and continuous-time theories, and in the fact that there are relatively few people who are comfortable with both.

The theory of discrete-time systems is very diverse, but it is generally characterized by an emphasis on the synthesis of a precisely specified process from a given set of primitives. For example, in switching theory one wants to construct a circuit having a specific behavior; in digital signal processing, one wants to design a filter with a certain frequency response; or, in computability theory, one wants a program using only certain instructions which computes a given function. At the risk of making too broad a generalization, the theoretical results thus involve the existence of systems with specific properties and synthesis procedures for completely describing these systems.

The emphasis in the continuous theory, on the other hand, has been on the behavioral analysis of general classes of systems having very high levels of mathematical structure. For example, systems defined by linear differential equations have much more formal structure than finite automata. Thus results exist which are both general and detailed and which relate the structure of these systems to their behavior and to the behaviors of structurally similar systems. With the exception the discrete counterparts of these linear results (about which we shall say more later), behavioral analysis methods for discrete-time systems apply to individual systems and not to general classes of discrete-time systems. One hopes, of course, that special classes of highly structured discrete-time systems will be identified, and that theories will be developed which do not as yet have continuous counterparts, but efforts in this direction are just beginning.

Even though discrete-time theory emphasizes synthesis and continuous-time theory emphasizes analysis, there are no mathematical or logical reasons for the separation of these theoretical orientations into two isolated traditions. In fact, there are numerous instances in which nearly identical mathematical structures are studied in virtual isolation from one another within different traditions. For example, automata (without the finiteness restriction) or sequential machines are, technically, essentially the same as what mathematicians call difference equations. The notations differ and usually difference equations involve more algebraic structure than the automata typically studied, but there is a one-one correspondence between these classes of structures. There is a similar correspondence between multidimensional difference equations of the kind used to approximate partial differential equations and the objects called cellular or tessellation automata.† Another example is provided by the structures which are called recursive digital filters in the field of digital signal processing.[9] These objects are the same as linear sequential machines[10] which are the same as linear difference equations.

Despite their near formal identity, automata and difference equations are part of different traditions. Difference equations, although discrete, are more closely associated with continuous methods since they are usually studied as approximations to differential equations. Consequently the study of automata differs quite drastically from the study of difference equations. There are major differences in the relevant intuition, the theoretical results that are deemed important, and the subclasses that are delineated in each area. However there are no logical obstacles to an integration of the intuition, theory, and applications of these areas. Similarly, there are no logical obstacles to the integration of discrete and continuous-time modelling formalisms and techniques. Although the system theory literature provides a framework for such integration as in Padulo and Arbib[11] and Zeigler,[12] it is not well-known among applications oriented model-builders and its consequences for modelling are just beginning to be explored.

The differences between discrete modelling and numerical analysis can be clearly seen in terms of the difference between modelling traditions. When a discrete model is derived to approximate the solution of a differential equation, the model is viewed within the tradition that surrounds the continuous modelling approach. The questions that are asked about discrete structures depend on their being viewed as approximations to continuous systems. For example, one is usually interested in error bounds and whether or not (and how) the behavior of the discrete model approaches the differential equation's solution as the step size, or mesh, converges to zero. These are questions about relationships between a continuous model and a

†Cellular automata are networks of identical automata which are interconnected in a regular way with the automata having neighboring positions in the network. Usually each component automaton, or cell, is required to have a finite state set so that the corresponding difference equation would have to be one involving functions into finite sets. This finiteness condition is the only major definitional difference even though the image of a network of automata is likely to be vastly different from an impression conveyed by a difference equation. More extensive discussions of cellular automata compared to other modelling traditions can be found in Zeigler and Barto[6] and Barto.[8]

discrete one, and the subject is usually subdivided on the basis of how discrete approximations are constructed given a differential equation rather than according to the behavioral properties of the discrete systems.

On the other hand, when a discrete model is formulated initially, it is not intended to be an approximation to a continuous model. In fact, it is often the intent of the model builder to represent aspects of a phenomenon that are considered as actually *being* discrete. The question of the desirability of a finer mesh or finer resolution level may not ever arise, and there may be no loss of information due to "undersampling" (a point we shall return to later). Indeed, the model builder need not have any experience with continuous techniques and numerical analysis. Perhaps the most familiar instance of this is in automata theoretic models of digital computing devices. Since logic gates and digital memory devices stabilize at a time scale faster than that of the driving clock pulse, it is in most cases possible to completely ignore their behavior between clock pulses. Only the stabilized component states at the end of each time interval are relevant to the future behavior of the system. Going to a time "mesh" finer than the clock frequency introduces an entirely new order of complexity to the model that is simply not required for many design and analysis purposes. Indeed, the use of a continuous modelling formalism, even if solvable, would undoubtedly obscure the behavioral simplicities that are captured by the discrete-time model.

Thus, even though the direct formulation of a discrete-time model might result in a structure similar to one that could have been derived to approximate the solution of a differential equation, the fact that it was not so derived makes a major portion of numerical analysis irrelevant. Properties of discrete systems, e.g. stability, that are critical in numerical analysis may still be important, but the complex problem of determining whether or not they reflect properties of a differential equation's actual solution need not be faced. The crucial questions in the direct approach concern the validity of of the model in accounting for observed data.

DISCRETE-TIME MODELS OF NATURAL SYSTEMS

The example of discrete-time models of digital circuits raises several questions when it is suggested that similar methods may be generally applicable for modelling systems which aren't products of conscious human design. Some of these questions quickly lead to fundamental philosophical problems which, while being relevant in a wide sense to the entire enterprise of modelling, really need not be faced in discussing discrete and continuous modelling traditions. Some of the misapprehensions about discrete and continuous modelling can be attributed, we feel, to varying assessments of what a model's validity implies about "reality". When disagreement hinges on an issue of this kind, one becomes entangled in epistemological problems that have very long histories.[13] Model builders probably would not explicitly claim that time does or does not really flow continuously, but the degree of realism implied by such a claim does tacitly contribute to misunderstanding by being implicit in strong biases toward particular modelling traditions.

It can be argued, for example, that just because *nothing relevant* happens in a digital circuit between discrete time steps does not imply that *nothing at all* happens. In fact, talking about digital components stabilizing between time steps presupposes a temporal continuum or, at least, a finer time scale at which the system's behavior can be described. This is indeed true for a digital circuit, and a similar argument might well be put forward whenever a system other than a purposefully synthesized digital system is modelled by a discrete-time system. If I am not mistaken, however, this argument is sometimes strengthened by modellers of the continuous tradition to the claim that approximations and/or omissions are *necessarily* present whenever the temporal dimension is represented by a structure other than the real numbers. Claiming that a continuous representation of time is in any sense necessary for the ultimate expression of natural regularities is tantamount to holding that nature itself is continuous, and we know much less about nature itself than whether it is discrete or continuous.

We're not saying that this realistic position is consciously held by modellers but only that it is reflected in remarks, often casually made, about discrete models. For example, after the description of a discrete-time model, it is not uncommon for an author to state that the discrete-time model represents an approximation to the differential equations which describe the actual dynamics of the system. It may well be true that such a differential equation model exists, but it would simply be another model useful for answering a possibly different set of questions. The term "dynamic" was originally associated only with continuous-time models since the differential equation was the only formalism available for modelling processes which unfold in time. We know now that the idea of dynamics is more general than the differential equation.

One possible reason for the belief that continuous representations are necessary for valid modelling was clearly expressed by A. N. Whitehead[14] in his description of a form of overstatement which he termed the "fallacy of misplaced concreteness." We have a natural tendency to overestimate the success of generalizations because we lose sight of the degree of abstraction involved. In other words, we tend to confuse models of experience with experience itself. Zeigler[12] makes a similar point when he cautions against confusing the "real system", by which he means the set of potentially acquirable data, and a "base model" of the real system. Any notion of structure or state of a system refers to some base model of the real system and not to the real system itself. Thus, what one means, for example, by the state of the real system is actually the state of a valid base model.

The disposition to reify abstractions is particularly strong when the abstractions have been very successful in accounting for observations, i.e. when models based on such abstractions have successfully undergone extensive validation testing and have displayed great predictive power. This is the case with differential equations. Models formulated as differential equations have been so successful that it is easy to overlook the considerable abstraction they entail. For example, it requires nontrivial assumptions to justify a differential equation model of a diffusion process which, modelled at a greater level of detail, is believed to arise from the interaction of discrete particles. Indeed, the classical diffusion equation would imply, if it were a valid model, that points in space which are arbitrarily far apart can instantaneously influence one another, an implication which contradicts other well established theory. The utility of the continuous formulation is obvious, but its range of validity should not be overestimated.

These remarks not only apply to dynamical models but also (and perhaps especially) to the set of real numbers. The very considerable abstraction involved in the construction of real numbers has unfortunately been obscured by treating real numbers as primitive concepts in introduction to calculus. Models formulated using real numbers and differential equations have had enormous success, but one unnecessarily limits conceptual range by concluding that models, to be valid, must be expressed within this tradition.

Another argument suggested by the digital circuit example is that since digital circuits manifest a simplicity when viewed at discrete instants of time by *design*, there is little reason to suppose that systems which are not so designed possess the properties necessary for valid discrete-time modelling. In other words, whether or not there is an underlying continuum, discrete-time models of naturally occurring systems are not likely to be valid even as far as providing *descriptions* of behavior at discrete times. This argument does not entail the same kind of overstatement as the first one. It is recognized that discrete methods *could* be applicable, but that by the nature of the phenomena to be modelled, discrete methods don't have the appropriate kind of expressive power. The evidence for this is not persuasive when one realizes that the kind of data with which continuous methods have found so much success is not the only kind of data we are capable of collecting. The inclination to reify models which are sufficiently valid has the additional consequence of limiting the kind of observations that are considered important, or worthy of investigation, or, in the extreme, worthy of perceiving. Not only is there a tendency to filter out data that is inconsistent with a popular model, but entire classes of observations which are neither inconsistent nor consistent with a model are excluded from consideration. Consequently, the apparent lack of properties which make discrete methods useful may be the result of observing only those aspects of a system which can be suitably modelled using differential equations. In Whitehead's words, "The concrete world has slipped through the meshes of the scientific net."[15]

In summary, our view is representative of the conception of modelling and model validity that is emerging from system theory. This view holds that a model's validity can only be discussed with respect to a particular set of attributes that are of interest to the modeller, that is, with respect to a specific "resolution level"[16] or a specific "experimental frame"[12] which characterizes the experimental access to a system. Thus, a model may be valid for some experimental frames and not for others, and there may be many valid models of whatever real system gave rise to a set of data. The validity of a discrete-time model does not, therefore, imply that time is discrete but only that the model faithfully summarizes a given set of data. Similarly, the validity of a continuous-time model does not imply that time is continuous. In fact, an analogy can be made between the development of this view and the change that has occurred in the way axioms and postulates are regarded. Wilder[17] discusses the gradual change, influenced by the invention of non-euclidean geometries, from the conception of axioms and postulates as *logical necessities* toward their being viewed as formal statements whose status as valid physical assertions is not a concern of mathematics. Similarly, it is possible to separate questions about the validity of dynamical models from questions about the ultimate nature of reality.

DISCRETE FUNCTIONS

Introductions to discrete-time systems usually begin with a definition of a *discrete function*, or, using the terminology of signal processing, a *digital signal*. We shall briefly discuss discrete functions and their use in modelling in order to focus sharply on some of the central issues in discrete and continuous modelling. Let I be any finite set of contiguous integers such as, for example, $\{0,1,2,3\}$. A rule f that assigns to each $x \in I$ a value $y=f(x)$ (for our purposes y will be a real number) is a discrete function. We'll write $f: I \to R$, where R denotes the set of real numbers, to describe such a function. Thus, a discrete function is simply a real-valued function whose domain is a finite set of integers rather than the more familiar interval of real numbers. We've been somewhat arbitrary in choosing this definition since it would be useful to include discrete functions which take on values that are not real numbers or which have domains of higher dimension. It might also be useful to consider functions whose domains are abstract sets that are not connected in any way with the properties of the integers (e.g. order property). However our definition is sufficient for our purposes of comparing discrete functions with more familiar concepts.

The use of the notation $y=f(x)$ for representing values of a discrete function seems to be a very natural appropriation of the usual notation for functions of the reals. Actually, however, this notation represents an interesting and subtle alteration of the usual view taken of these structures. In more traditional approaches, the similarity between discrete functions and functions of the reals tends to be obscured by the practice of calling discrete functions *sequences* or sometimes simply *vectors*. For example, a vector $f=(f_0, f_1, f_2, f_3)$ is the same as the discrete function $f: I \to R$ where $I=\{0, 1, 2, 3\}$ and $f(x)=f_x$. It's just more common in the discrete case to write arguments as subscripts and the function values as "coordinates". This very minor notational change is significant because it reflects a mingling of discrete and continuous mathematical traditions.† In the study of finite-dimensional vectors, the index I is usually important only with respect to its size, which gives the dimension of the linear space, and as a set of labels for the coordinates (thus any other set of the same size would do). Any other structure that the set I may have, such as linear order or group structure, is important only in relatively advanced topics. Given the widespread knowledge of functions having intervals of the real numbers as domains, the functional notation, as opposed to the coordinate notation, makes it more natural to consider index sets (i.e. domains) that are more than just sets. In finite dimensional linear algebra, one does not discuss, for example, a "monotonically increasing vector" or a "linear vector" since these concepts depend on the index set I having various kinds of algebraic structure. A monotonic or a linear discrete function, on the other hand, can be directly understood by analogy with the corresponding concepts for functions of the reals.

This direct correspondence between the *concept* of a discrete function and the *concept* of a function of the reals should not be confused with any sort of correspondence between *particular* discrete functions and *particular* functions of the reals. In many applications it is very natural to regard a discrete function as having been derived by sampling from a function of the reals. For example, in digital signal processing, a digital signal (i.e. a discrete function) is often viewed as having been obtained from some underlying analog signal by a sampling process. In fact, the values of digital signals are usually called samples. But, as Steiglitz[1] points out, digital signals need not have been derived from any analog signal. He gives the example of a digital signal each value of which represents the total yardage of a football team in a particular game. There is one value per game and no analog signal is involved. Not only in

†In modern functional analysis, real valued functions of the reals are viewed as vectors, an approach which represents the other side of this blending of traditions. A function $f: R \to R$ is viewed as a vector in an uncountably infinite dimensional linear space. Its values, $f(x)$, $x \in R$, are its coordinates in this space. The modern vector view of functions of the reals permits concepts which originated in a discrete or finite dimensional setting to be extended to uncountably infinite dimensional function spaces. A similar enrichment of discrete methods may occur by thinking of finite dimensional vectors as discrete functions.

practice are some discrete functions independent of any natural relationship with a function of the reals, but on the theoretical side, a discrete function can be viewed as an entirely independent, well-defined, and precisely manipulatable object. A discrete function is not, for example, a discontinuous and hence theoretically troublesome function of the reals. The domain of a discrete function, as defined above, is a set of integers. The function is not defined for real numbers between the integers.

The above remarks can shed some light on the relevance to discrete and continuous modelling of what is known as the "sampling theorem." This result says, roughly, that if a real (or complex) valued function of the reals is smooth enough (i.e. lacks spectral components above a certain frequency), then sample values can be taken at small enough intervals so that no information is lost in the process, i.e. the exact original function can be recovered. Sampling at larger intervals loses information. The smoother the function is, the larger the sampling interval can be. This is an important result in many applications, but its significance as a universal principle should not be overestimated. It is tempting to conclude, by invoking the sampling theorem, that discrete systems are only capable of faithfully representing behavior that is sufficiently smooth. However, this conclusion cannot be justified. In its usual form, the sampling theorem refers to the process of sampling a function of the reals to produce a discrete function. For discrete functions that did not arise from such a sampling process, the issue of information preservation is not illuminated by the sampling theorem. Further, the relationship between smoothness and discreteness expressed by the sampling theorem is only one particular example of a range of other possible relationships between functions of the reals and discrete functions.

To understand this last remark it's necessary to characterize the kind of result the sampling theorem expresses. The process of sampling a function of the reals is clearly a many-to-one operation from the set of all functions of the reals to an appropriate set of discrete functions. In fact, an uncountably infinite number of functions of the reals are mapped to every discrete function by the sampling operation. The sampling theorem says that if the original function happens to be a member of the subset of sufficiently smooth functions, then it can be recovered from the resultant discrete function by a suitable operation. In other words, the sampling operation restricted to the set of "smooth functions" is invertible and its inverse is known. Clearly, then, *any* function f of the reals can be recovered from its sampled version (whatever the sampling rate) provided that we have, in addition to the resultant discrete function, enough *other* information about f, namely that we know 1) that f belongs to a subset of functions on which the sampling operation is invertible and 2) we know how to compute the inverse operation for that subset. The set of "smooth" or "band-limited" functions is one such subset and is important in applications because membership in it is often a natural consequence of assumptions about the inertia of measuring instruments. Another such subset consists of functions that are non-zero only at the points to be sampled. Recovery in this case consists of simply converting the discrete function to the corresponding function of the reals whose value is zero between the sample points. No information is lost if we know that the original function f was a member of that subset. This is true even though such a function is never band-limited.

The assumption that a function has non-zero values only on a countable subset of the reals might also be a natural consequence of a modelling technique or measuring method. Discrete event systems, such as those discretely simulated using a simulation language like GPSS, can be thought of as continuous-time systems whose variables can change values only in discontinuous jumps. It's not generally a serious restriction to assume that the changes can occur at a known countable subset of the real numbers. Arrivals of customers at a bank might be represented by such a function.[12]

We emphasize, however, that it is important to distinguish between discrete-functions and functions of the reals which are non-zero at only discrete points. In fact, there is a discrete form of the sampling theorem based on the Discrete Fourier Transform[18] which sharply illustrates the magnitude of this distinction. A function of the reals which is non-zero only at discrete points is, as a function of the reals, extremely non-smooth. It has discontinuous jumps. However, it is possible to define what is meant by a sufficiently "smooth" discrete function in terms of its *discrete* Fourier components and apply the discrete sampling theorem. This result says that such a smooth discrete function can be sampled, say at every kth point, to produce a discrete function defined on a smaller domain from which the original function can be recovered by a suitable "smoothing" process. Thus, although a function of the reals having non-zero values at discrete points is very non-smooth, the corresponding discrete function obtained by sampling at these discrete points might be smooth in a discrete sense: its values at successive points not differing greatly. Of course our remarks about the possibility of other reconstruction techniques apply in the discrete case as well.

One of the most appealing aspects to the use of real valued functions of the reals is the ease with which their general properties can be grasped from graphical representations. Although graphs of discrete functions can be similarly displayed, they are not so well suited to the back-of-the-envelope figuring that plays an important part in using mathematical methods. The graph of a discrete function consists of all the ordered pairs $(x,f(x))$, $x \in I$. and might be displayed as in Figure 1. For small domains it's probably more natural to use the vector n-tuple notation. Thus, $(3, -4, 4.5)$ could be considered a kind of graph of the function $f: I \rightarrow R$ where $I = \{0, 1, 2\}$. One does not normally represent f's graph as in Figure 2. But this representation is clearly possible and is certainly more suggestive for vectors which are being viewed as discrete functions. Of course, digital computers can generate displays of discrete functions with no difficulty, and one can always pretend, in casual figuring, that a smooth curve represents the graph of a discrete function. It's noteworthy to relate a remark made by Greenspan[2] that if $I = \{k(10)^{-6} | k = 0, 1, \ldots, 10^6\}$, the graph of $f: I \rightarrow R$ given by $f(x) = x^2$, when drawn on a normal book page, is indistinguishable to the naked eye from the graph of $f(x) = x^2$ for x in the real interval $[0, 1]$.

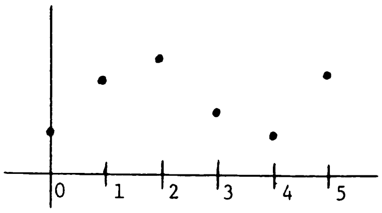

FIGURE 1 The graph of a discrete function.

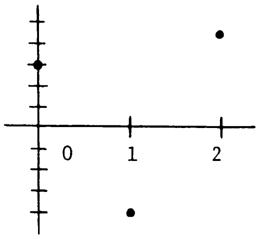

FIGURE 2 The vector $(3, -4, 4.5)$ could be graphed as a discrete function $f: I \rightarrow R$ where $I = \{0, 1, 2\}$..

This remark also illustrates the fact that it is perfectly feasible to use symbolic expressions to define discrete functions. For the formula $f(x) = x^2$, $x \in I$, to make sense it is only necessary that multiplication of elements of I is a meaningful operation and always results in an element of f's range. It is also possible to define operators on discrete functions in terms of symbolic manipulation of these formulae. Thus, by turning to discrete functions one does not give up the possibility of concise symbolic expression. One gains, however, the advantage that using symbolic expressions is not the *only* means of completely specifying functions as it is in the continuous case. Discrete functions can be completely defined by listing their values, e.g. storing the values in a computer so that "addresses" correspond to function arguments and "contents" correspond to function values, or by providing an algorithm whose input is a function argument and whose output is the corresponding value. The latter possibility is referred to, as in the opening quotation of this article, as replacing equations by algorithms. In a sense, of course, the description of an algorithm, or even the tabulation of a function's values, is a symbolic means of defining a function which, in principle, could be written as an equation. However, in its usual usage, the term equation refers to formulae consisting of constants, variables, the symbols for basic arithmetical operations, and a variety of other symbols to indicate differentiation, integration, etc. The primitive operations used in specifying algorithms (e.g. looping and conditional branching) permit the concise definition of functions which are impossible or very awkward to express by conventional algebraic means.

SYMBOLIC AND COMPUTATIONAL SIMULATION

Differential equations and difference equations are both ways of specifying the constraints that are assumed to act locally in a system to produce its global behavior. For systems that describe temporal processes, these constraints act locally in time. A differential equation is used to formalize these local constraints as relations which must hold between the values of a system's attributes at any time and how these values are changing at that same time. For difference equations, the constraints are expressed as relations between present attribute values (and perhaps past values) and their values at the next discrete time step. In both the continuous and discrete cases, the objective is to determine what global behavior is implied by an initial condition and the uniform application of these local constraints at each point in time. The global behavior is the solution of the equation. The behavior is a function of the real numbers for a differential equation and a discrete function for a difference equation.

The term simulation usually refers to the process by which one determines a model's behavior from its structure, or, in the terminology used above, the method used to find a model's global behavior based on knowledge of an initial condition and the locally acting constraints. Since an equation's solution can be viewed as the behavior of a model specified by the equation, it is possible to think of solving an equation as a form of simulation. Although this term is usually applied only to discrete systems or to certain kinds of discrete approximations of continuous systems in which solutions are explicitly generated step-by-step, it is not misleading to think of symbolically solving an equation in closed form as a form of simulation. We can call this *symbolic simulation.* If, on the other hand, a computer is used to generate a model's behavior, we call the process *computational simulation.* It is more conventional to refer to computer simulation of models as computational *solution*, but the idea of simulation is more general than the concept of solution since the latter refers to equations, i.e. certain symbolic expressions, which are not the only means of specifying models. In many cases equations are not explicitly used since an algorithmic formulation may be more feasible or more intuitively appealing.

One of the major justifications for formulating models in terms of functions of the reals and derivatives is that there are many symbolic methods which can be useful for manipulating functions of the reals. Perhaps the most basic symbolic methods rest on the use of rules for differentiating or integrating functions by merely manipulating their formulae. The ability to symbolically determine and use derivatives and integrals helps make the notion of rate-of-change so powerful a conceptual primitive. In formulating a model as a set of differential equations, one hopes that symbolic methods will be useful in finding its behavior.

Although symbolic simulation techniques exist for certain kinds of discrete models, they are neither as well-known nor as well-developed as they are for continuous models. Simulation by computer is usually the method used to gain insight into a discrete model's behavior. This is especially true for automata theoretic models which are often expressed using set-theoretic language in which symbols for arithmetic operations have no meaning. The local constraints comprise a transition function which may be given by a table rather than by a formula. Simulation becomes, in effect, a series of table look-ups which can be performed exactly and quickly by digital computer. For discrete models having more algebraic structure, the transition function may be expressed by arithmetical formula which can be computed at each time step by means of arithmetic, or, as Greenspan[2] emphasizes, *high-speed* arithmetic. Other kinds of discrete models such as discrete-event models of queuing systems are also conveniently simulated computationally whether symbolic simulation techniques exist or not.

Computational simulation of discrete-time models is trivial compared to computationally approximating the behavior of a continuous model (although efficient computational simulation of large discrete models can involve many complexities). Hence, the fact that symbolic techniques are helpful for only the simplest, i.e. linear, differential equations is used to justify the direct formulation of discrete-time models and the complete by-passing of continuous models. Further, it is pointed out by Greenspan[2] that the experimental results which are to be modelled are originally discrete sets of data. "Theoreticians then analyze these data and, in the classical spirit, infer continuous models. Should the equations of these models be nonlinear, these would be solved today on computers by numerical methods, which results again in discrete data. Philosophically, the middle step of the activity sequence... is consistent with the other two steps. Indeed, it would be simpler and more consistent to replace the continuous model inference by a discrete model inference..."[2]. Simulation is then performed arithmetically by a high speed computer. Many of the subtleties of numerical approximations of continuous models simply do not arise.

However persuasive this argument is, there is another aspect to the use of differential equations and symbolic methods which should be considered. A formula giving the solution of a differential equation does more than give a single behavioral trajectory of the model. By containing parameters used in the model description, a solution's formula can express the form of the behavior for a large class of initial conditions, forcing functions, and perhaps for a large class of related models. A computational simulation, on the other hand, produces a single trajectory for a single initial condition and forcing function. Symbolic methods give coherence to *classes* of models and help increase our understanding of systems by showing, concisely, what structural factors contribute to what behavioral characteristics. Indeed, in some cases it is possible to deduce certain properties of a system's general behavior without ever determining a single actual behavioral trajectory. A single computational simulation of a discrete model does not produce this kind of understanding unless it is one of a pattern of simulation runs designed to establish structural-behavioral correspondences through computational experimentation. This difference between computational and symbolic simulation is the same as the difference between arithmetic and algebra. Algebraic methods allow generalized arithmetic problems to be solved by separating the logical form of a problem from its specific arithmetical computations.

We do not mean to imply that symbolic techniques are applicable only to continuous-time models, and we shall discuss their use for certain types of discrete-time models later. Our point here is that the computational power so immediately available for discrete simulation makes it all too easy to obtain specific results without their contributing to an understanding of a system's dynamics. A specific discrete model's behavior is very easily generated, but it is a much more difficult task to develop a feeling of "why" the model behaved in a particular way. True, computational simulation can be applied where known symbolic methods are completely inadequate, but it is too pessimistic to argue that the only models which can be concisely understood are those specified by equations which have already been thoroughly treated by classical methods, i.e., by "simple" equations. It may in fact be tautological to say that symbolic solutions are possible only for simple (i.e. linear) equations since the simplicity of a process is directly related to concise expression of its regularities. To someone not knowing the linear theory, linear systems undoubtedly would appear to be quite complex.

High speed computational methods have the potential for helping us understand a much larger class of models by producing large numbers of specific results which can act as guides to theorizing. This is what von Neumann meant by the *heuristic use* of computers. He felt that by generating computer solutions to many specific equations, one might be able to discover general properties and develop a corresponding theory.[19] Although specific results rather than general theories are ultimately the aim of model building, the specific results which are most useful, judging from classical continuous modelling, are often those which follow from a general perspective about a class of related models.

Discrete-time models are very suitable for this approach to the use of computers, but it is also true that symbolic methods are applicable to those discrete models which are algebraically analogous to the continuous models which can be symbolically analyzed. In particular, linear difference equations can be understood as thoroughly as linear differential equations by the application of finite dimensional linear algebra (which is substantially simpler than the uncountably infinite dimensional linear algebra of functional analysis). It's true that the discrete linear theory lacks some of the subtlety of the continuous theory, but one wonders how much of this complexity is useful in modelling and how much has been generated by the theory itself. For example, the convergence of infinite sequences of functions of the reals is a central problem of functional analysis. Yet if experimental data is always a discrete set and only a finite number of experiments are ever performed, when does the convergence problem arise apart from in the analysis of an inferred continuous model? For many applications, the discrete linear theory suffices, but since the continuous theory is so readily available, it is automatically adopted whether its additional complexities are needed or not.

One example of this tendency to uncritically adopt continuous methods involves the use of methods based on Fourier analysis. These methods are often learned in a specific context so that an understanding of the principles used is tied to a specific formulation. Among the misconceptions arising in this way is the belief that functions need to be real or complex valued functions of one or several real variables in order for Fourier analysis techniques to be applicable. Less standard applications tend to be viewed as approximations to the continuous case or as necessarily derived from the continuous case. As a result, it is often felt that no real working knowledge can be achieved unless one feels comfortable with integrals, continuity, convergence properties, distributions and other concepts necessary for a thorough understanding of functions of real variables. However, as the discrete form of the Fourier Transform becomes more widely known it is becoming clear that a prior knowledge of the continuous theory is unnecessary.

The increased interest in discrete Fourier analysis is due to the widespread use of the Fast Fourier Transform (FFT) algorithm for computing the Discrete Fourier Transform (DFT)[†]. The DFT can be understood *completely* within an algebraic framework that involves none of the complexities (e.g. convergence and distributions) of the continuous Fourier Transform. *Any* real or complex valued discrete function of N points can be expressed *exactly* as a unique linear combination of the discrete functions $\phi_{\omega}(x) = e^{2\pi i \omega x/N}$ where ω and x are integers between 0 and $N-1$ (inclusive). If one is careful to remember that the convolution theorem for the discrete case refers to cyclic or circular convolution, then the DFT can be applied to discrete-time linear systems in the same way that the continuous Fourier Transform applies to continuous-time linear systems. Moreover, it is not necessary to know the continuous theory or even to know that the continuous theory exists.

When obvious advantages do not result from the formulation of a model within the continuous tradition, the continuous framework represents what might be called mathematical overkill. The conceptual subtlety of continuous models brings with it the possibility of applying powerful analysis techniques. But when these techniques are not used, the formulation entails unnecessary complexities—the choice of a continuous model being determined by the researcher's background, the tradition prevalent within a discipline, or because the modeller is unaware of or unfamiliar with other modelling techniques. On the other hand, there are instances in which discrete models and computational simulation are used when there exist applicable classical methods. For example, instead of exhaustively listing a function in tabular form, it may have been possible to express it concisely by a formula; or, instead of resorting to computational simulation, it may have been possible to formulate a model as a symbolically tractable system of difference or differential equations. This represents what might be called mathematical underkill since existing and pertinent methods are not used. Both mathematical overkill and underkill result from the relative isolation in which the discrete and continuous traditions have developed and can be minimized by an integration of methodologies which keeps the issues of model validity and predictive power in the forefront.

BEHAVIORAL REPERTOIRE

We have emphasized that a discrete-time system need not be based, intuitively or formally, on an underlying continuous-time model. The answer to the question "what happens between time steps?" may be simply "nothing" or "nothing relevant." Yet if discrete modelling ever displaces classical continuous methods to a substantial degree, it will be pertinent to ask whether or not there are significant differences between the classes of system behavior that can be accounted for by discrete models, on the one hand, and differential equation models on the other. To fully answer this question would require, at least, a careful specification of what is meant by "accounted for" which we shall not attempt to do here. Rather, we shall point out some facts which bear on this question.

First, for the less problematic side of the issue, there are examples of behaviors which can be generated by very simple discrete-time models which, if similar behavior even could be produced by a differential equation, would probably require a much more complex specification. May[22] indicates that simple nonlinear differential equations describing population growth (e.g. the logistics equation) describe systems with very simple behavior, whereas the corresponding simple nonlinear difference equations have very complex behaviors, some of which are aptly described as "chaotic". Instead of regarding the behavioral regimes of these discrete-time models as artifacts of discrete approximations to continuous processes, one could view them, as May indicates, as possibly valid representations of actually observable phenomena. Thus, while not conclusively demonstrated, it is plausible that the repertoire of discrete-time models includes behaviors which cannot be produced by differential equations of similar complexity.

But what about the other side of the question? Are there classes of behavior exhibited by differential equation models which cannot be accounted for by discrete models? Clearly, since the behavior

[†] Although the theory of discrete Fourier analysis is less complex than the continuous case, it developed much later than the continuous theory as a special case of the abstract theory of harmonic analysis of functions whose domains are topological groups.[20,21] The discrete case is the simplest special case of this theory since all the topological complexities disappear when attention is restricted to finite groups with the discrete topology (every function of such a group is technically a continuous function!). A relatively minor computational innovation (the FFT) has made what was formerly a trivial special case of a very abstract general theory into a subject that is now studied and applied without reference to the rest of the theory.

produced by a discrete-time model consists of discrete functions, a discrete-time model, by definition, cannot possess the temporal resolution level (assuming that high resolution is desirable which it often is not) of a continuous-time model. Thus, at the outset, we could conclude that *no* continuous behavior can be accounted for by a discrete model. However, in applications this lack of resolution is not the real problem since the time step can be made as small as necessary to represent the behavior with *enough* fine temporal detail as required. Since measuring instruments result in discrete data, the detail required is always less than that provided by a function defined on the entire continuum.

The major problem arises because of the notorious difficulty in providing a discretely specified model (e.g. a difference equation) whose behavior *exactly* agrees with the behavior of the continuous model at designated time samples. In the terms of numerical analysis, the simulation of such a discrete model would produce a numerical approximation that is "infinitely accurate at the mesh points," that is, the behavior of the discrete model would be equal to the sampled behavior of the continuous model. The discrete models constructed in numerical analysis behave only approximately like the continuous system and only do so in a restricted time period even if an ideal computer producing no round-off error is imagined. That infinitely accurate numerical approximations exist only in rare instances leads one to suspect that the behavioral repertoire of the class of continuous-time models may be richer than that of the class of discrete-time models, even putting aside the issue of temporal resolution. That is, given a continuous-time model there may not exist a discrete-time model whose structure generates, in a local manner, behavior which is equal to the sampled continuous behavior.

However, with a very general definition of well-specified discrete and continuous-time systems it can be shown that this is not true. For every continuous-time system there does exist a discrete-time system whose behavior agrees exactly with that of the continuous system at time samples which are integer multiples of an arbitrarily chosen positive number. This result, due to Zeigler,[12] says that such a discrete-time system exists, but it does not imply that it is easy to construct that system given only the continuous model's structure, e.g. given only a differential equation. The actual construction of the discrete system requires a knowledge of the continuous system's behavior, e.g. the solution of the differential equation, so that this result is not helpful for initially finding the solution. The rare cases where infinitely accurate numerical methods exist are those instances in which the discrete-time system can be constructed without this behavioral knowledge.

Nevertheless, the existence of such exact discrete models is important in the discrete modelling tradition where continuous models are not used at all. By restricting oneself to the use of discrete models one does not, *ab initio*, exclude the possibility of generating, from local rules and up to any discrete resolution, the full range of behaviors produced by continuous models. The problem of the construction of continuous-discrete model pairs does not arise since discrete models are formulated directly.

It is beyond the scope of this article to develop the theoretical framework in which this result can be rigorously proven, but we can indicate the character of the result by discussing a simple example. The initial value problem given by

$$\frac{dq}{dt}=aq,\ q(0)=q_0 \qquad (1)$$

has the solution

$$q(t)=q_0 e^{at}, t \geqq 0. \qquad (2)$$

The corresponding discrete-time system is specified by the difference equation

$$q(t+h)=e^{ah}q(t), q(0)=q_0 \qquad (3)$$

The solution of (3) is the discrete function q_0e^{at} for $t \in \{kh|k=0, 1, \ldots\}$. This can be seen by induction where the crucial step follows from the fact that by eq. (3) $q(t+h)=e^{ah}(q_0e^{at})=q_0e^{a(t+h)}$. Note that the coefficient e^{ah} in eq. (3) is obtained from the solution of the differential equation and not from the equation itself.

The correspondence in this example depends on the specific property of exponential functions that $a^x \cdot a^y = a^{x+y}$. However this fact can be viewed as a specific form of a property possessed by other well-defined dynamical systems. This property is known as the *semigroup property* of state determined systems[23] or, in a more general form which includes input, as the *composition property*[12]. Very briefly, a system with this property can be characterized at any time t by a *state* q_t such that its state at any time

$t+h$ is a function of q_t and h but not of t.† Suppose a continuous-time system starts in the initial state q_0 at time $t=0$ and at any time $t=h$ is in state q_h. If the system satisfies the composition property, then q_h is a function of q_0 and h. We can write

$$q_h=\delta(q_0, h).$$

If q_0 and q_h are indeed states of the system, then

$$q_{2h}=\delta(\delta(q_0, h), h)$$

and, in general

$$q_{(k+1)h}=\delta(q_{kh}, h)$$

Thus, the function δ can be used to iteratively generate a discrete function which agrees with the behavior of the continuous system at times which are integer multiples of h. For the system specified by the differential equation (1), $\delta(q_0, h)=q_0 e^{ah}$. Similar constructions are possible whether the system is linear or nonlinear.‡

Numerical analysts have not focused on discrete systems derived in this way from continuous systems since, as we have said, the behavior of the continuous system needs to be known for the derivation. If this behavior were known, a discretization would be unnecessary. Zeigler[12] remarks, however, that the behavior need only be known for the time interval $[0, h]$ which might be determined by a standard numerical method. Using this result for specifying a discrete-time system and then iteratively generating the behavior of the discrete system for longer periods may result in a decrease of error propagation. In addition, given a model consisting of interconnected components that in isolation from other components are described by differential equations whose solutions are known, it is possible to simulate the model using the discrete-time version of the components.[24]

A result having the character of that reported here is also useful in applications where synthesis rather than analysis is necessary, for example, if a discrete-time system is to be constructed whose behavior should approximate the known behavior of a continuous-time system. The theory of digital signal processing is partially concerned with this task, and the kind of construction described here is closely related to the technique called impulse invariant filtering.[9] A digital filter can be designed and implemented on a digital computer whose impulse response is equal (except for quantization error) to the sampled impulse response of an arbitrary continuous-time linear filter.

The reason for reporting this result here is that it shows that in a very strong sense the behavioral repertoire of discrete-time models is at least as rich as that of continuous-time models. The constructional difficulties occur only in going from a continuous model to the corresponding discrete model. If a discrete model were constructed to account for empirically observed behavior, rather than to match the behavior of a continuous model, these difficulties do not arise. Of course there remains the possibility that in a particular application a valid continuous model may be easier to construct than a valid discrete model, but restricting attention to discrete models does not further limit the kind of behavior that can be generated.

We can take this result further. Most standard numerical methods are based on approximations to the derivative, i.e. on discrete versions of the rate-of-change concept. The form of the "infinitely accurate" discrete systems described here indicates that simply *change* rather than rate-of-change is the appropriate conceptual primitive for discrete modelling. The difference equation (3) was formulated on the basis of asking "what change does the continuous system undergo from $t=0$ to $t=h$?" A similar question might be asked about a system under observation in the natural world possibly resulting in the direct formulation of a valid discrete-time model. It might be argued here that examining mere change rather than rate-of-change is what postponed the understanding of motion provided by Newton. The quantities that remained invariant in simple mechanical systems were velocities or accelerations. However, since Newton's time we have learned a great deal about modelling dynamical processes and, in particular, about what is meant by the state of a system. The invariant property of a system is, more generally, a function that tells how states change to other states. The concept of instantaneous velocity is appropriate for modelling certain kinds of systems, but it is only one way of specifying such a function.

†For the case of systems without input this implies *time-invariance*, but if input is considered it's possible for a system to have the composition property without being time invariant. See Zeigler.[12]

‡For the case with input, a continuous input segment defined from time $t=0$ to $t=h$ is regarded as a single input symbol. This is quite natural for piecewise constant input functions (as in sampled-data systems) but can be extended to other kinds of input segments.[12]

CONCLUSIONS

We have tried to articulate a point of view and to present some facts which would dispel common criticisms of discrete modelling as an alternative to modelling with differential equations. Most of these criticisms seem to be the result of unfamiliarity with discrete styles of mathematical thinking and a tendency to reify the abstractions used in models which have such long histories of success. There remains, however, a set of issues that cannot be so resolved. Rather than being criticisms of the *principle* or *potential* of discrete modelling, these issues pertain to current discrete modelling practices and to the fact that classical models of dynamical systems happen not to be expressed in discrete form.

Digital computers permit simulation of models whose complexity far transcends the current possibilities of symbolic techniques. In many cases, the purpose of such simulations is to experiment with alternative configurations of an existing or proposed physical construction without having to actually alter or construct it until numerous possibilities have been tried with the model. For example, models of industrial processes, traffic flow, and computer operating systems are often simulated for this purpose. Such models are not designed with an explicit goal of helping to "understand" the system. The system, in fact, may be one whose entire mechanism is regarded (perhaps mistakenly) as exposed and already understandable. This is why in much of the literature on this kind of modelling the term "simulation" rather than "modelling" is emphasized. The appropriate model is regarded as almost obvious while the emphasis is placed on the generation and analysis of its behavior. Theory, aside from statistical theory, plays very little role in this process.

The adoption of this style of modelling and simulation for purposes of unraveling observational patterns in order to "explain" or "understand" them immediately leads to difficulties. The terms explanation and understanding are admittedly problematic, but at the very least their meaning involves the ability to embed a particular model into a larger existing conceptual framework. In many scientific areas, the existing conceptual frameworks rely so heavily on continuous mathematics that discrete models, even if valid, tend to appear as merely descriptive models without adequate explanatory significance. Indeed, explanatory significance may in fact be lacking as long as contact with classical theory is not established or as long as sufficiently encompassing discrete theories are not constructed. The methodological biases produced by a prevailing modelling formalism are, in a sense, justified by their own prior existence. The often asked "What if digital computers were available to Newton?" is, after all, an academic question.

One can't, of course, conclude that continuous methods (including numerical analysis) must therefore remain the major tools in scientific modelling. On the contrary, the ease with which discrete modelling and simulation techniques can be applied in situations where classical methods are completely inadequate is precisely what is needed for further theoretical development. Computational experimentation can suggest the form of structural-behavioral correspondence in classes of systems that are not yet understood, but computational power alone is not a substitute for the careful simplification and theoretical generalization that have helped make classical methods so fruitful.

REFERENCES

1. K. Steiglitz, *An Introduction to Discrete Systems.* John Wiley, New York, 1974.
2. D. Greenspan, *Discrete Models.* Addison-Wesley, Reading, Mass., 1973.
3. D. Greenspan, "An Algebraic, Energy Conserving Formulation of Classical Molecular and Newtonian n-Body Interaction." *Bulletin of the American Mathematical Society,* **79,** No. 2, March 1973.
4. D. Greenspan, "An Arithmetic Particle Theory of Fluid Mechanics." *Computer Methods in Applied Mechanics and Engineering,* **3,** North Holland, 1974.
5. D. Greenspan, "Computer Newtonian and Special Relativistic Mechanics." *Second U.S.A. Japan Computer Conference Proceedings,* 1975.
6. B. P. Zeigler and A. G. Barto, "Alternative Formalisms for Biosystem and Ecosystem Modelling." *Simulation Councils Proceedings Series,* **5,** No. 2, Dec. 1975.
7. J. W. Sutherland, *Systems: Analysis, Administration, and Architecture.* Van Nostrand Reinhold, New York, 1975.
8. A. G. Barto, "Cellular Automata as Models of Natural Systems." Ph.D. Thesis, Computer and Communication Sciences, The University of Michigan, 1975.
9. B. Gold and C. M. Rader, *Digital Processing of Signals.* McGraw-Hill, New York, 1969.
10. M. A. Harrison, *Lectures on Linear Sequential Machines.* Academic Press, New York, 1969.
11. L. Padulo and M. A. Arbib, *System Theory.* Saunders, Philadelphia, Pa., 1974.
12. B. P. Zeigler, *Theory of Modelling and Simulation.* John Wiley, New York, 1976.
13. B. F. Gaines, "System Identification, Approximation and Complexity." *International Journal of General Systems,* **3,** No. 3, 1977, pp. 145–174.
14. A. N. Whitehead, *Science and The Modern World.* Free Press, New York, 1967.

15. A. N. Whitehead, *Modes of Thought*. Free Press, New York, 1966, p. 18.
16. G. J. Klir, *An Approach to General Systems Theory*. Van Nostrand Reinhold, New York, 1969.
17. R. L. Wilder, *The Foundations of Mathematics*. John Wiley, New York, 1967.
18. J. W. Cooley, P. A. W. Lewis and P. D. Welch, "The Finite Fourier Transform." *IEEE Trans. on Audio and Electroacoustics*, **Au-17**, No. 2, June 1969.
19. J. von Neumann, *Theory of Self-Reproducing Automata*. Edited by A. W. Burks, University of Illinois Press, Urbana, 1966.
20. W. Rudin, *Fourier Analysis on Groups*. John Wiley, New York, 1962.
21. E. Hewitt and K. A. Ross, *Abstract Harmonic Analysis*. Springer-Verlag, Berlin, 1963.
22. R. M. May, "Biological Populations Obeying Difference Equations: Stable Points, Stable Cycles, and Chaos." *Journal of Theoretical Biology*, **51**, No. 2, June 1975, pp. 511 524.
23. R. E. Kalman, P. L. Falb, and M. A. Arbib, *Topics in Mathematical Systems Theory*, McGraw-Hill, New York, 1969.
24. B. P. Zeigler, "Persistence and Patchiness of Preditor-Prey Systems Induced by Discrete Event Population Exchange Mechanisms." (Submitted to *Journal of Theoretical Biology*).

Andrew G. Barto is currently a research associate at the Department of Computer and Information Sciences, University of Massachusetts, Amherst, Mass. He is on leave from the faculty of the School of Advanced Technology, State University of New York at Binghamton. He studied both as an undergraduate and as a graduate student at the University of Michigan receiving a B.S. in mathematics in 1970 and a Ph.D. in computer science in 1975. His interests include system theoretic models of the modelling process, computational tools for modelling, and model building as it is related to human cognition. In connection with this latter area, he is particularly interested in highly distributed parallel and associative information processing.