NASA TECHNICAL
MEMORANDUM

Report No. 53868

STUDIES IN SYSTEM SIMULATION

R. Saeks
Department of Electrical Engineering
University of Notre Dame

Prepared for the Computation Laboratory
System Analysis Branch

August 27, 1969

NASA

*George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama*

ACKNOWLEDGMENT

The research reported herein was conducted while the author was a participant in the NASA-ASEE Summer Faculty Fellowship Program at the Marshall Space Flight Center. The author gratefully acknowledges the help given to him by his research counterpart, Dr. H. Trauboth, and the directors of the fellowship program both at Marshall and the University of Alabama.

TABLE OF CONTENTS

Page

SUMMARY 1

INTRODUCTION 1

DIGITAL SYSTEM SIMULATION BY MEANS OF COMPONENT DISCRETIZATION 2

Introduction 2
Stability Analysis 5
Digital Simulation 12
Conclusions 13

PADÈ APPROXIMATION AND THE DEGREE REDUCTION PROBLEM . . 14

Introduction 14
System Specification 15
Degree Reduction Problems 20
Conclusions 24

LINEAR DATA COMPRESSION FOR TOPOLOGICAL MATRICES 25

Introduction 25
Representation Theory 26
Recovery Algorithms 32
Conclusions 35

REFERENCES 37

TECHNICAL MEMORANDUM X-53868

STUDIES IN SYSTEM SIMULATION

SUMMARY

The results of three mathematical studies of the feasibility of various methods of large scale digital system simulation are discussed in this report.

A class of digital simulation formulae for large scale systems wherein numerical methods are applied directly to the system components is considered. Stability and error analyses are undertaken and function analytic techniques are employed to derive a class of stable, error preserving numerical methods for such simulation formulae. Techniques for implementing the component discretization process for systems containing both continuous and discrete components are indicated.

The simulation of a dynamical system by a lower order system is considered. The problem is formulated in terms of the Taylor series coefficients of the system impulse response from which lower order Padè approximations may be obtained by computationally appealing formulae.

A data compression scheme for integer matrices wherein prime numbers are used for addressing is formulated. The technique allows a vector with integer valued entries to be characterized by a single positive number that is a linear function of the given vector and from which the column vector can be uniquely recovered by purely algebraic techniques. The procedure is computationally limited only by the magnitude and storage accuracy required for the characterizing integer and is ideally suited for the characterization of topological matrices, which because of their inherent sparseness, are not affected by these limitations.

INTRODUCTION

The advent of ever larger systems over the past decade has necessitated significant changes in the methods and theory of system simulation. Systems with a handful of components and 10 or 20 degrees of freedom have given way to systems containing hundreds of components and having thousands of degrees of freedom. As such the semi-intuitive simulation procedures that once sufficed are giving way to formal algorithmic procedures designed to make optimal use of available computational capacity. Such procedures require that one make the maximum possible use of the connectivity structure inherent in a system, i.e., isolating large systems into smaller subsystems, minimizing the complexity of such subsystems, etc. The studies discussed in this report are aimed to the further development of such techniques.

This report is composed of three such studies of the theory underlying various techniques of large scale system simulation. Although formulated with computational feasibility in mind, these studies are mathematical, not computational, in nature. The results have not been implemented nor have computer programs been written. Rather, mathematical studies of the feasibility of certain approaches to the large scale system analysis problem are undertaken. Existence theorems and series representations are used throughout the derivation of the various results, though once formulated the various results can be implemented by purely algebraic techniques without iteration or root finding.

All three studies are motivated by the realization that practical large scale simulation must deal with the components of a system as separate identities that are combined only after considerable analysis is undertaken at the component level. The first study, therefore, deals with the problems associated with applying numerical techniques directly to the system components, and a class of numerical methods that circumvent the stability problems inherent in such an approach are delineated. The second study is concerned with the possibility of approximating a subsystem by one of lower order prior to combining it with the remaining system components. Finally, the third study deals with a data compression technique for storing the large topological matrices encountered in large scale system simulation.

DIGITAL SYSTEM SIMULATION BY MEANS OF COMPONENT DISCRETIZATION

Introduction

A differential system is typically composed of a collection of dynamical components characterized by (in general nonlinear) differential equations

$$Dx_i = f_i(x_i, I_i, t) \qquad (1a)$$

$$i = 1, 2, \ldots, m$$

$$O_i = h_i(x_i, I_i, t) \qquad (1b)$$

together with a set of algebraic connection equations of the type

$$K(I, O, U, W, t) = 0 \quad . \qquad (2)$$

Here "D" is the differential operator, $I = \mathrm{col}(I_i)$ and similarly for O and X while U and W are system input and output vectors respectively. Note that the equations induced by the common connection models (block diagram, linear graph, etc.) are always linear. It is, however, often convenient[1] to include the effects of nonlinear algebraic components in the connection (rather than the component) equations; hence we allow for the possibility of nonlinear connection equations.

Such a system is typically simulated by combining the component equations with those for the connections to form a differential equation characterizing the entire system[2, 3]

$$DX = F(X, U, t) \qquad (3a)$$

$$W = H(X, U, t) \quad , \qquad (3b)$$

which may be simulated by standard numerical methods. Although the process of forming equations (3) from (1) and (2) is often complex, we may symbolically write

$$F(X, U, t) = K_F[f_i](X, U, t) \qquad (4)$$

and

$$H(X, U, t) = K_H[h_i](X, U, t) \qquad (5)$$

[1] N. Prasad and J. Reiss, The Digital Simulation of Interconnected Systems (in preparation).

[2] Ibid.

[3] R. Saeks, State Equation Formulation for Multipart Networks, IEEE Transaction on Circuit Theory (to appear Feb. 1970).

where K_F and K_H are function valued (F) functions of a function variable (f_i) determined by the connection equations. The digital simulation of such a system thus takes the form

$$\lambda(E)\overline{X} = \overline{F}(\overline{X}, U, t) = \overline{K}_F[f_i](\overline{X}, U, t) \qquad (6a)$$

$$\overline{W} = H(\overline{X}, U, t) = \overline{K}_H[h_i](\overline{X}, U, t) \qquad (6b)$$

where $\lambda(E)$ is a function of the shift operator, E, which in some sense approximates the derivative, and $\overline{F} = \overline{K}_F$ is a corresponding approximation of $F = K_F$.

An alternate approach to the simulation problem is to apply numerical methods directly to the component differential equations, (1), yielding discrete approximations to the given differential components characterized by the difference equations

$$\lambda_i(E)\overline{x}_i = \overline{f}_i(\overline{x}_i, \overline{I}_i, t) \qquad (7a)$$

$$i = 1, 2, \ldots, m$$

$$\overline{O}_i = h_i(\overline{x}_i, \overline{I}_i, t) \qquad (7b)$$

where $\lambda_i(E)$ and $\overline{f}_i$ are approximations, as before, of D and f_i. Now observing that the functions K_F and K_H, being determined entirely by the connection constraints, are unchanged by the component discretization, we obtain a discrete simulation of the overall system as

$$\lambda(E)\overline{X} = K_F[\overline{f}_i](X, U, t) \qquad (8a)$$

$$\overline{W} = K_H[h_i](\overline{X}, U, t) \qquad (8b)$$

where $\lambda(E) = \text{diag}\,[\lambda_i(E)]$. This approach to the simulation problem is termed digital simulation by component discretization and is the subject of the present work.

The component discretization concept is not new, having long been employed in the special case of linear time-invariant systems wherein one replaces the "Laplace transform" of a differential system by the "z-transform" of a discrete approximation. The difficulty with these approaches, and the basic problem to be considered in the sequel, is two-fold. First, a "good" approximation at the component level may not yield a "good" system approximation, and secondly, stable component approximations may yield an unstable system approximation. This latter problem has been considered by Fowler [1] for the case of linear time-invariant systems composed of a single loop, but no general solution has been given.

In the following sections the concepts of approximation error and stability of a numerical method are formalized, and necessary and sufficient conditions for a numerical method applied to the components of a system to yield a stable system equation are given. Finally, in the last section various techniques for implementing a digital simulation program by component discretization are indicated. In particular the simulation of systems containing both continuous and discrete components is considered and the possibility of using different numerical methods for different components is explored.

Stability Analysis

A numerical method is defined as the substitution of the difference equation

$$\lambda(E)\overline{x} = \overline{f}(\overline{x}, I, t) \qquad (9)$$

for the differential equation

$$Dx = f(x, I, t) \quad . \qquad (10)$$

1. Definition. A numerical method is component stable if the difference equation, (9), resulting from the substitutions

$$\lambda(E) \longleftrightarrow D$$

and

$$\overline{f} \longleftrightarrow f$$

is stable whenever the given differential equation, (10), is stable.

2. Definition. A family of numerical methods is system stable if each numerical method is component stable and moreover, the system difference equation

$$\lambda(E)\overline{X} = K_F[\overline{f}_i](\overline{X}, U, t)$$

resulting from the substitutions

$$\begin{aligned} \lambda_i(E) &\longleftrightarrow D \\ & & i = 1, 2, \ldots, m \\ \overline{f}_i &\longleftrightarrow f_i \end{aligned}$$

in the component differential equations is stable whenever the system differential equation

$$DX = K_F[f_i](X, U, t)$$

is stable.

The concept of component stability has been widely studied for both the cases of ordinary and partial differential equations. In the particular case of a linear time-invariant system with

$$\lambda(E) = E - 1 \tag{11}$$

necessary and sufficient conditions on the "approximation" $\overline{f}$ of f to assure component stability are known [2]. In the case of system stability, where one requires that numerical methods applied to the components of a stable system yield a stable difference equation characterizing the entire system, a " theory" is available only for single loop systems [1]. In the following, necessary and sufficient conditions for a numerical method to be system stable are derived, and the error inherent in the resulting approximation is analyzed.

Initially we observe that if one allows unrestricted connections (except for the requirement that a differential equation, such as (6), characterizing the entire system exist), the function K_F may have arbitrarily large sentitivity. Thus any small variation in the f_i will yield, for a sufficiently ill behaved K_F, to an unstable system difference equation. We therefore have:

3. Lemma. A necessary condition for a family of numerical methods

$$\begin{aligned} &\lambda_i(E) \longleftrightarrow D \\ &\overline{f}_i \longleftrightarrow f_i \end{aligned} \qquad i = 1, 2, \ldots, m$$

to be system stable is that

$$\overline{f}_i = f_i \qquad i = 1, 2, \ldots, m$$

Of course since system stability subsumes component stability we also have:

4. Lemma. A necessary condition for a family of numerical methods

$$\begin{aligned} &\lambda_i(E) \longleftrightarrow D \\ &\overline{f}_i \longleftrightarrow f_i \end{aligned} \qquad i = 1, 2, \ldots, m$$

to be system stable is that they are component stable.

Consistent with the preceding lemmas it will be necessary to delineate those component stable numerical methods for which $\overline{f}_i = f_i$ (that is a condition on the function $\lambda_i(E)$, which assures component stability when $\overline{f}_i = f_i$ is required). Of course we must also assure that the operator $\lambda(E)$ approximates the derivative operator if the solutions of the discretized system are to approximate those of the given differential system, and moreover, $\lambda_i(E)$ must be chosen so as to assure the finite dimensionality of the discrete approximating system. For linear time-invariant systems the class of $\lambda_i(E)$ that satisfies these conditions is characterized by the following lemma:

5. Lemma. Let an arbitrary linear time-invariant component be characterized by a stable differential equation. Then a necessary and sufficient condition on the numerical method

$$\begin{aligned} &\lambda_i(E) \longleftrightarrow D \\ &\overline{f}_i = f_i \end{aligned}$$

to assure that the resulting discretized component will be characterized by a stable, finite dimensional difference equation whose solutions approximate those of the given component (differential equation) is that the function $\lambda_i(w)$, viewed as a complex valued function of a complex variable, satisfy the following conditions:

a. $\lambda_i(w)$ is rational.

b. $\lambda_i(w)$ approximates $\ln(w)$.

c. If $|w| > 1$, $\lambda_i(w)$ is analytic and $\mathrm{Re}\lambda_i(w) > 0$.

Proof: Condition a is the usual necessary and sufficient condition for a difference equation to be finite dimensional while condition b assures that $\lambda_i(E)$ approximates D for if

$$\lambda_i(w) \approx \ln(w) \tag{12}$$

then

$$\lambda_i(E) \approx \ln(E) = D \tag{13}$$

Here the validity of the operator approximation (13) follows from the complex function approximation by means of the spectral mapping theorem [3] and the operator equality of equation (13) is the usual semi-group representation of the shift operator through its infinitesimal generator, D [3]. Finally condition c is a necessary and sufficient condition for component stability. This may be demonstrated by representing the component differential equation by its "transfer function," $H_i(p)$, in which case the substitution

$$\lambda_i(E) \longleftrightarrow D \tag{14}$$

corresponds to the complex function substitution

$$\lambda_i(z) \longleftrightarrow p \tag{15}$$

where p is the "Laplace Transform" variable and z is the "z-Transform" variable. Clearly the difference equation resulting from the substitution (14) has "transfer function"

$$\overline{H}_i(z) = H_i[\lambda_i(z)] \tag{16}$$

Now $H_i(p)$ is stable if, and only if, it is analytic in the region $\mathrm{Re}\,p > 0$ while $\overline{H}_i(z)$ is stable if, and only if, it is analytic in the region $|z| > 1$. If condition c is satisfied and the given differential equation is stable, $\overline{H}_i(z)$ is the composition of a function taking the region $|z| > 1$ analytically to the region $\mathrm{Re}\,p > 0$ and a function analytic in this region. $\overline{H}_i(z)$ is therefore analytic in the region $|z| > 1$ and thus represents a stable difference equation, as required. Conversely, if condition c is not satisfied, one can always find an $H_i(p)$ for which $\overline{H}_i(z)$ does not represent a stable difference equation. The three conditions therefore combine to yield a component stable numerical method of the required type while the failure of any one implies that the numerical method is either unstable, infinite dimensional or does not approximate the given differential equation.

It should be noted that the above lemma yields a condition that assures that every stable differential equation will be approximated by a stable difference equation. There are, of course, many values of function $\lambda_i(E)$ which do not satisfy the conditions of the lemma yet still take some stable differential equations to stable difference equations. For instance the forward difference formula takes stable differential equations with eigenvalues in a restricted region to stable difference equations [2], but it does not take all stable differential equations to stable difference equations. Although the above proof of the lemma is valid only for the linear time-invariant case, we conjecture (but have not proven) that the result holds in general.

The result of lemma 5, which gives a necessary and sufficient condition for a class of numerical methods to be component stable, also, upon combination with lemmas 3 and 4, yields a necessary condition for system stability. In fact, this condition is necessary and sufficient.

6. Theorem. Let the stable components of an arbitrary stable linear time-invariant system be discretized by the substitutions

$$\begin{aligned} \lambda_i(E) &\longleftrightarrow D \\ \overline{f}_i &\longleftrightarrow f_i \end{aligned} \qquad i = 1, 2, \ldots, m$$

Then a necessary and sufficient condition to assure that the resultant discretized system difference equation is stable, finite dimensional and has solutions that approximate those of the given system differential equation is that

$$\bar{f}_i = f_i \qquad i = 1, 2, \ldots, m$$

and $\lambda_i(w)$, viewed as a complex valued function of a complex variable, satisfies the following three conditions.

a. $\lambda_i(w)$ is rational.

b. $\lambda_i(w)$ approximates $\ln(w)$.

c. If $|w| > 1$, $\lambda_i(w)$ is analytic and $\mathrm{Re}\lambda_i(w) > 0$.

Proof: The necessity of the conditions follows from the preceding lemmas. To demonstrate the sufficiency we must show that the discrete system difference equation resulting from the process of the theorem is system stable, finite dimensional and has solutions that approximate those of the given differential system. The latter two properties follow from the corresponding properties of the discretized components and the sufficiency of lemma 5. Finally for stability the sufficiency of lemma 5 assures that each component is stable while the discrete system difference equation induced by the component discretization is

$$\lambda(E)\bar{X} = K_F[f_i](X, U, t) \tag{17}$$

where $\lambda(E) = \mathrm{diag}[\lambda_i(E)]$. Now equation (17) is just the discretization of the given system differential equation, viewed as a single component, by means of the numerical method

$$\lambda(E) \longleftrightarrow D \tag{18}$$

$$\bar{K}_F[f_i] = K_F[f_i] \tag{19}$$

Since each $\lambda_i(E)$ satisfies the conditions of lemma 5 so does their direct product, $\lambda(E)$; hence the numerical method of equations (18) and (19) is component stable (by lemma 5) and therefore takes the stable system differential equation to a stable system difference equation. The specified collection of numerical methods is therefore system stable and the theorem is proven.

The theorem completely delineates the class of numerical methods that can be successfully employed for digital system simulation by component discretization. Two questions, however, remain. First, what is the effect of the component discretization process on simulation error? Secondly, do any numerical methods satisfying the conditions of the theorem exist? In the former case we observe that the system difference equation obtained by the component discretization process (17) is precisely the same difference equation that would have been obtained by applying the numerical method of equations (18) and (19) directly to the system differential equation. We therefore have:

7. Corollary. The simulation error resulting from the process of theorem 6 is identical to that which would result upon applying the numerical method

$$\lambda(E) \longleftrightarrow D$$

$$\overline{K}_F[\overline{f}_i] = K_F[f_i]$$

directly to the system differential equation.

Finally it must be determined whether any numerical methods satisfying the conditions of the theorem exist. This is indeed the case and, in fact, both trapezoidal

$$\lambda_i(E) = (E + 1)^{-1}(E - 1) \qquad (20)$$

and backwards

$$\lambda_i(E) = E^{-1}(E - 1) \qquad (21)$$

integrations satisfy the required conditions as does the second order scheme

$$\lambda_i(E) = (2E^2)^{-1}(3E^2 + 1) \qquad (22)$$

Of course numerical methods of arbitrarily high order that satisfy the required conditions can be obtained by standard approximation techniques. It is noteworthy that the conditions of the theorem are never satisfied when λ_i is a polynomial in E, as is the case for Simpson's rule and most standard integration techniques.

Digital Simulation

A digital simulation program employing component discretization has two distinguishing characteristics. First, since one discretizes continuous components as a first step in such a program, components that are discretely specified may be included in the system simply by skipping the initial discretization step. Secondly, since each component is discretized independently, different approximations $[\lambda_i(E)]$ may be used for different components. One may therefore use "better" numerical methods for simulation of those components that have small time constants and/or whose behavior is highly sensitive than for the remaining components.

A component discretization simulation program might allow for four classes of dynamical components along with the usual memoryless and connection components. These include continuous finite dimensional components specified by an ordinary differential equation, continuous (possibly infinite dimensional) components specified by a convolution integral, discrete components specified by either a difference equation or a discrete convolution, and finally components specified by a computer program, such as might arise if data analysis techniques are used to estimate the dynamics of an unknown device. Clearly if such a system is to be digitally simulated, one must use component discretization since some of the components are specified discretely and others (characterized by convolution integrals) cannot be stored on a computer in continuous form even though they are specified continuously.

In implementing such a program, one would input the discrete components as given and the continuous components together with a specified numerical method, i.e. $\lambda_i(E)$, satisfying the conditions of theorem 6. Now as a first step the program discretizes the continuous components. The finite dimensional components characterized by differential equation are discretized by making the substitution

$$\lambda_i(E) \longleftrightarrow D \qquad (23)$$

and then converting the resultant higher order difference equation to a first order difference equation by standard methods. On the other hand those components specified by convolution integrals are discretized by converting them to a discrete convolution with weighting coefficients determined by the numerical method. Once all of the components have been discretized, the remainder of the simulation may be carried out by standard discrete system methods.

It is interesting to note that the process of discretizing a continuous component by the numerical methods of theorem 6 is essentially a matter of rearranging and indexing the matrices characterizing the given continuous component and involves no "real computation." Additional computation, corresponding to the degree of the numerical method, is, however, reflected in the readout process wherein one is dealing with a difference equation of higher order than the given differential equation. For this reason no computational savings is gained by carrying out the discretization at the component level since the readout process is still carried out at the system level. This is not the case for those numerical methods with $\bar{f}_i \neq f_i$, in which case the calculation of $\bar{f}_i$ is usually more easily carried out at the (essentially decoupled) component level than at the system level, but such processes do not have the system stability characteristics required to successfully carry out the component discretization process.

Conclusions

Although we have carried out a rather long and tedious derivation, the results of the theory are readily applicable. As long as one employs integration techniques of the type indicated by theorem 6, system stability is assured independently of the connections or type of components in a system. In fact, even if one integrates with too large a step size, the resultant solution, though inaccurate, will tend toward zero rather than becoming unstable.

It is interesting to note that in a number of computer aided network analysis programs wherein component discretization is used, it has been found experimentally that trapezoidal integration is stable while Simpson's rule and other polynomial integration routines may be unstable. This is, of course, verified by our theory, which also yields a means for obtaining higher order stable integration routines. Another area wherein component discretization is employed is in the construction of a Digital Difference Analyzer. In such a device one immediately replaces integrators in the program with a discrete approximation independently of the remainder of the program and is therefore using component discretization, if implicitly. As in the case of network analysis, it has been found experimentally that trapezoidal integration is stable, as indicated by the present theory.

PADÈ APPROXIMATION AND THE DEGREE REDUCTION PROBLEM

Introduction

In system analysis it is commonly desired to replace a given system by one that has a lower degree (order) but whose external behavior is similar or identical to the given system. Such problems are termed "degree reduction problems" and are the subject of the present work.

Possibly the most common manifestation of the degree reduction problem in system analysis is the problem of eliminating uncontrollable and/or unobservable modes from a system to be simulated. That is the removal of internal responses that have no effect on the overall external system behavior. Although such modes represent unlikely situations for a single component, they occur commonly in interconnected systems wherein the effect of a mode in one component is canceled by an equal and opposite effect in another component. For instance in an electric network two parallel capacitors exhibit one rather than two independent modes.

A second class of degree reduction problems encountered in system analysis is concerned with modes that have a small but nonzero effect on the overall system behavior. Such characteristics are often encountered in control systems wherein a mode is nominally unobservable but, because of component variations and/or simulation error, appears in the output with a small residue. For instance a plant mode that has nominally been canceled by compensation techniques may in fact appear with a small residue because of variations of the components from their nominal values. In such a case the computational saving achieved by neglecting such a mode may justify the error induced into the simulation.

Finally practical considerations, such as the complexity of the system or the amount of memory required for simulation, may force one to approximate a given system by one of lower degree even at the cost of considerable simulation error. We thus have a third class of degree reduction problems — approximation.

The study of these three classes of degree reduction problems in a context amenable to digital system simulation is the purpose of this work.

Since the theory is essentially a reformulation of the realizability theory of Youla [4, 5], Ho [6, 7], and Kalman[4], we state the main results without proof, referring to the recent text of Kalman, Falb, and Arbib [8]. In the following a "universal system specification" is formulated wherein a system is characterized by an infinite sequence of matrices A_i. This sequence, which has only a finite number of independent terms and is thus computationally feasible, can be obtained by inspection from the system impulse response, transfer function, and state equations or directly from measured data and thus serves as a natural intermediary between the various system specifications. Once such a sequence has been constructed (starting with any of the usual system characterizations or measured data), minimal state equations for the system that solve the various degree reductions problems are obtained. Since the entire procedure is algebraic, the resulting algorithms are ideally suited for digital system simulation, no approximation or iteration steps being required.

System Specification

The most common methods for characterizing a linear time-invariant (finite dimensional) dynamical system are the state equation

$$DX = FX + GU \tag{24a}$$

$$Y = HX \tag{24b}$$

the system impulse response, $K(v)$, such that

$$Y(t) = \int_{-\infty}^{\infty} K(t-q)\,U(q)\,dq \tag{25}$$

and the transfer function, $T(p)$, such that

$$Y(p) = T(p)U(p) \tag{26}$$

where $Y(p)$ and $U(p)$ are the "Laplace Transforms" of $Y(t)$ and $U(t)$ respectively. Now it is well known that these characterizations are equivalent yet the interrelationship between the various characterizations are quite complex. If one, however, expands $K(v)$ and $T(p)$ in appropriate series, a commonality between the three characterizations can be found. Indeed this commonality corresponds to readily measurable parameters of the system.

[4] B. L. Ho and R. E. Kalman, the Realization of Constant Input-Output Maps, SIAM Journal on Control (to appear).

Consider a system impulse response matrix, $K(v)$, which if it represents a finite dimensional dynamical system, has a Taylor series expansion

$$K(v) = \sum_{i=1}^{\infty} A_i v^{(i-1)} \tag{27}$$

about the point $v = 0$. Here the ith coefficient matrix, A_i, is the $(i-1)$st derivative of $K(v)$ evaluated at $v = 0$ and the coefficient sequence completely characterizes the system. Although as defined, the A_i sequence appears to be infinite, only a finite number of the terms are independent; hence the values of A_i form a computationally feasible characterization for a finite dimensional dynamical system. In fact, we have the following fundamental result, which completely characterizes the finite dimensionality of the A_i sequence.

1. Theorem. Let $K(v)$ be the impulse response of a finite dimensional system. Then there exists a set of real constants, $b_1, b_2, \ldots, b_n$, such that

$$A_i = -\sum_{k=1}^{n} b_k A_{k+i-n-1} \qquad i = n+1, n+2, \ldots$$

Moreover, the smallest n for which this is true is the dimension of the minimal system (i.e. number of state variables) that realizes $K(v)$ exactly, and the minimal polynomial of such a system is

$$\lambda(z) = z^n + b_1 z^{n-1} + b_2 z^{n-2} + \ldots + b_n$$

A short proof of this fundamental theorem is given in Reference 8. Consistent with the theorem, a knowledge of the first n A_i's together with the n b_i's is sufficient to completely characterize the system. Equivalently the values of the first $2n$ A_i's completely characterize the system since the values of b_i's can be calculated from these by means of the equality of the theorem.

Although the values of A_i have been derived in terms of the system impulse response, they have an equally natural interpretation in terms of the system transfer function. In this case one takes a Laurant expansion of $T(p)$ about infinity and obtains the series expansion

$$T(p) = \sum_{i=1}^{\infty} A_i/p^i \tag{28}$$

The fact that the Laurant coefficients are indeed the same as the Taylor coefficient of the impulse, response follows immediately upon an application of the initial value theorem. Of course theorem 1 holds whether the A_i sequence is obtained from the impulse response or the transfer function.

Clearly the A_i sequence serves as a natural intermediary between the frequency and time domains, and in fact completely characterizes the properties of both. In fact the A_i sequence is also naturally related to the state characterization of the system. To see this we observe that for any state equation that realizes a given (external) system

$$K(v) = He^{Fv}G \quad , \tag{29}$$

which upon differentiating and evaluating at zero yields

$$A_i = HF^{i-1}G \tag{30}$$

The solution of the converse problem of identifying a state equation, such as (24), or equivalently the three matrices H, F, and G, given a sequence of A_i , is by no means obvious. Fortunately it has a computationally appealing solution, though one that requires a rather tedious derivation. This result, due to Youla [4] and Ho [6, 7] is predicated on the properties of a Hankel matrix associated with the A_i sequence. We therefore let J be the block symmetric matrix.

$$J = \begin{bmatrix} A_1 A_2 A_3 \dots & A_n \\ A_2 A_3 A_4 \dots & A_{n+1} \\ A_3 A_4 A_5 & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ A_n \; A_{n+1} \dots & A_{2n-1} \end{bmatrix} \qquad (31)$$

where n is any integer such that the condition of theorem 1 holds. Similarly we define J' to be the same matrix with the subscripts shifted up by one, i.e., the 1-1 entry in J' is A_2 and the n-n entry is A_{2n}. With this mode of specifying the A_i sequence, a minimal state equation realizing the system characterized by the A_i sequence exactly may be obtained by purely algebraic manipulation.

2. Theorem. Let a system have Hankel matrix, J, and let P and M be arbitrary nonsingular matrices such that

$$PJM = \left[\begin{array}{c|c} I_n & 0 \\ \hline 0 & 0 \end{array}\right]$$

Then

$$DX = FX + GU$$

$$Y = HX$$

is a state equation of minimal dimension realizing the given system exactly if

$$F = \left[\begin{array}{c|c} I_n & 0 \end{array}\right] PJ'M \left[\begin{array}{c} I_n \\ \hline 0 \end{array}\right]$$

$$G = \left[I_n \,\vdots\, 0 \right] PJ \begin{bmatrix} I_r \\ \hline 0 \end{bmatrix}$$

$$H = \left[I_p \,\vdots\, 0 \right] JM \begin{bmatrix} I_n \\ \hline 0 \end{bmatrix}$$

Here I_n is the n by n unit matrix, n is the rank of J, I_r and I_p are r and p dimensional unit matrices, r is the number of system inputs, p is the number of outputs, and in the partitioned matrices, 0 is the zero matrix of conformable dimension.

Although the proof of the theorem is quite involved (see for instance Reference 8) its application is quite straightforward. Operationally J is formed from the A_i sequence and is diagonalized by P and M. These matrices are then used to calculate F, G, and H. Although it is necessary to diagonalize J, P and M need not be in any way related; hence the diagonalization may be carried out by independent row and column reduction processes, which yield the diagonalization in a fixed number of steps without iteration or root determination.

If one knows a priori a minimal set of b_i, as in the next section, such that theorem 1 is satisfied, an even simpler state realization that eliminates the diagonalization is possible.

3. Theorem. Let a system have Hankel matrix, J, and let b_i, $i=1, 2, \ldots, n$ be a minimal set of real numbers such that the equality of theorem 1 holds. Then

$$DX = FX + GU$$

$$Y = HX$$

is a state equation of minimal dimension realizing the given system exactly if

$$F = \begin{bmatrix} 0 & I_n & & & & \\ & & I_n & 0 & & \\ & 0 & & I_n & & \\ & & & & \ddots & \\ & & & & & I_n \\ -b_1 I_n & & \cdots & & & -b_n I_n \end{bmatrix}$$

$$G = J \left[\begin{array}{c} I_r \\ \hdashline 0 \end{array}\right]$$

$$H = \left[\begin{array}{c:c} I_p & 0 \end{array}\right]$$

Consistent with the preceding development the A_i sequence, or more accurately its first $2n$ terms, serves naturally as a universal system specification for linear time-invariant (finite dimensional) dynamical systems. The sequence can be obtained from either the time, frequency or state representation or alternatively from direct measurements of the system. Conversely any of the three system representations can be obtained from the A_i sequence by purely algebraic manipulation. Moreover, the close relationship between the A_i sequence and measurable system parameters (the impulse response, frequency response, etc.) renders it an ideal medium in which to carry out degree reduction and/or system approximation.

Degree Reduction Problems

With the formulation of the A_i sequence and its relationship to the common system models we may proceed to attack the various degree reduction problems. In fact, much of the preceding theory was developed with the solution of the first degree reduction problem in mind (i.e., the problem of eliminating uncontrollable and/or unobservable states from a state equation). This is achieved by starting with an arbitrary state equation representing a given system, such as

$$DX = FX + GU \tag{32a}$$

$$Y = HX \quad , \tag{32b}$$

and constructing (the first 2n members of) its A_i sequence as per equation (30). Now J and J' are constructed from this sequence and used in theorem 2 to obtain a minimal state equation having the same input-output behavior as the given equations.

$$DZ = \overline{F}Z + \overline{G}U \tag{33a}$$

$$Y = \overline{H}Z \quad . \tag{33b}$$

As such, one can construct a pair of state equations that have the same external behavior as the given equations but without the unobservable and uncontrollable states; hence the first degree reduction problem is solved. In fact, the solution is entirely algebraic.

Unlike the first degree reduction problem, the second and third problems require that one find a state equation whose impulse response approximates that of a given higher order system. One must therefore choose a mode of approximation before a solution can be obtained. There are, of course, many possible approaches to the approximation problem; for instance, L_1, L_2, Chebychev, etc., each of which has merits in various contexts. In the present problem a Padè approach [9] will be taken, primarily because of the computational simplicity inherent in such an approach; that is, an impulse response $\overline{K}(v)$ is said to be a kth order Padè approximation of an impulse response $K(v)$ if the first k Taylor series coefficients of $\overline{K}(v)$ coincide with those of $K(v)$. In terms of the A_i sequence we therefore require that

$$\overline{A}_i = A_i \qquad i = 1, 2, \ldots, k \quad . \tag{34}$$

Consistent with the close relationship between the A_i sequence and the various system models, such a mode of approximation is computationally appealing (if not of great physical relevance).

From theorem 1 it follows that given any arbitrary set of matrices

$$\overline{A}_i \qquad i = 1, 2, \ldots, n \tag{35}$$

and any set of n real numbers

$$b_i \qquad i = 1, 2, \ldots, n \tag{36}$$

the A_i sequence defined by equation (35) for $i \leq n$ and by

$$\overline{A}_i = -\sum_{k=1}^{n} b_k \overline{A}_{k+i-n-1} \tag{37}$$

for $i > n$ has an n dimensional state equation realization (for instance as obtained by theorem 3). Moreover, the first n Taylor series coefficients of the corresponding impulse response are the specified values of $\overline{A}_i$, and the minimal polynomial of the resulting realization is

$$\lambda(z) = z^n + b_1 z^{n-1} + b_2 z^{n-2} + \ldots + b_n \tag{38}$$

Clearly if one is given an arbitrary A_i sequence, we can construct an nth order system, which is an nth order Padè approximation to the given A_i sequence, simply by letting

$$\overline{A}_i = A_i \qquad i = 1, 2, \ldots, n \tag{39}$$

and choosing the remainder of the $\overline{A}_i$ sequence by use of equation (37) for some arbitrarily specified set of b_i. In fact, since the values of b_i are arbitrary, one can completely specify the system dynamics and still achieve the required approximation. Formally we have:

4. Theorem. Let an arbitrary system be characterized by an A_i sequence. Then for any set of coefficients

$$b_i \qquad i = 1, 2, \ldots, n$$

the sequence

$$\overline{A}_i = A_i \qquad\qquad i = 1, 2, \ldots, n$$

$$\overline{A}_i = -\sum_{k=1}^{n} b_k \overline{A}_{k+i-n-1} \qquad\qquad i = n+1, n+2, \ldots$$

characterizes an n-dimensional system having minimal polynomial

$$\lambda(z) = z^n + b_1 z^{n-1} + \ldots + b_n$$

which approximates the given system to the nth degree in the Padè sense.

Consistent with theorem 4 the second and third degree reduction problems are solvable by algebraic means if one employs the Padè mode of approximation. It should be noted that the Padè mode of approximations assures that the approximation of the impulse response will be "good" (near $v = 0$) while stability assures that both the actual and approximate impulse responses will tend to zero for large v ; hence the Padè mode of approximation is quite reasonable. Of course the quality of the approximation can be improved if one makes an appropriate choice of the characteristic polynomial $\lambda(z)$, which is arbitrary except for the requirement that it be stable. One approach that further improves the approximation near $v = 0$ is to choose the b_i's so as to increase the order of the Padè approximation. That is the values of b_i are chosen so that

$$\overline{A}_i = A_i \qquad\qquad n < i \leq m \quad , \qquad (40)$$

thereby obtaining an mth order approximation with an nth order system. The circumstances under which this can be done have been studied[5] and will not be delineated here. If appropriate rank conditions on J are satisfied, it is, however, possible to achieve a Padè approximation with order as high as 2n .

An alternative approach, which allows one to control the approximation when v is neither large nor small, is to choose $\lambda(z)$ so as to approximate the characteristic function of the given system. In this way one can guarantee that the dynamics of the approximate system (i.e., oscillitory frequencies, decay

[5] B. L. Ho and R. E. Kalman, op. cit.

rates, etc.) are similar to those of the given system. In particular for the second degree reduction problem where one simply desires to eliminate negligible but nonzero modes if $\lambda(z)$ is taken as the factor of the characteristic function of the given system corresponding to the nontrivial modes, then the dynamics of the approximate system will be the same as those of the given system except for the deletion of the negligible modes (i.e. those with small residues) and slight variations of the residues to the remaining modes to compensate for the effects of the deleted poles at $v = 0$. The use of Padè approximations of a system impulse response thus yields a complete solution for the second degree reduction problem and a solution to the third degree reduction problem in those circumstances wherein the Padè criterion is relevant. Moreover, in both cases the procedure is completely algebraic and computationally appealing.

Conclusions

Although the derivation of the preceding results is quite complex, the results lend readily to the development of computational algorithms for the solution of the degree reduction problem. The required steps to implement such an algorithm are as follows.

1. First Degree Reduction Problem. Given an arbitrary state equation representing a given system, characterized by the three matrices F, G, and H we first form the A_i sequence for the system via $A_i = HF^{i-1}G$. Now these values of A_i are used in theorem 2 to form new minimal state equations, characterized by the matrices $\overline{F}$, $\overline{G}$, and $\overline{H}$. Since this equation is minimal, all uncontrollable and unobservable modes have been removed while, according to the theory, the system (as observed externally) is unchanged.

2. Second Degree Reduction Problem. Given an arbitrary state equation characterized by matrices F, G, and H, which has characteristic polynomial $\lambda(z)$, let us assume that $\lambda(z) = \lambda_1(z)\,\lambda_2(z)$ where $\lambda_1(z)$ corresponds to n significant modes and $\lambda_2(z)$ corresponds to m negligible modes. Now the second degree reduction problem is to find an nth order system that is the same as the given system except for the elimination of the negligible modes. First one forms the A_i sequence via $A_i = HF^{i-1}G$ and lets $\overline{A}_i = A_i$ for i less than or equal to n. Now the values of b_i are taken as the coefficients

of $\lambda_1(z)$ and the remainder of the $\overline{A}_i$ sequence is formed via equation (37). Finally either theorem 2 or 3 is used to find the approximate nth order system characterized by matrices $\overline{F}$, $\overline{G}$, and $\overline{H}$.

3. Third Degree Reduction Problem. Given an arbitrary kth order system, characterized by matrices F, G, and H and characteristic polynomial $\lambda(z)$, the third degree reduction problem is to find an nth order system that approximates the given system in some sense. Initially we choose $\overline{\lambda}(z)$ to be an nth order polynomial that in some sense approximates $\lambda(z)$. Now the A_i sequence, $A_i = HF^{i-1}G$, is formed and used to define $\overline{A}_i = A_i$ for i less than or equal to n. Finally the remainder of the $\overline{A}_i$ sequence is obtained by equation (37), using the coefficients of $\overline{\lambda}(z)$ for the b_i's, and the state equations for the approximating system, with matrices $\overline{F}$, $\overline{G}$, and $\overline{H}$, are obtained either by theorem 2 or 3. The key to the third degree reduction problem is the choice of the approximation used to choose $\overline{\lambda}(z)$. A decision on this can only be made in the context of the application. One possible choice, however, is to use a Padè approximation about zero. Since this would assure that the low frequency characteristics of the given system were preserved by the approximate while the Padè approximation of the A_i sequence assures that the high frequency characteristics are preserved, one would expect a reasonably close approximation.

LINEAR DATA COMPRESSION FOR TOPOLOGICAL MATRICES

Introduction

The topological, or connection, information in a large scale system is often conveniently characterized by a connection matrix of some type. Such matrices typically have integer entries corresponding (in some sense) to the existence of a connection between two components and zeros in the remaining entries. Since in a large scale system a given component is typically connected only to two or three others, these matrices are generally sparse, though this sparseness may have no discernible pattern. With the ever increasing size of modern systems, such matrices may have tens of thousands of entries; hence some form of data compression is necessary if they are to be effectively employed in the digital simulation of such systems.

There are two basic approaches to the topological data compression problem. One is based on the fact that memory words are used inefficiently by topological matrices (since many of the entries are zero and need not be stored) and the other based on the fact that available word length is not used efficiently, since the entries are usually taken from an alphabet of small integers (including zero) which do not require a full word length for storage. A number of data compression techniques based on the first approach are commonly used. Typically these store only the nonzero entries together with an address and are often predicated on an a priori knowledge that the nonzero entries are arranged in some specified pattern (such as one nonzero entry per column, etc.). It is a data compression scheme of the second type that is considered here wherein a single element from a large alphabet characterizes a vector of elements taken from a small alphabet. As such, one stores a large integer rather than a collection of small ones, hence making more efficient use of available word length. In the theory neither sparseness nor "smallness" of the matrix entries is required, but computational considerations demand both; hence the technique is well suited for topological matrices but not generally applicable.

In the following sections the data compression scheme is developed without regard to computational considerations (i.e. sparseness, magnitude of the entries, etc.), first for vectors with nonnegative integer entries, then for vectors with arbitrary integer entries and finally for matrices with arbitrary integer entries. In all cases the scheme is shown to be linear (either in the sense of a group or a semi-group); hence linear operations on the matrices or vectors may be implemented by carrying out corresponding operations on their compressed characterizations. Finally algebraic recovery algorithms are formulated and an error analysis is carried out wherein the limitations that must be imposed on the technique to assure accurate recovery of the given matrix via computational methods are delineated.

Representation Theory

Initially we will consider a column vector of nonnegative integers. The length of such a vector need not be restricted and hence these vectors may be taken as infinite if one requires, in addition, that they have only a finite number of nonzero entries (i.e. they are zero almost everywhere). Such a vector is then a member of the space

$$V = \sum_{i=1}^{\infty} N^0 \tag{41}$$

(N^0 the nonnegative integers) which forms a semi-group under componentwise addition. A vector a in V is denoted by $a = (a_1, a_2, a_3, \ldots) = (a_i)$. Also let p_i be the ith prime number. That is $p_1 = 2$, $p_2 = 3$, $p_3 = 5$, $p_4 = 7$, $p_5 = 11$, etc. (An ordered list of prime numbers is given in Reference 10.)

Now given any vector a in V , let $\theta(a)$ be the positive integer

$$\theta(a) = \prod_{i=1}^{\infty} p_i^{a_i} \tag{42}$$

Note that since a in V is zero almost everywhere all but a finite number of the factors $p_i^{a_i}$ are unity; hence for all practical purposes the product of equation (42) is finite and can be readily computed. The possibility of identifying a single integer with a column vector of integers is certainly not surprising and can be done by any number of formulae. What is possibly more surprising in this case, however, is that the function $\theta(a)$ is a linear function on the semi-group V and moreover an isomorphism. As such, a can be recovered uniquely from $\theta(a)$. To this end we have:

1. Theorem. The function

$$\theta : V \longrightarrow N^1$$

$$a \longrightarrow \theta(a)$$

defined by equation (42) is a semi-group isomorphism. (Here N^1 is the multiplicative semi-group of positive integers).

Proof: To prove the theorem we must show that the function, θ , is one to one, onto and linear. In the first case if $\theta(a) = \theta(b)$, then

$$\prod_{i=1}^{\infty} p_i^{a_i} = \theta(a) = \theta(b) = \prod_{i=1}^{\infty} p_i^{b_i} \tag{43}$$

Now since only a finite number of the factors $p_i^{a_i}$ and $p_i^{b_i}$ are not unity, the unique prime factorization theorem [10] applies to equation (43); hence $a_i = b_i$ for all values of i showing that $a = b$, as required. Therefore θ is one to one.

To verify that θ is onto N^1, let n be an arbitrary positive integer with prime factorization.

$$n = \prod_{i=1}^{m} p_i^{a_i} = \prod_{i=1}^{\infty} p_i^{a_i} \tag{44}$$

where on the right side of equation (44), a_i is taken as zero if p_i is not a prime factor of n. Clearly

$$n = \theta(a) = \theta((a_i)) \tag{45}$$

and θ is therefore onto.

Finally if a and b are in V,

$$\theta(a + b) = \prod_{i=1}^{\infty} p_i^{(a_i+b_i)} = \left(\prod_{i=1}^{\infty} p_i^{a_i}\right)\left(\prod_{i=1}^{\infty} p_i^{b_i}\right) = \theta(a)\theta(b) \quad ; \tag{46}$$

hence θ is linear as required.

Note that repeated application of the formula $\theta(a+a) = \theta(a)\theta(a)$ yields the scalar multiplication formula

$$\theta(ka) = \theta(a)^k \tag{47}$$

for any nonnegative integer k. The theorem assures, at least in theory, that a vector a in V can be recovered uniquely from $\theta(a)$, but no computational algorithm for the recovery process is indicated by the proof. In fact several computationally appealing algorithms are possible and will be formulated in the following section.

The preceding theory can be extended to the case of vectors with entries composed of arbitrary integers if one adopts a rational, rather than integer, representation. In this case one is dealing with vectors from the space

$$U = \sum_{i=1}^{\infty} N \tag{48}$$

of infinite vectors having a finite number of nonzero integer entries. Clearly U is a group under componentwise vector addition.

For an arbitrary vector a in U let

$$a = a^+ - a^- \tag{49}$$

be a decomposition of a into the difference of two vectors from V and define $\theta(a)$ as

$$\theta(a) = \theta(a^+)/\theta(a^-) \tag{50}$$

where $\theta(a^+)$ and $\theta(a^-)$ are defined as per equation (43) for a^+ and a^- in V . This is well defined for if $\underline{a}^+$ and $\underline{a}^-$ represent the decomposition of a wherein the positive entries are taken for $\underline{a}^+$ and the negative entries for $\underline{a}^-$, any other decomposition has the form $a^+ = \underline{a}^+ + k$ and $a^- = \underline{a}^- + k$ where k is in V ; hence

$$\begin{aligned}\theta(a^+)/\theta(a^-) &= \theta(\underline{a}^+ + k)/\theta(\underline{a}^- + k) = \theta(\underline{a}^+)\theta(k)/\theta(\underline{a}^-)\theta(k) \\ &= \theta(\underline{a}^+)/\theta(\underline{a}^-) \qquad ,\end{aligned} \tag{51}$$

showing that $\theta(a)$ is independent of the decomposition employed. Also $\theta(a)$ defined as per equation (50) is an extension of $\theta(a)$ as defined in equation (42). That is if a is in V , then the two definitions of $\theta(a)$ coincide (by taking $a^- = 0$). An argument similar to that of theorem 1 will yield the following result wherein it is shown that θ is a group isomorphism when defined on U ; hence the unique recovery and linearity properties obtained for θ when operating on V also hold for its extension.

2. Theorem. The function

$$\theta : U \longrightarrow Q^+$$

$$a \longrightarrow \theta(a)$$

defined by equation (50) is a group isomorphism.

Although we have formulated the preceding theorem in terms of rational numbers, in practice one would probably store the numerator and denominator integers separately, in which case it would only be necessary to recover a vector in V from an integer and one would not have to deal with rational numbers computationally. To justify such an approach, however, it is necessary to show that a can be recovered uniquely from $\theta(a)$ independently of the integers used to represent $\theta(a)$. Assuming an algorithm is available with which to calculate $\theta^{-1}(n)$ for any integer (see the following section), it suffices here to show that

$$a = \theta^{-1}(n) - \theta^{-1}(m) \tag{52}$$

is independent of the integers n and m used to represent $\theta(a) = n/m$. To do this let n and m be the unique pair [10] of relatively prime integers such that $\theta(a) = n/m$. Then any other such representation, such as $\theta(a) = p/q$, satisfies the equalities

$$p = kn \tag{53}$$

and

$$q = km \tag{54}$$

with k in N^1 ; hence

$$\begin{aligned} \theta^{-1}(p) - \theta^{-1}(q) &= \theta^{-1}(kn) - \theta^{-1}(km) \\ &= \theta^{-1}(k) + \theta^{-1}(n) - \theta^{-1}(k) - \theta^{-1}(m) \\ &= \theta^{-1}(n) - \theta^{-1}(m) \quad . \end{aligned} \tag{55}$$

The vector a recovered is therefore independent of the particular integers p/q used to represent $\theta(a)$; hence it is reasonable to store two separate integers from which common factors need not be removed before recovering a .

The preceding vector representations can be extended to form a matrix representation in a number of ways. Possibly the most convenient is to simply "unfold" a p by r matrix to form a pr vector and apply the preceding representations directly. This, unfortunately, is not algebraically well behaved and results in extremely large integer representations, which, as will be shown in the sequel, is the primary limitation on the application of the scheme. We therefore prefer to apply the representation separately to each column vector of a matrix separately, thereby yielding a representation for a matrix as a row vector of column representations. That is for a matrix, M , we have

$$\theta(M) = \text{row}[\theta(m_1), \theta(m_2), \ldots, \theta(m_r)] = \text{row}\theta(m_i) \qquad (56)$$

where m_i is the ith column vector of the matrix M . So defined, θ is a natural extension of the preceding vector representation since if M is composed of a single column vector, the single representing rational number of equation (56) is the same as that of equation (50). Also the linearity of θ is preserved via the formula

$$\theta(M+N) = \text{row}[\theta(m_i)\theta(n_i)] \qquad (57)$$

and the one to one, onto properties of θ follow from the corresponding properties for the individual column vectors. As before there is no limitation on the size of the matrices employed; hence we deal with infinite matrices that are zero almost everywhere, this group being denoted by W . Unlike the previous development, our representation is a vector of positive rational numbers, rather than a single number. Since our matrices are zero almost everywhere, the representing vectors are one almost everywhere. We therefore denote by S^+ the space of infinite vectors with positive rational entries, which are one almost everywhere. Clearly this space is a group under componentwise multiplication for which we have the following representation theorem.

3. Theorem. The function

$$\theta : W \longrightarrow S^+$$

$$M \longrightarrow \theta(M)$$

defined by equation (56) is a group isomorphism.

Since θ is linear, one can add matrices by carrying out a componentwise multiplication on the entries of their representations. Unfortunately a similar formula for matrix multiplication does not exist. We can, however, multiply the representation of a matrix by an unrepresented matrix to obtain the representation of its product. A little algebra will reveal that for matrices M and N

$$\theta(MN) = \text{row}\left[\prod_{j=1}^{r} \theta(m_j)\, n_{ji}\right] \qquad ; \tag{58}$$

hence matrix multiplication can be done in a sense.

Finally we note that one can carry out column operations directly on $\theta(M)$ for column interchange; multiplication of a column by a scalar and the addition of two columns can all be carried out directly in terms of the $\theta(m_i)$.

Recovery Algorithms

In the preceding development we have shown that, in theory, a can be uniquely recovered from $\theta(a)$. It, however, remains to obtain a computationally feasible algorithm for calculating a given $\theta(a)$. Clearly it suffices to consider the case where $\theta(a)$ is in N^1 and a is in V since the representations for a in U and a in W are just collections of representation of the former type. Our fundamental lemma in this respect is the following wherein the square bracket notation is used to denote the least integer operation [10].

4. Lemma. For any integer n and prime number p

$$\left[1 - n/p^k + [n/p^k]\right]$$

is one if p^k divides n and zero otherwise.

Proof: If p^k divides n , the n/p^k is an integer; hence

$$n/p^k = [n/p^k] \tag{59}$$

and

$$\left[1 - n/p^k + [n/p^k]\right] = [1] = 1 \tag{60}$$

On the other hand if p^k does not divide n ,

$$n/p^k - [n/p^k] \tag{61}$$

is a fraction between zero and one (exclusive); hence one minus this fraction is also such a fraction and therefore its greatest integer is zero as required. In essence the lemma says that the number $\left[1 - n/p^k + [n/p^k]\right]$ is one if n/p^k is an integer and zero if it is a fraction and is therefore readily evaluated computationally. Recognizing that the number $\theta(a)$ (for a in V) has a factor $p_i^{a_i}$, the lemma leads immediately to the following recovery theorem.

5. Theorem.

$$\left[1 - \theta(a)/p_i^k + \left[\theta(a)/p_i^k\right]\right] = 1$$

if and only if $k \leq a_i$.

Operationally there are a number of ways in which theorem 5 can be applied. If one is dealing with possibly large integers, a_i, the most efficient approach is to search for the largest k such that $\left[1 - \theta(a)/p_i^k + \left[\theta(a)/p_i^k\right]\right] = 1$ (i.e. the largest k such that $\theta(a)/p_i^k$ is an integer). Assuming that the integers a_i are known to be bounded by 2^m, this search can be done in m steps, as follows.

6. Corollary. Let a in V be such that $a_i < 2^m$. Then a_i can be calculated from $\theta(a)$ in no more than m steps by means of the following algorithm.

a. Let $d = 2^{m-1}$; let $j = 1$.

b. If $\theta(a)/p_i^d$ is an integer $a_i \geq d$, go to c.

If $\theta(a)/p_i^d$ is not an integer $a_i < d$, go to d.

c. Let $d = d + 2^{m-j-1}$; let $j = j + 1$; go to b.

d. Let $d = d - 2^{m-j-1}$; let $j = j + 1$; go to b.

Clearly after m steps of the above algorithm the value of a_i has been "trapped" and the recovery is complete.

A second approach to the recovery process is well suited for the case when a_i is known to be small, as for instance in the storage of topological matrices. In this case a_i can be calculated by the following formula, which converges in $a_i + 1$ steps.

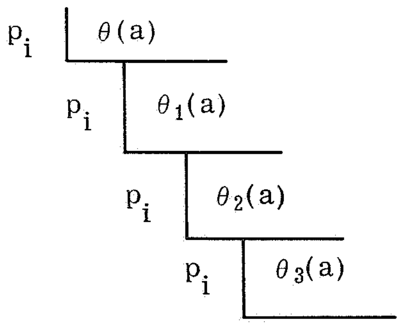
7. Corollary.

$$a_i = \sum_{k=1}^{\infty} \left[1 - \theta(a)/p_i^k + \left[\theta(a)/p_i^k\right] \right]$$

Here if a term is zero, all future terms will also be zero; hence the summation may cease when the first zero term is obtained after $a_i + 1$ steps.

A final algorithm also requires $a_i + 1$ steps and is essentially a variation of that if corollary 7 but requires the division of smaller integers and always divides by p_i rather than p_i^k .

8. Corollary. Let



represent a continued division algorithm. Then $a_i = k - 1$ such that $\theta_k(a)$ is the first quotient that is not an integer; at this point the algorithm terminates.

Clearly all of the above algorithms are predicated on our ability to determine whether or not $\theta(a)/p^k$ is an integer. As such $\theta(a)$ must be stored with sufficient accuracy to determine this fact if a is to be recovered exactly. Now any integral error in $\theta(a)$ certainly results in the change of some prime factors; hence the integer $\theta(a)$ must be known exactly. Of course since it is known a priori that $\theta(a)$ is an integer, one can accept fractional errors of less than half, but no integral errors are acceptable if a is to be recovered exactly. On a computer, therefore, $\theta(a)$ must be stored as an integer number; it does not suffice to store a few significant figures together with an exponent. The application of the data compression scheme is thus limited by the size of the integers $\theta(a)$ that can be stored on the available computer. Of course multiple precision words can be used but even then the number $\theta(a)$ may become too large for the machine unless the vector a is in some way restricted. In the particular case of the vectors resulting from topological matrices a is sparse; hence many of the factors in $\theta(a)$ are unity and are composed of small integers; hence the prime factors of $\theta(a)$ are raised to small powers only. These two conditions thus tend to keep $\theta(a)$ within reasonable bounds and render the procedure practicable for such matrices. Note that although the sparseness of the topological matrices is necessary to render the data compression scheme feasible, one does not need to assume a specified pattern of sparseness and may therefore manipulate compressed topological matrices without regard for variations in the sparseness pattern so long as the magnitude of $\theta(a)$ remains within reasonable bounds.

Conclusions

To implement the preceding data compression scheme one needs for the encoding process an ordered table of primes and for the decoding process a subroutine for evaluating $\left[1 - \theta(a)/p^k + \left[\theta(a)/p^k\right]\right]$ (that is an algorithm for determining whether or not $\theta(a)/p^k$ is an integer). In the former case one can, of course, simply tabulate a sufficiently long list of primes, but a more efficient approach is to use a polynomial approximation. That is a polynomial $P(x)$ such that

$$p_i \leq P(i) < p_i + 1 \tag{62}$$

which yields p_i exactly upon taking the integer part of $P(i)$. Since the primes are a reasonably smooth monotonic function of the integers, a relatively low order polynomial $P(x)$ can be used to calculate, exactly, a large number of primes. In the case of the decoder one can, of course, calculate $\theta(a)/p^k$ exactly prior to determining whether or not it is an integer, but since one is not really interested in the value of $\theta(a)/p^k$, a simpler subroutine is possible wherein the calculation of $\theta(a)/p^k$ is carried only far enough to determine whether or not it is an integer.

REFERENCES

1. Fowler, M. E.: A New Numerical Method for Simulation. Simulation. vol. 4, no. 5, May 1965, pp. 324-330.

2. Calahan, D. A.; and Abbott, N. E.: Stability Analysis of Numerical Integration, Proceedings of the 10th Midwest Symposium on Circuit Theory. Purdue Univ., 1967.

3. Yosida, K.: Functional Analysis. Springer-Verlag, New York, 1968.

4. Youla, D. C.: The Synthesis of Networks Containing Lumped and Distributed Elements. Proceedings of the Symposium on Generalized Networks. Polytechnic Institute of Brooklyn, Polytecnic Press, Brooklyn, N. Y., April, 1966.

5. Youla, D. C.: The Synthesis of Linear Dynamical Systems from Prescribed Weighting Patterns, SIAM Journal on Applied Mathematics, vol. 14, no. 3, May 1966, pp. 527-549.

6. Ho, B. L.: Effective Construction of Linear State-Variable Models from Input-Output Data. Proceedings of the Third Allerton Conference on Circuit and System Theory. Univ. of Ill., 1965.

7. Ho, B. L.: An Effective Construction of Realizations from Input-Output Discriptions. Ph.D. Thesis, Stanford Univ., 1965.

8. Kalman, R. E.; Falb, P. L.; and Arbib, M. A.: Topics in Mathematical System Theory. McGraw-Hill, New York, 1969.

9. Rice, J. R.: The Approximation of Functions. Addison-Wesley, Reading, Mass., 1962-1969.

10. McCoy, N. H.: The Theory of Numbers. MacMillan, New York, 1965.

APPROVAL

TM X-53868

STUDIES IN SYSTEM SIMULATION

By R. Saeks

The information in this report has been reviewed for security classification. Review of any information concerning Department of Defense or Atomic Energy Commission programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

This document has also been reviewed and approved for technical accuracy.

Carl C Prince

For H. HOELZER
Director, Computation Laboratory

DISTRIBUTION

INTERNAL

DIR
Dr. von Braun

AD-S
Dr. Stuhlinger

DEP-T
Dr. E. Rees

S&E-DIR
Mr. H. Weidner

PD-DIR
Dr. W. R. Lucas
Dr. W. A. Mrazek
Mr. F. L. Williams

S&E-COMP-DIR
Dr. H. Hoelzer
Mr. Carl Prince

S&E-COMP-C
Dr. H. Kerner

S&E-COMP-CS
Dr. H. Trauboth (10)
Mr. D. Marion

S&E-COMP-CR
Mr. A. Dean

S&E-COMP-S
Dr. Polstorff
Mr. R. Lawrence

S&E-ME-DIR
Dr. M. Siebel
Mr. Wuenscher

S&E-AERO-DIR
Dr. E. Geissler
Mr. H. J. Horn
Dr. D. L. Teuber

S&E-CSE-DIR
Dr. W. Haeussermann

S&E-ASTR-DIR
Mr. F. B. Moore
Mr. J. Lucas
Mr. F. S. Wojtalik
Mr. C. N. Swearingen
Mr. H. H. Hosenthien

S&E-ASTN-DIR
Mr. K. L. Heimburg
Dr. R. Head

S&E-SSL-DIR
Mr. G. B. Heller

S&E-QUAL-DIR
Mr. Grau

S&E-R-DIR
Dr. W. G. Johnson

A&TS-MS-IL (8)

A&TS-MS-IP (2)

A&TS-MS-H

A&TS-TU (6)

A&TS-PAT
Mr. L. D. Wofford, Jr.

DISTRIBUTION (Concluded) TM X-53868

EXTERNAL

Dr. R. Barfield
Department of Mechanical Systems
Engineering
University of Alabama
University, Ala. 35486

Dr. R. Saeks (15)
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

Scientific and Technical Information
Facility (25)
P. O. Box 33
College Park, Maryland 20740
Attn: NASA Representative (S-AK/RKT)

National Aeronautics and Space Administration
Washington, D. C. 20546
Attn: Alfred Gessow, RR-2

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No. TM X-53868

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Studies in System Simulation

5. Report Date August 27, 1969

6. Performing Organization Code

7. Author(s) R. Saeks

8. Performing Organization Report No.

9. Performing Organization Name and Address
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

10. Work Unit No.

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama 35812

13. Type of Report and Period Covered
Technical Memorandum

14. Sponsoring Agency Code

15. Supplementary Notes
Dr. Saeks was a participant in the NASA-ASEE Summer Faculty Fellowship Program at the Marshall Space Flight Center.

16. Abstract

The results of three mathematical studies of the feasibility of various methods of large scale digital system simulation are reported. Specific studies deal with the effects of applying numerical techniques directly to the components of a system, approximation techniques for reducing the order of subsystems, and data compression techniques for the computer storage of the large topological matrices encountered in system simulation.

17. Key Words

18. Distribution Statement
STAR Announcement

19. Security Classif. (of this report) Unclassified

20. Security Classif. (of this page) Unclassified

21. No. of Pages 44

22. Price $ 3.00