



122
700
THS

ON THE NUMERICAL SOLUTION OF LINEAR
AND NONLINEAR SYSTEM MODELS

Thesis for the Degree of Ph. D.
MICHIGAN STATE UNIVERSITY
Dinkar Shanker Rao
1962

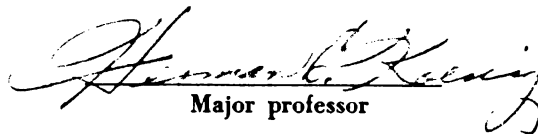
This is to certify that the
thesis entitled

ON THE NUMERICAL SOLUTION OF LINEAR AND NONLINEAR SYSTEM
MODELS
presented by

Dinkar S. Rane

has been accepted towards fulfillment
of the requirements for

Doctors degree in Electrical Engineering


Major professor

Date November 9, 1962

O-169



ABSTRACT

ON THE NUMERICAL SOLUTION OF LINEAR AND NONLINEAR
SYSTEM MODELS

Dinkar Shankar Rane

ON THE NUMERICAL SOLUTION OF
LINEAR AND NONLINEAR SYSTEM MODELS

by

Dinkar Shankar Rane

AN ABSTRACT OF A THESIS

Submitted to

Michigan State University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

Department of Electrical Engineering

1962

ON THE NUMERICAL SOLUTION OF LINEAR AND NONLINEAR SYSTEM
MODELS

Abstract of Thesis

by

Dinkar S. Rane

The general availability of digital computers has lead to the development of various numerical techniques for solving linear and nonlinear systems. The relative merit of each such technique varies from problem to problem. However, in all cases one of the primary considerations is computation time and numerical accuracy. The consideration of computation time has lead to the development of this thesis. The numerical accuracy has been increased in most cases or at least kept to the same degree as in the existing methods.

The mathematical model of the system used is assumed to be given in normal or standard form. Whenever components with nonlinear algebraic terminal equations are present, the mathematical model of the system may appear in a semi-normal form. The general mathematical model is then taken as

$$\begin{bmatrix} X \\ Y \\ 0 \end{bmatrix} = \begin{bmatrix} F_1(X,Y,Z) \\ F_2(X,Y,Z) \\ F_3(X,Y,Z) \end{bmatrix}$$

For linear system, i.e., when

$$\frac{d}{dt} X = AX + BE(t)$$

a recursive formula is developed for the numerical solution. This formula can meet any prescribed degree of accuracy without increasing the computation time per step. Theoretically the computation time is one-fourth that of the standard Runge-Kutta method.

Some mixed, linear and nonlinear forms are also considered. The Runge-Kutta method has been modified to increase the computational speed. The numerical accuracy has been increased in some cases while the original degree of accuracy has been unchanged in others. The overall effect is a faster numerical solution with a high degree of accuracy. The results are compared with the results by other existing methods such as the Runge-Kutta method, predictor-corrector methods and others.

Finally the semi-normal form of the mathematical model is considered and two procedures are described for obtaining a numerical solution. These procedures also yield a shorter computation time without a loss in accuracy.

Numerical examples are given for the purpose of comparing the actual relative computation time and numerical accuracy and indeed support the theoretical development.

ON THE NUMERICAL SOLUTION OF
LINEAR AND NONLINEAR SYSTEM MODELS

By

Dinkar Shankar Rane

A THESIS

Submitted to

Michigan State University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

Department of Electrical Engineering

1962

1-250817
6-12-66

ACKNOWLEDGEMENT

The author is grateful to Dr. H.E. Koenig, his thesis advisor and major professor, for his constant guidance during the preparation of this thesis and during the period of investigation preceding its completion and to Dr. R.J. Reid and Dr. G.P. Weeg for their helpful suggestions. He also wishes to thank Dr. L.W. VonTersch and Professor J.W. Donnell for their constant encouragement.

TABLE OF CONTENTS

Section		Page
1.	INTRODUCTION	1
2.	LINEAR SYSTEMS OF DIFFERENTIAL EQUATIONS.	3
3.	MIXED LINEAR AND NONLINEAR DIFFERENTIAL EQUATIONS.	13
4.	NONLINEAR ALGEBRAIC EQUATIONS.	36
5.	EXAMPLE SOLUTIONS	44
6.	CONCLUSION.	53

LIST OF TABLES

Table	Page
5.1 SOLUTION OF LINEAR SYSTEM OF EQUATIONS.....	46
5.2 SOLUTION OF MIXED LINEAR AND NONLINEAR EQUATIONS WITH CLOSED LINEAR SET.....	48
5.3 SOLUTION OF A CLASS OF LINEAR AND NONLINEAR SYSTEMS.....	52
5.4 SOLUTION OF ANOTHER CLASS OF LINEAR AND NONLINEAR SYSTEMS.....	55
5.5 SOLUTION OF SYSTEM MODEL WITH NONLINEAR ALGEBRAIC EQUATIONS.....	57

LIST OF APPENDICES

Appendix	Page
A.	
RECURSSIVE FORMULA FOR VARIOUS DEGREES	
OF ACCURACY.	61
B.	
RUNGE-KUTTA METHOD AND A PREDICTOR	
CORRECTOR METHOD.	42

I. INTRODUCTION

A prerequisite for any system design procedure, is some kind of mathematical model of the system. The model may be obtained by various transform techniques such as the Laplace transform, or in the time domain as a set of differential equations. The transform techniques have been applied very effectively in the area of linear systems but do not apply to nonlinear systems. The time-domain analysis, however, can be extended to include nonlinear systems in general.

A time-domain mathematical model of physical systems particularly suited to computer solutions is in the normal or standard form. General procedures for formulating mathematical models of physical systems in this form are given by Wirth⁽¹⁾.

The objective of this thesis is to develop high speed numerical methods for obtaining solutions to the type of mathematical model found in the analysis of physical systems. The general form of the mathematical model considered is the seminormal form

$$\frac{d}{dt} \begin{bmatrix} X \\ Y \\ 0 \end{bmatrix} = \begin{bmatrix} F_1(X,Y,Z) \\ F_2(X,Y,Z) \\ F_3(X,Y,Z) \end{bmatrix} + \begin{bmatrix} E_1 E_1(t) \\ E_2 E_2(t) \\ E_3 E_3(t) \end{bmatrix} \quad (1.1)$$

where X , Y , $E_1(t)$ and $E_3(t)$ are vectors; F_1 , F_2 , F_3 are vector functions and the last equation is an algebraic equation nonlinear in Z .

The problem, then is to develop a method for obtaining a numerical solution of nonlinear differential and algebraic equations.

Various methods have been suggested for obtaining the numerical solutions to such problems.^(2,3,4) The most common of these methods are Runge-Kutta, Numerov's and the various predictor-corrector schemes, such as those of Nystrom, Milne, Hartree etc. All these methods are compared with the procedures of this thesis.

In this thesis special forms of the general mathematical model are considered. In section 2, linear differential systems are considered and a recursive formula is developed for the solution. The solution at nth interval of time in terms of initial conditions is shown to have a form resembling that of the nth state vector in Markov Processes.

In section 3, combinations of linear and nonlinear differential systems are considered. A procedure applicable to a special class of nonlinearity is considered first with subsequent extension to a more general form.

In section 4, the procedure is extended to certain classes of mathematical models involving both nonlinear algebraic and differential equations.

II. LINEAR SYSTEMS OF DIFFERENTIAL EQUATIONS

Let the mathematical model of a linear system be of the form

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1k} \\ a_{21} & a_{22} & \dots & a_{2k} \\ \vdots & \vdots & & \vdots \\ a_{k1} & a_{k2} & \dots & a_{kk} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1m} \\ b_{21} & b_{22} & \dots & b_{2m} \\ \vdots & \vdots & & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nm} \end{bmatrix} \begin{bmatrix} e_1(t) \\ e_2(t) \\ \vdots \\ e_m(t) \end{bmatrix} \quad (2.1)$$

or in vector notation

$$\frac{d}{dt} [X(t)] = [A] [X(t)] + [B] [E(t)] \quad (2.2)$$

The solution to (2.1) for the vectors $X(h)$, $X(2h)$... $X(nh)$... is obtained first as a Taylor's series expansion, and then this solution is put in a concise form to obtain a recursion formula.

2.1 Taylor's Series Expansion

An analytic function $f(t+h)$, where h is a constant, can be expanded in terms of $f(t)$ and its successive derivatives with respect to t as

$$f(t+h) = f(t) + h f'(t) + \frac{1}{2} h^2 f''(t) + \dots + \frac{1}{n!} h^n f^{(n)}(t) + \dots \quad (2.3)$$

Consequently using vector notation, (2.3) can be expanded as

$$X(t+h) = X(t) + h \frac{d}{dt}X(t) + \frac{h^2}{2} \frac{d^2}{dt^2} X(t) + \dots + \frac{h^n}{n!} \frac{d^n}{dt^n} X(t) + \dots \quad (2.4)$$

where the indicated derivatives of the vector $X(t)$ as calculated from (2.2) are

$$\frac{d}{dt} X(t) = A X(t) + BE(t) \quad (2.5.1)$$

$$\begin{aligned} \frac{d^2}{dt^2} X(t) &= A \frac{d}{dt} X(t) + BE'(t) \\ &= A^2 X(t) + ABE(t) + BE'(t) \end{aligned} \quad (2.5.2)$$

$$\begin{aligned} \frac{d^3}{dt^3} X(t) &= A^2 [AX(t) + BE(t)] + ABE'(t) + BE''(t) \\ &= A^3 X(t) + A^2 BE(t) + ABE'(t) + BE''(t) \end{aligned} \quad (2.5.3)$$

$$\begin{aligned} \frac{d^4}{dt^4} X(t) &= A^3 [AX(t) + BE(t)] + A^2 BE'(t) + ABE''(t) + BE'''(t) \\ &= A^4 X(t) + A^3 BE(t) + A^2 BE'(t) + ABE''(t) + BE'''(t) \end{aligned} \quad (2.5.4)$$

In equations (2.5.1) through (2.5.4) the derivatives of $E(t)$ can be obtained approximately in terms of $E(t)$, $E(t + \frac{h}{2})$ etc., by using the forward differences. The third and higher derivatives involve $E(t + \frac{3h}{2})$, $E(t + 2h)$ etc. However, an approximation is made for this, so that all the derivatives are expressed in terms of $E(t)$, $E(t + \frac{h}{2})$, $E(t + h)$.

$$\begin{aligned} E'(t) &= \frac{d}{dt} E(t) \simeq \frac{E(t + \frac{1}{2} h) - E(t)}{\frac{1}{2} h} \\ &= \frac{2}{h} [E(t + \frac{1}{2} h) - E(t)] \end{aligned} \quad (2.6)$$

$$\begin{aligned} E''(t) &= \frac{d}{dt} E'(t) \simeq \frac{2}{h} [E'(t + \frac{h}{2}) - E'(t)] \\ &= \frac{4}{h^2} [E(t + h) - 2 E(t + \frac{1}{2} h) + E(t)] \end{aligned} \quad (2.7)$$

$$\begin{aligned} E'''(t) &\simeq \frac{2}{h} [E''(t + \frac{h}{2}) - E''(t)] \\ &\simeq \frac{8}{h^3} [E(t + \frac{3h}{2}) - 3 E(t + h) + 3 E(t + \frac{h}{2}) - E(t)] \end{aligned} \quad (2.8)$$

Using quadratic extrapolation $E(t + \frac{3h}{2})$ is approximated in terms of $E(t + h)$, $E(t + \frac{h}{2})$ and $E(t)$, so that

$$E(t + \frac{3h}{2}) \simeq \frac{5}{2} E(t + h) - 2 E(t + \frac{h}{2}) + \frac{1}{2} E(t) \quad (2.9)$$

Equation (2.7) then becomes

$$E'''(t) \approx \frac{3}{h^3} \left[-\frac{1}{2} E(t+h) + E\left(t + \frac{h}{2}\right) - \frac{1}{2} E(t) \right] \quad (2.10)$$

Substitution of (2.6) through (2.10) and (2.5.1) through (2.5.4) in (2.4) gives

$$\begin{aligned} X(t+h) &\approx [U + hA + \frac{1}{2} h^2 A^2 + \frac{1}{6} h^3 A^3 + \frac{1}{24} h^4 A^4] X(t) \\ &\quad + h \left[\frac{1}{2} U + \frac{1}{3} hA + \frac{1}{12} h^2 A^2 + \frac{1}{24} h^3 A^3 \right] EX(t) \\ &\quad + h \left[\frac{1}{12} h^2 A^2 \right] EX\left(t + \frac{h}{2}\right) + h \left[\frac{1}{2} U + \frac{1}{6} hA \right] EX(t+h) \end{aligned} \quad (2.11)$$

or

$$X(t+h) \approx [\alpha] X(t) + [\beta_1] E(t) + [\beta_2] E\left(t + \frac{1}{2} h\right) + [\beta_3] E(t+h) \quad (2.12)$$

Where

$$[\alpha] = [U] + h[A] + \frac{1}{2} h^2 [A]^2 + \frac{1}{6} h^3 [A]^3 + \frac{1}{24} h^4 [A]^4 \quad (2.13)$$

$$[\beta_1] = h \left[\frac{1}{2} [U] + \frac{1}{3} h [A] + \frac{1}{12} h^2 [A]^2 + \frac{1}{24} h^3 [A]^3 \right] [B] \quad (2.14)$$

$$[\beta_2] = h \frac{1}{12} h^2 [A]^2 [B] \quad (2.15)$$

$$[\beta_3] = h \left[\frac{1}{2} [U] + \frac{1}{6} h [A] \right] [B] \quad (2.16)$$

If $X(nh)$ is denoted as $X(n)$ and $X(nh + h)$ as $X(n + 1)$, then at the discrete points, $t = nh$, (2.12) becomes

$$X(n+1) = [\alpha] X(n) + \beta_1 E(n) + \beta_2 E(n + \frac{1}{2}) + \beta_3 E(n+1) \quad (2.17)$$

Equation (2.17) is a recursion formula for a solution to the system of equations (2.2) i.e.

$$X(1) = [\alpha] X(0) + \beta_1 E(0) + \beta_2 E(\frac{1}{2}) + \beta_3 E(1) \quad (2.18.1)$$

$$X(2) = [\alpha] X(1) + \beta_1 E(1) + \beta_2 E(\frac{3}{2}) + \beta_3 E(2) \quad (2.18.2)$$

and so on.

Note that the matrix $[\alpha]$ is of order $n \times n$ and $[\beta_1]$, $[\beta_2]$, and $[\beta_3]$ are of order $n \times m$. These matrices are calculated once for all at the beginning.

In many practical problems the input signals are all of the same general form, i.e.,

$$EE(t) = \begin{bmatrix} e_1(t) \\ \vdots \\ e_k(t) \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_k \end{bmatrix} e(t) = [E] e(t) \quad (2.19)$$

so that

$$[\rho_1] E(n) = [\beta_1] [E] e(n)$$

If we let

$$[\beta_1] [E] = \lambda_1 \quad (2.20.1)$$

$$[\beta_2] [E] = \lambda_2 \quad (2.20.2)$$

$$[\beta_3] [E] = \lambda_3 \quad (2.20.3)$$

then the recursion relation in (2.17) reduces to

$$X(n+1) = [\alpha] X(n) + \lambda_1 e(n) + \lambda_2 e(n + \frac{1}{2}) + \lambda_3 e(n+1) \quad (2.21)$$

where $\lambda_1, \lambda_2, \lambda_3$ are column matrices.

2.2 Comparison with Other Solution Procedures

There are many numerical procedures available for a solution to a set of linear differential equations. Most common and widely used of these is the Runge-Kutta method. The predictor-corrector methods of Milne, Hartree and Nystrom and others, are also used for special forms of linear differential equations. However, the procedure given above gives a much faster solution and the accuracy can be maintained above any desired degree. The degree of accuracy is a function only of the expressions for $[\alpha], [\beta_1], [\beta_2], [\beta_3]$ as shown in Appendix A.

The recursive formula (2.12) remains unchanged. Thus, regardless of the degree of accuracy the computation time is the same. Actual computer programming indeed proves this point.

Inspection of the widely used Runge-Kutta method shows that for a system with n variables and m driving functions, $(4n^2 + 4nm + 5n)$ multiplications and $(4n^2 + 11n)$ additions are required. On the other hand, in the procedure of equation (2.17), $(n^2 + 3nm)$ multiplications and $(n^2 + 3nm)$ additions are required. This gives a saving of $(3n^2 + nm + 5n)$ multiplications and $(3n^2 + nm + 11n)$ additions. This represents a relative time factor of approximately 4 to 1 for relatively small m . Even for the largest value of m , namely $m = n$, the time factor is better than 2 to 1. In practice, however, this advantage is reduced slightly because of the time required to print out the intermediate results.

The bookkeeping required in the procedure given above is also considerably less as compared to that in the Runge-Kutta method. In Runge-Kutta method the solution at the n th interval must be retained. At the same time, the values of the matrices K_1, K_2, K_3, K_4 as defined in Appendix C are also to be stored. The evaluation of K_1, K_2, K_3, K_4 is not symmetric, hence this requires additional bookkeeping. On the other hand in the solution given above it is necessary to store only the solution at n^{th} interval and the coefficient matrices $[\alpha] [\beta_1] [\beta_2] [\beta_3]$ once calculated are used throughout the solution of a given problem.

The comparison of the procedure given above with the various predictor corrector methods also shows an advantage. The formulas for a typical predictor-corrector method suggested by Hamming are given in Appendix B. A careful study of these formulas shows that the first and second derivatives of the variables have to be evaluated twice for each step, thereby increasing the computation time considerably over that required in the above procedure. A further study indicates that the value of $X(t)$, $X'(t)$ and $X''(t)$ at the points $(n-1)$ and n are required to evaluate the solution at $(n+1)$. This represents a substantial amount of bookkeeping hence puts a limit on the size of problem that can be solved. Another disadvantage in the method of Hamming as compared to that given above is the use of the second derivatives while the system of equations (2.2) involves only the first derivatives.

2.3 State Vector and State Matrix

The recursive formula (2.17) expresses the solution at any interval, in terms of the solution at the preceeding interval and the driving function at all the previous intervals and mid-intervals. By recursive substitution, formula (2.17) can be used to develop an expression for the solution at any interval in terms of only the initial conditions on the dependent variables and the driving function at all the previous intervals and mid-intervals. Such an expression is very convenient to use in many cases. Moreover, in the special case of constant input, this expression has a very interesting form. Even in the general case of time varying driving functions the modified expression has a striking similarity with expression in the Value-Determination operation of Markov Process.

The recursive formula 2.17 can be rewritten in detail at the intervals $h, 2h, \dots nh$ as:

$$X(1) = \alpha X(0) + \beta_1 E(0) + \beta_2 E(\frac{1}{2}) + \beta_3 E(1) \quad (2.22)$$

$$\begin{aligned} X(2) &= \alpha X(1) + \beta_1 E(1) + \beta_2 E(1\frac{1}{2}) + \beta_3 E(2) \\ &= \alpha^2 X(0) + \alpha [\beta_1 E(0) + \beta_2 E(\frac{1}{2}) + \beta_3 E(1)] \\ &\quad + [\beta_1 E(1) + \beta_2 E(1\frac{1}{2}) + \beta_3 E(2)] \end{aligned} \quad (2.23)$$

$$\begin{aligned} X(3) &= \alpha X(2) + \beta_1 E(2) + \beta_2 E(2\frac{1}{2}) + \beta_3 E(3) \\ &= \alpha^3 X(0) + \alpha^2 [\beta_1 E(0) + \beta_2 E(\frac{1}{2}) + \beta_3 E(1)] \\ &\quad + \alpha [\beta_1 E(1) + \beta_2 E(1\frac{1}{2}) + \beta_3 E(2)] \\ &\quad + [\beta_1 E(2) + \beta_2 E(2\frac{1}{2}) + \beta_3 E(3)] \end{aligned} \quad (2.24)$$

$$\begin{aligned}
 X(n) &= \alpha X(n-1) + \beta_1 E(n-1) + \beta_2 E(n-1 + \frac{1}{2}) + \beta_3 E(n) \\
 &= \alpha^n X(0) + \alpha^{n-1} [\beta_1 E(0) + \beta_2 E(\frac{1}{2}) + \beta_3 E(1)] \\
 &\quad + \alpha^{n-2} [\beta_1 E(1) + \beta_2 E(1\frac{1}{2}) + \beta_3 E(2)] \\
 &\quad \vdots \\
 &\quad + \alpha [\beta_1 E(n-2) + \beta_2 E(n-2\frac{1}{2}) + \beta_3 E(n-1)] \\
 &\quad + [\beta_1 E(n-1) + \beta_2 E(n-1\frac{1}{2}) + \beta_3 E(n)] \quad (2.25)
 \end{aligned}$$

or

$$X(n) = \alpha^n X(0) + \sum_{j=0}^{n-1} \alpha^{n-j-1} [\beta_1 E(j) + \beta_2 E(j + \frac{1}{2}) + \beta_3 E(j+1)] \quad (2.26)$$

For a driving function of constant amplitude

$$E(j) = E(j + \frac{1}{2}) = E(j+1) = E \quad (2.27)$$

If we let

$$\beta_1 + \beta_2 + \beta_3 = \beta.$$

then

$$X(n) = \alpha^n X(0) + \sum_{k=0}^{n-1} \alpha^k \beta E \quad (2.28)$$

Comparing this to the value-determination formula in the Markov process, namely

$$V(n) = P^n V(o) + \sum_{j=0}^{n-1} P^j Q \quad (2.29)$$

We see that they are identical in form.

If for time varying driving functions we let

$$\bar{E}(j) = \beta_1 E(j) + \beta_2 E(j + \frac{1}{2}) + \beta_3 E(j + 1) \quad (2.30)$$

then equation (2.20) becomes

$$X(n) = \alpha^n X(o) + \sum_{j=0}^{n-1} \alpha^{n-j-1} \bar{E}(j) \quad (2.31)$$

Equations (2.20) and (2.31) indicate that the linear mathematical model of a control system given by (2.2) can be put in the same form as the mathematical models of Markov Processes. Thus, the optimization procedures using dynamic programming given by Howard⁽⁶⁾ can be applied directly to deterministic systems. The possibility of exploiting these techniques through this type of recursion formula represents an area worthy of further investigation.

In many control systems applications the only information required is the output signals as the function of the input signals. In other words, only the response of the system is of interest. This is also the case when numerical solution to a higher order differential equation is required. Let the differential equation be

$$\frac{d^n x}{dt^n} + C_{n-1} \frac{d^{n-1} x}{dt^{n-1}} + C_{n-2} \frac{d^{n-2} x}{dt^{n-2}} + \dots + C_1 \frac{dx}{dt} + C_0 x + f(t) = 0 \quad (2.32)$$

and let

$$\frac{dx}{dt} = x_1, \quad \frac{d^2 x}{dt^2} = x_2 \dots \quad \frac{d^{n-1} x}{dt^{n-1}} = x_{n-1} \quad (2.33)$$

Equation (2.32) when written in normal form then becomes

$$\frac{d}{dt} \begin{bmatrix} x \\ x_1 \\ x_2 \\ \vdots \\ x_{n-2} \\ x_{n-1} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{n-1} \\ -c_0 x - c_1 x_1 \dots - c_{n-1} x_{n-1} - f(t) \end{bmatrix}$$

$$= \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 1 \\ c_0 & -c_1 & -c_2 & \dots & -c_{n-2} & -c_{n-1} \end{bmatrix} \begin{bmatrix} x \\ x_1 \\ x_2 \\ \vdots \\ x_{n-2} \\ x_{n-1} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ -f(t) \end{bmatrix} \quad (2.34)$$

In such a case only the solution for x is of interest. In control systems, the response can be obtained conveniently from the earlier results and particularly from equation (2.31). Let this set of equations be arranged so that the output variables $X_o(t)$ are the first- k elements of the state vector $X(t)$ and let $R(n)$ denote the first k -rows of $[\alpha]^n$. Then from equation 2.31 the expression for $X_o(n)$ is

$$X_o(n) = R(n) X(o) + \sum_{j=0}^{n-1} R(n-j-1) \bar{E}(j) \quad (2.35)$$

The expression for the output variables X_o at $t = h, 2h, \dots, nh$ can be written as

$$\begin{bmatrix} X_o(1) \\ X_o(2) \\ X_o(3) \\ \vdots \\ X_o(n-1) \\ X_o(n) \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ R(3) \\ \vdots \\ R(n-1) \\ R(n) \end{bmatrix} X(o) + \begin{bmatrix} R(o) & 0 & \dots & 0 & 0 \\ R(1) & R(o) & \dots & 0 & 0 \\ R(2) & R(1) & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ R(n-2) & R(n-3) & & R(o) & 0 \\ R(n-1) & R(n-2) & & R(1) & R(o) \end{bmatrix} \begin{bmatrix} \bar{E}(o) \\ \bar{E}(1) \\ . \\ \vdots \\ \bar{E}(n-2) \\ \bar{E}(n-1) \end{bmatrix} \quad (2.36)$$

In the special case where there is only one input signal, as for example, when the control system is driven from a digital computer, $R(n)$ represents the first k rows of the matrix $[\alpha]^n$, i.e., $R(o) =$

$$\begin{bmatrix} 1, & 0, & 0, & \dots & 0 \\ 0, & 1, & 0, & \dots & 0 \\ \vdots & \vdots & \vdots & \dots & \vdots \end{bmatrix} \text{ and (2.36) becomes}$$

$$\begin{bmatrix} X_o(1) \\ X_o(2) \\ X_o(3) \\ \vdots \\ X_o(n-1) \\ X_o(n) \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ R(3) \\ \vdots \\ R(n-1) \\ R(n) \end{bmatrix} X(o) + \begin{bmatrix} S_o & 0 & 0 & \dots & 0 & 0 \\ S_1 & S_o & 0 & \dots & 0 & 0 \\ S_2 & S_1 & S_o & \dots & 0 & 0 \\ \vdots & \vdots & & & \vdots & \vdots \\ S_{n-2} & S_{n-3} & S_{n-2} & \dots & S_o & 0 \\ S_{n-1} & S_{n-2} & S_{n-3} & \dots & S_1 & S_o \end{bmatrix} \begin{bmatrix} \bar{e}_1(o) \\ \bar{e}_1(o) \\ \bar{e}_1(2) \\ \vdots \\ \bar{e}_1(n-1) \end{bmatrix}$$

(2.37)

where S_j is a column matrix.

Equation (2.34) can also be rewritten as the vector

$$[X_o] = [R] [X(o)] + [S] [\bar{E}_1] \quad (2.35)$$

Where $[S]$ is lower triangular in the submatrices $S_{ij} = S_{i+1, j+1}$.

Equation (2.35) is a particularly convenient model to use in the analysis of discrete state systems such as sampled data control systems.

In such a system, the vectors of discrete magnitudes $c_i(j)$, $j = 1, 2, \dots$ represent the discrete output levels of the digital computer.

III. MIXED LINEAR AND NONLINEAR DIFFERENTIAL EQUATIONS

The vast majority of physical systems of practical interest in engineering contain both linear and nonlinear components, i.e., the system seldom contains all nonlinear components. When such is the case it is sometimes possible to select the variables (i.e. a formulation tree in the system graph) so that the normal form model appears in the form

$$\frac{d}{dt} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} F_1(X_1, X_2, t) \\ F_2(X_1, X_2, t) \end{bmatrix} \quad (3.1)$$

Where F_2 is a linear vector function of the form

$$F_2(X_1, X_2, t) = A X_2 + D E(t) \quad (3.2)$$

and F_1 is nonlinear vector function. For such a system a substantial saving in computer time can be realized by combining the procedure given in section 2 with other known procedures.

3.1 Modified Runge-Kutta Solution for Mixed Linear and Nonlinear Systems With Closed Linear Sets

Solution to the second set of (3.1) can be obtained independent of the first set. Let it be obtained by the procedure of section 2.

The modification of the Runge-Kutta method required to use this procedure is explained by first examining the standard Runge-Kutta method.

The standard Runge-Kutta procedure for obtaining a solution to (3.1) is

$$X_1(n+1) = X_1(n) + \frac{1}{6} [K_1 + 2K_2 + 2K_3 + K_4] \quad (3.3)$$

$$X_2(n+1) = X_2(n) + \frac{1}{6} [L_1 + 2L_2 + 2L_3 + L_4] \quad (3.4)$$

where:

$$K_1 = h F_1 [X_1(n), X_2(n), nh] \quad (3.5)$$

$$L_1 = h F_2 [X_2(n), nh] \quad (3.6)$$

$$K_2 = h F_1 [X_1(n) + \frac{1}{2} K_1, X_2(n) + \frac{1}{2} L_1, nh + \frac{1}{2} h] \quad (3.7)$$

$$L_2 = h F_2 [X_2(n) + \frac{1}{2} L_1, nh + \frac{1}{2} h] \quad (3.8)$$

$$K_3 = h F_1 [X_1(n) + \frac{1}{2} K_2, X_2(n) + \frac{1}{2} L_2, nh + \frac{1}{2} h] \quad (3.9)$$

$$L_3 = h F_2 [X_2(n) + \frac{1}{2} L_2, nh + \frac{1}{2} h] \quad (3.10)$$

$$K_4 = h F_1 [X_1(n) + K_3, X_2(n) + L_3, nh + h] \quad (3.11)$$

$$L_4 = h F_2 [X_2(n) + L_3, nh + h] \quad (3.12)$$

It can be seen that in order to obtain a solution to the first set, the quantities L_1, L_2, L_3, L_4 must be obtained from the second set. However, if the recursive formula of section 2 is used to obtain a solution to the second set, the quantities L_1, L_2, L_3, L_4 are not available.

To show how the Runge-Kutta method can be conveniently modified so as to avoid this difficulty, suppose a solution to the second set has been obtained by a recursive formula of section 2 at the points $t = h, 2h, 3h, \dots, nh, (n+1)h$ and $t = \frac{1}{2}h, \frac{3}{2}h, \frac{5}{2}h, \dots, (\frac{2n+1}{2})h$.

Consider now the recursion formula

$$X_1(n+1) = X_1(n) + \frac{1}{6} [K'_1 + 2 K'_2 + 2 K'_3 + K'_4] \quad (3.13)$$

where

$$K'_1 = h F_1 [X_1(n), X_2(n), nh] \quad (3.14)$$

$$K'_2 = h F_1 [X_1(n) + \frac{K'_1}{2}, X_2(n + \frac{1}{2}), nh + \frac{h}{2}] \quad (3.15)$$

$$K'_3 = h F_1 [X_1(n) + \frac{K'_2}{2}, X_2(n + \frac{1}{2}), nh + \frac{h}{2}] \quad (3.16)$$

$$K'_4 = h F_1 [X_1(n) + K'_3, X_2(n+1), nh + h] \quad (3.17)$$

It can be noted that equations (3.13) through (3.17) are similar to equations (3.4) through (3.8) except for the following changes.

$$\begin{aligned} X_2(n) + \frac{1}{2} L_1 & \text{ is replaced by } X_2(n + \frac{1}{2}) \\ X_2(n) + \frac{1}{2} L_2 & \text{ is replaced by } X_2(n + \frac{1}{2}) \\ X_2(n) + L_3 & \text{ is replaced by } X_2(n+1) \end{aligned}$$

The following discussion shows that with this modification there is no loss of accuracy yet that there is a saving in computation time.

Geometrically, the Runge-Kutta formula actually calculates the derivatives of $X(n)$ at points nh , $nh + \frac{1}{2}h$, $nh + h$. The solution at point $(n+1)$ is then calculated by adding to $X(n)$, the weighted average of the derivatives multiplied by the time interval h .

The modified procedure given above is exactly equivalent to this geometric interpretation. Mathematically, $F_1 [X_1(t), X_2(t), t]$ can be rewritten as $F[X_1(t), t]$ since X_2 is also a time varying function and is known in terms of t . Hence

$$K'_1 = h F_1 [X_1(n), X_2(n), nh] = K_1 \quad (3.18)$$

$$\begin{aligned} K'_2 &= h F_1 \left[X_1(n) + \frac{K'_1}{2}, X_2\left(n + \frac{1}{2}\right), nh + \frac{h}{2} \right] \\ &= h F_1 \left[X_1(n) + \frac{K'_1}{2}, nh + \frac{h}{2} \right] \\ &\cong K_2 \end{aligned} \quad (3.19)$$

$$\begin{aligned} K'_3 &= h F_1 \left[X_1(n) + \frac{K'_2}{2}, X_2\left(n + \frac{1}{2}\right), nh + \frac{h}{2} \right] \\ &= h F_1 \left[X_1(n) + \frac{1}{2} K'_2, nh + \frac{h}{2} \right] \\ &\cong K_3 \end{aligned} \quad (3.20)$$

$$\begin{aligned} K'_4 &= h F_1 [X_1(n) + K'_3, X_2(n+1), nh + h] \\ &= h F_1 [X_1(n) + K'_3, nh + h] \cong K_4 \end{aligned} \quad (3.21)$$

If the number of equations in the nonlinear and linear sets are j and k , respectively, the computation time by the standard Runge-Kutta method is $4(j+k)$ units while by the procedure described, it is $(4j+2k)$ units.

An alternate way of modifying the Runge-Kutta method to realize even a larger time saving is to obtain a numerical solution to the second set (linear set) by the recursive formula of section 2, at time $h, 2h, 3h \dots nh (n+1) h$. Solution to the first set is then obtained as follows.

Since

$X_2(n+1), X_2(n)$ are known, calculate the average

$$\bar{X}_2 = \frac{1}{2} [X_2(n) + X_2(n+1)] \quad (3.22)$$

Consider the recursion formula

$$X_1(n+1) = X_1(n) + \frac{1}{6} [K_1'' + 2 K_2'' + 2 K_3'' + K_4''] \quad (3.23)$$

where

$$K_1'' = h F_1 [X_1(n), X_2(n), nh] \quad (3.24)$$

$$K_2'' = h F_1 [X_1(n) + \frac{1}{2} K_1'', \bar{X}_2, nh + \frac{h}{2}] \quad (3.25)$$

$$K_3'' = h F_1 [X_1(n) + \frac{1}{2} K_2'', \bar{X}_2, nh + \frac{h}{2}] \quad (3.26)$$

$$K_4'' = h F_1 [X_1(n) + K_3'', X_2(n+1), nh + h] \quad (3.27)$$

Equations (3.23) through (3.27) are similar to equations (3.4) through (3.8) except for the following changes

$$X_2 + \frac{1}{2} L_1 \text{ is replaced by } \bar{X}_2$$

$$X_2 + \frac{1}{2} L_2 \text{ is replaced by } \bar{X}_2$$

$$X_2 + L_3 \text{ is replaced by } X_2(n+1)$$

The following discussion shows that the procedure described has at least the same degree of accuracy as the standard Runge-Kutta method and that there is again a saving in computation time. The first part of the statement is again based on the geometrical interpretation of the Runge-Kutta method. The value K_1 as calculated by equation (3.5) is an approximation of the first derivative of X at nh when K_2 , K_3 are multiplied by h as shown in equation (3.6) and (3.7) and the result is an approximation to the derivatives of X at $h + \frac{1}{2} h$. However, the solution is **not** known at $nh + \frac{1}{2}h$ but is approximated by $X_2(n) + \frac{1}{2} L_1$. In the above procedure this approximation is taken as $\frac{1}{2} [X_2(n) + X_2(n+1)]$. This is, in fact a much better approximation than the one in the standard Runge-Kutta method.

Mathematically we have

$$\bar{X}_2 = \frac{1}{2} [X_2(n) + X_2(n+1)]$$

but from the recursion formula (2.17)

$$X_2(n+1) = \alpha X_2(n) + \beta_1 e(n) + \beta_2 e(n + \frac{1}{2}) + \beta_3 e(n + 1) \quad (3.23)$$

$$\begin{aligned} \bar{X}_2 &= \frac{1}{2} [X_2(n) + X_2(n) + (\alpha-u) X_2(n) + \beta_1 e(n) + \beta_2 e(n + \frac{1}{2}) \\ &\quad + \beta_3 e(n+1)] \\ &= X_2(n) + \frac{1}{2} [\alpha-u] X_2(n) + \frac{1}{2} [\beta_1 e(n) + \beta_2 e(n + \frac{1}{2}) + \beta_3 e(n+1)] \end{aligned} \quad (3.29)$$

Therefore

$$\begin{aligned} (\alpha-U) X_2(n) &= [U+hA + \frac{1}{2} h^2 A^2 + \frac{1}{6} h^3 A^3 + \frac{1}{24} h^4 A^4 -U] X_2(n) \\ &= h[A + \frac{1}{2} h A^2 + \frac{1}{6} h^2 A^2 + \frac{1}{24} h^3 A^4] X_2(n) \\ &\approx L_1 \end{aligned} \quad (3.30)$$

It is also to be noted that

$$(\alpha-U) X_2(n) \approx L_2 \quad (3.31)$$

Similarly it can be shown that

$$X_2(n+1) \approx X_2(n) + L_3 \quad (3.32)$$

If the numbers of equations in the nonlinear and linear set are again j and k , respectively, the computation time by standard Runge-Kutta method is $4(j+k)$ units. While by the second modified procedure given above is $(4j+k)$ units, or a saving of $3k$ units.

3.2 Modified Runge-Kutta Procedure for Mixed Linear and Nonlinear Systems.

In this section a more general set of mixed linear and nonlinear equations in the general form is considered. Let the mathematical model of the system be given as

$$\frac{d}{dt} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} F_1 (X_1, X_2, t) \\ F_2 (X_1, X_2, t) \end{bmatrix} \quad (3.33)$$

Where

$F_1 (X_1, X_2, t)$ is the nonlinear vector function and $F_2 (X_1, X_2, t)$ is a linear vector function of the form

$$F_2 (X_1, X_2) = A_1 X_1 + A_2 X_2 \quad (3.34)$$

The mathematical models of the vast majority of physical systems are of this form since only rarely are all components in the system all linear or all nonlinear. This form of mathematical model might also arise from nonlinear differential equations of the third or higher order. This is shown by a simple example -

$$\frac{d^n x}{dt^n} + f \left[\frac{d^{n-1} x}{dt^{n-1}}, \frac{d^{n-2} x}{dt^{n-2}}, \dots, \frac{dx}{dt}, x, t \right] = 0 \quad (3.35)$$

Then if $\frac{dx}{dt} = X_1, \frac{d^2 x}{dt^2} = X_2, \dots, \frac{d^{n-1} x}{dt^{n-1}} = X_{n-1}$

The higher order equation reduces to the following set of first order equations

$$\frac{d}{dt} \begin{bmatrix} x \\ X_1 \\ \vdots \\ X_{n-2} \\ X_{n-1} \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{n-1} \\ -f(x, X_1, \dots, X_{n-2}, X_{n-1}, t) \end{bmatrix} \quad (3.36)$$

Using the standard Runge-Kutta procedure given in Appendix C, we have for the linear equations

$$\frac{d}{dt} X_2 = A_1 X_1 + A_2 X_2$$

the recursion formula

$$X_2(nh+h) = X_2(n) + \Delta X_2$$

where

$$\Delta X_2 = \frac{1}{6} [L_1 + 2 L_2 + 2 L_3 + L_4] \quad (3.37)$$

and

$$L_1 = h A_1 X_1(n) + A_2 X_2(n) \quad (3.38)$$

$$L_2 = h A_1 \left[X_1(n) + \frac{K_1}{2} \right] + A_2 \left[X_2(n) + \frac{L_1}{2} \right] \quad (3.39)$$

$$L_3 = h A_1 \left[X_1(n) + \frac{K_2}{2} \right] + A_2 \left[X_2(n) + \frac{L_2}{2} \right] \quad (3.40)$$

$$L_4 = h A_1 \left[X_1(n) + K_3 \right] + A_2 \left[X_2(n) + L_3 \right] \quad (3.41)$$

Substituting (3.38) through (3.41) into (3.37) gives

$$\begin{aligned} \Delta X_2 &= h \left[U + \frac{1}{2} h A_2 + \frac{1}{6} h^2 A_2^2 + \frac{1}{24} h^3 A_2^3 \right] [A_1 X_1 + A_2 X_2] \\ &= \frac{h}{6} \left[(2U + \frac{1}{2} h A_2 + \frac{1}{4} h^2 A_2^2) A_1 K_1 + (U + \frac{1}{2} h A_2) A_1 K_2 \right. \\ &\quad \left. + A_1 K_3 \right] \end{aligned} \quad (3.42)$$

where

$$K_1 = h F [X_1(n), X_2(n), nh] \quad (3.43)$$

$$K_2 = h F \left[X_1(n) + \frac{K_1}{2}, X_2(n) + \frac{L_2}{2}, nh + \frac{h}{2} \right] \quad (3.44)$$

$$K_3 = h F \left[X_1(n) + \frac{K_2}{2}, X_2(n) + \frac{L_2}{2}, nh + \frac{h}{2} \right] \quad (3.45)$$

$$K_4 = h F [X_1(n) + K_3, X_2(n) + L_3, nh + h] \quad (3.46)$$

Inspection of equations (3.43) through (3.46) shows that K_1, K_2, K_3 are of the same order; and at least for the present, will be assumed to be equal for this special case, i.e., $K_3 \approx K_2 \approx K_1 \approx h F [X_1(n), X_2(n), nh]$ (3.47). The error at each step can be calculated in terms of the parameters and can be corrected later if it becomes too large. With this approximation (3.42) can be rewritten as

$$\Delta X_2 = h \alpha [A_1 X_1(n) + A_2 X_2(n)] + h \beta A_1 K_1 \quad (3.48)$$

where

$$\alpha = [U + \frac{1}{2} h A_2 + \frac{1}{6} h^2 A_2^2 + \frac{1}{24} h^3 A_2^3] \quad (3.49)$$

$$\beta = \frac{1}{6} [4 U + h A_2 + \frac{1}{4} h^2 A_2^2] \quad (3.50)$$

The recursion formula therefore becomes

$$X_2(n+1) = X_2(n) + h \alpha [A_1 X_1(n) + A_2 X_2(n)] + h \beta A_1 K_1 \quad (3.51)$$

The error in equation (3.51) is expressed in terms of K_1 , K_2 where K_j , $j = 1, 2, 3, 4$ are defined as

$$K_1 = h F [X_1(n), X_2(n), nh] \quad (3.52)$$

$$K_2 = h F [X_1(n) + \frac{K_1}{2}, X_2(n) + \frac{1}{2} \Delta X_2, nh + \frac{h}{2}] \quad (3.53)$$

$$K_3 = h F \left[X_1(n) + \frac{K_2}{2}, X_2(n) + \frac{1}{2} \Delta X_2, nh + \frac{h}{2} \right] \quad (3.54)$$

$$K_4 = h F \left[X_1(n) + K_3, X_2(n+1), nh + h \right] \quad (3.55)$$

and

$$X_1(n+1) = X_1(n) + \frac{1}{\ell} [K_1 + 2K_2 + 2K_3 + K_4] \quad (3.56)$$

The error in (3.56) is

$$\epsilon(n+1) = \frac{1}{\ell} h \left[2U + \frac{1}{2} h A_2 \right] [A_1] [K_1 - K_2] \quad (3.57)$$

If the error ϵ as given by (3.57) is too large, the correction may be made by subtracting $\epsilon(n+1)$ from $X_2(n+1)$ as calculated by 3.56 and the calculations in equations (3.52) through (3.57) repeated. Only one such repetition is normally required and would not even be necessary for a carefully chosen step size.

The computation time for this procedure is substantially less than the time for the Runge-Kutta (standard) method. This especially is true if the mathematical model has a small number of nonlinear equations and a large number of linear equations. Let these numbers be j and k respectively. Then the computation time is $(4j+k)$ units as compared to $4(j+k)$ units by the Standard Runge-Kutta Method.

3.3 More on the Solution of Mixed Linear and Nonlinear Systems

Another special, but very common class of nonlinear differential equations encountered in the study of physical systems is the form

$$\frac{d}{dt} X = F(X) + G(X) \quad (3.58)$$

where

$$F(X) = AX \quad (3.59)$$

and

$G(X)$ is nonlinear in X containing quadratic or higher degree terms in X . The procedure to be developed for solving the equations involves first a linear approximation to (3.58) followed by a change in the parameters to take into account the nonlinearity, i.e., the nonlinear function $G(X)$ is neglected as a starting point only. Let the linear approximation to (3.58) be taken as

$$\frac{d}{dt} X(t) = AX(t) \quad (3.60)$$

From the results of section two the solution to (3.60) is

$$X(t+h) = \alpha X(t) \quad (3.61)$$

where

$$\alpha = U + hA + \frac{1}{2} h^2 A^2 + \frac{1}{6} h^3 A^3 + \frac{1}{24} h^4 A^4 + \dots \quad (3.62)$$

The form of the solution to (3.50) is then taken as

$$X(t+h) = \alpha Z(t+h) \quad (3.63)$$

Equation (3.64) is obtained by Taylor's series expansion of $X(t+h)$

which gives:

$$X(t+h) = \alpha X(t) + \alpha \left[\int_t^{t+h} G[X(t)] dt \right] \quad (3.65)$$

Equation (3.65) then gives the recursive formula

$$X(n+1) = \alpha[X(n) + M(n+1)] \quad (3.66)$$

where

$$M(t) = \int G[X(t)] dt \quad (3.67)$$

Numerical integration of (3.67) is obtained by any of the standard numerical methods, such as Runge-Kutta method. The scheme then reduces to the following modified Runge-Kutta Procedure.

Procedure:

A numerical solution to a set of nonlinear differential equations (3.53) is given by the recursive formula

$$X(n+1) = \alpha[X(n) + M(n+1)] \quad (3.68)$$

Where

$$M(n+1) = \frac{1}{6} [M_1(n+1) + 2M_2(n+1) + 2M_3(n+1) + M_4(n+1)] \quad (3.69)$$

and

$$M_1(n+1) = h G [X(n)] \quad (3.70)$$

$$M_2(n+1) = h G [X(n) + \frac{1}{2} M_1(n+1)] \quad (3.71)$$

$$M_3(n+1) = h G [X(n) + \frac{1}{2} M_2(n+1)] \quad (3.72)$$

$$M_4(n+1) = h G [X(n) + M_3(n+1)] \quad (3.73)$$

Example:

The exact equation for motion of a pendulum is

$$\frac{d^2 x_1}{dt^2} + \frac{g}{L} \sin x_1 = 0 \quad (3.74)$$

For relatively small oscillations (3.74) can be taken as

$$\frac{d^2 x_1}{dt^2} + \frac{g}{L} (x_1 - \frac{1}{6} x_1^3) \quad (3.75)$$

Then if $\frac{g}{L} = k$

Let

$$\frac{dx_1}{dt} = x_2 \quad (3.76)$$

and

$$\frac{d^2 x_1}{dt^2} = \frac{dx_2}{dt} = -k \left(x_1 - \frac{1}{6} x_1^3 \right) \quad (3.77)$$

The system of first order equations therefore is

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -k & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{k}{6} x_1^3 \end{bmatrix} \quad (3.78)$$

or in the notations of this section

$$\frac{d}{dt} X = AX + G(X) \quad (3.79)$$

Solution by the above Runge-Kutta procedure is realized by the recursive formula

$$X(n+1) = \alpha [X(n) + M(n+1)] \quad (3.80)$$

where

$$M(n+1) = \frac{1}{6} [M_1(n+1) + 2M_2(n+1) + 2M_3(n+1) + M_4(n+1)] \quad (3.81)$$

and

$$M_1(n+1) = h \begin{bmatrix} 0 \\ \frac{k}{6} x_1^3(n) \end{bmatrix} \quad (3.82)$$

$$M_2(n+1) = h \begin{bmatrix} 0 \\ \frac{k}{6} [x_1(n) + \frac{1}{2} M_2(n+1)]^3 \end{bmatrix} \quad (3.83)$$

$$M_3(n+1) = h \begin{bmatrix} 0 \\ \frac{k}{c} [x_1(n) + \frac{1}{2}M_2(n+1)]^3 \end{bmatrix} \quad (3.34)$$

$$M_4(n+1) = h \begin{bmatrix} 0 \\ \frac{k}{c} [x_1(n) + M_3(n+1)]^3 \end{bmatrix} \quad (3.35)$$

IV NONLINEAR ALGEBRAIC EQUATIONS

When systems contain components modeled in terms of nonlinear algebraic equations it is not always possible to develop a system model in normal form, i.e., it may not be possible to eliminate the nonlinear algebraic equations from the system model. The mathematical model, in this case is referred to as being in seminormal form and for the purposes of our discussion is represented in the form

$$\frac{d}{dt} \begin{bmatrix} X \\ 0 \end{bmatrix} = \begin{bmatrix} F(X,Y, t) \\ G(X,Y, t) \end{bmatrix} \quad (4.1)$$

Where $G(X,Y, t)$ is a vector function nonlinear in X and Y or at least nonlinear in Y ; so that Y cannot be expressed explicitly in terms of X and t . If $G(X,Y, t)$ is linear in Y , the vector Y can be eliminated from the first equation and the model reduced to normal form.

Many attempts have been made to obtain a solution to the nonalgebraic equations. None of the procedures, however, is general. Each method is applicable only to a certain class of problems. The procedures to be used then depends on the nature of the nonlinearities involved in a given problem.

4.1 Method of Differentiation

One method of solution related to recent work by Wirth⁽¹⁾ is to transform the nonlinear algebraic equations into differential equations by partial differentiation. The procedure is as follows:

The algebraic equation in (4.1) viz.

$$G(X, Y, t) = 0$$

is differentiated with respect to t , so that

$$\frac{d}{dt} G(X, Y, t) = \frac{\partial G}{\partial X} \frac{dX}{dt} + \frac{\partial G}{\partial Y} \frac{dY}{dt} + \frac{\partial G}{\partial t} = 0 \quad (4.2)$$

Where $\frac{\partial G}{\partial Y}$ and $\frac{\partial G}{\partial X}$ is the Jacobian of the vector G with respect to the vectors Y and X respectively.

For example consider the set of equations

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} f_1(x_1, x_2, x_3, y_1, y_2, t) \\ f_2(x_1, x_2, x_3, y_1, y_2, t) \\ f_3(x_1, x_2, x_3, y_1, y_2, t) \\ g_1(x_1, x_2, x_3, y_1, y_2, t) \\ g_2(x_1, x_2, x_3, y_1, y_2, t) \end{bmatrix} \quad (4.3)$$

The form of equation (4.2) for this system of equations is

$$\begin{bmatrix} \frac{\partial G_1}{\partial x_1} & \frac{\partial G_1}{\partial x_2} & \frac{\partial G_1}{\partial x_3} \\ \frac{\partial G_2}{\partial x_1} & \frac{\partial G_2}{\partial x_2} & \frac{\partial G_2}{\partial x_3} \end{bmatrix} \begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \frac{dx_3}{dt} \end{bmatrix} + \begin{bmatrix} \frac{\partial G_1}{\partial y_1} & \frac{\partial G_1}{\partial y_2} \\ \frac{\partial G_2}{\partial y_1} & \frac{\partial G_2}{\partial y_2} \end{bmatrix} \begin{bmatrix} \frac{dy_1}{dt} \\ \frac{dy_2}{dt} \end{bmatrix} + \begin{bmatrix} \frac{\partial G_1}{\partial t} \\ \frac{\partial G_2}{\partial t} \end{bmatrix} = 0 \quad (4.4)$$

If the matrix $\frac{\partial G}{\partial Y}$ is nonsingular, then the solution (4.4) for the derivative vector $\frac{\partial Y}{\partial t}$ is

$$\begin{aligned} \frac{\partial Y}{\partial t} &= - \left[\frac{\partial G}{\partial Y} \right]^{-1} \left[\frac{\partial G}{\partial X} \frac{dx}{dt} + \frac{\partial G}{\partial t} \right] \\ &= - \left[\frac{\partial G}{\partial Y} \right]^{-1} \left[\frac{\partial G}{\partial X} F(X, Y, t) + \frac{\partial G}{\partial t} \right] \\ &= P(X, Y, t) \end{aligned} \quad (4.5)$$

The conditions under which $\left[\frac{\partial G}{\partial Y} \right]^{-1}$ exists are established in reference (1). The initial conditions on Y must satisfy the nonlinear algebraic equation

$$G[X(0), Y(0), 0] = 0 \quad (4.6)$$

By considering (4.5) in conjunction with (4.1) the mixed system of nonlinear algebraic and differential equations are transformed into the set of nonlinear differential equations in normal form

$$\frac{d}{dt} \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} F(X, Y, t) \\ P(X, Y, t) \end{bmatrix} \quad (4.7)$$

The numerical procedures of sections 2 and 3 can be used to obtain a solution.

Example:

Let the mathematical model of a system be given as

$$\frac{d}{dt} \begin{bmatrix} X \\ 0 \end{bmatrix} = \begin{bmatrix} f(x, y, t) \\ g(x, y, t) \end{bmatrix} = \begin{bmatrix} ax + by \\ x - \sin y + \sin \omega t \end{bmatrix} \quad (4.8)$$

Taking the time derivative of g with respect to t we have

$$\frac{dg}{dt} = \frac{dx}{dt} - \cos y \frac{dy}{dt} + \omega \cos \omega t = 0$$

or

$$\begin{aligned} \frac{dy}{dt} &= (ax + by + \omega \cos \omega t) / \cos y \\ &= p(x, y, t) \end{aligned} \quad (4.9)$$

The nonlinear differential equations to be solved then are

$$\frac{d}{dt} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} ax + by \\ p(x, y, t) \end{bmatrix} \quad (4.10)$$

with initial conditions satisfying the equation

$$f [x(o), y(o), 0] = x(o) - \sin y(o) + \sin (o) = 0 \quad (4.11)$$

$y(o) = \pi/6$ is such a value so that

$$g [x(o), y(o), 0] = 0.5 - \sin \frac{\pi}{6} + \sin o = 0.$$

Therefore, the initial conditions for (4.10) are

$$x(o) = 0.5 \text{ and } y(o) = \pi/6.$$

4.2 An Alternate Procedure

In many a problem the technique as indicated in section 4.1 may not be applicable for the reason that Jacobian $\frac{\partial G}{\partial Y}$ may not exist or is singular. Even if the Jacobian is nonsingular the final mathematical form may be very complex and, as such, is difficult to handle numerically. The following numerical procedure overcomes this difficulty.

Again consider the mathematical model

$$\frac{d}{dt} \begin{bmatrix} X \\ 0 \end{bmatrix} = \begin{bmatrix} F(X, Y, t) \\ G(X, Y, t) \end{bmatrix} \quad (4.11)$$

The basic procedure involves an estimate of $X(n+1)$ from values of $X(n)$ and $Y(n)$. Before considering the details of how this estimate is to be made let it be assumed for the present that an estimated value of $X(n+1)$ is available. The problem then reduces to the solution of the set of nonlinear algebraic equations

$$G[X(n+1), Y(n+1), nh+h] = 0 \quad (4.12)$$

for $Y(n+1)$. The Newton-Raphson can perhaps be applied to obtain this solution using $Y(n)$ as starting values. However, if the true solution for the vector $Y(n+1)$ is considerably different from $Y(n)$ this method may not converge, or it may converge very slowly, thereby requiring an excessive number of iterations to arrive at a solution. In an attempt to avoid this difficulty an estimate of $Y(n+1)$ is made using quadratic extrapolation. Using $\bar{Y}(n+1)$ to represent this estimate we have

$$\begin{aligned} \bar{Y}(n+1) &= Y(n) + [Y(n) - Y(n-1)] + \frac{1}{2} [Y(n) - 2Y(n-1) + Y(n-2)] \\ &= \frac{5}{2} Y(n) - 2 Y(n-1) + \frac{1}{2} Y(n-2) \end{aligned} \quad (4.13)$$

Then

$$G [X(n+1), Y(n+1), nh+h] = G [X(n+1), \bar{Y}(n+1), nh+h]$$

$$\simeq [Y(n+1) - \bar{Y}(n+1)] \frac{\partial G}{\partial Y} [X(n+1), \bar{Y}(n+1), nh+h]$$

and as a first degree approximation

$$Y(n+1) = \bar{Y}(n+1) - \left[\frac{\partial G}{\partial Y} (n+1) \right]^{-1} \bar{G}(n+1) \quad (4.14)$$

This last expression is precisely the Newton-Raphson formula. The modification that $\bar{Y}(n+1)$ as evaluated by (4.13) is used instead of $Y(n)$. On the basis of a typical example it appears that only one such calculation is necessary for convergence. Although it may first appear from the form of (4.14) that a first order approximation is used, it is actually a third order approximation, since $\bar{Y}(n+1)$ has been estimated by quadratic extrapolation.

The disadvantage of this method appears to be in the evaluation of the inverse $\left[\frac{\partial G}{\partial Y} \right]^{-1}$ at every step. However it may be possible to evaluate this inverse analytically.

The estimate of $X(n+1)$ required in the above procedure can be established by any of the predictor type recursion formulas, but Milne's formula seems to be more appropriate.

Milne's formula is

$$X(n+1) = X(n-3) + \frac{4h}{3} [2 F(n-2) - F(n-1) + 2 F(n)] \quad (4.15)$$

After the vector $Y(n+1)$ is calculated by (4.14) the estimated vector $X(n+1)$ can be corrected by any of the integration formulas that use estimated solutions. Simpson's formula is one such formula and gives

$$\begin{aligned} X(n+1) &= X(n-1) + \frac{T}{3} [X'(n+1) + 4 X'(n) + X'(n-1)] \\ &= X(n-1) + \frac{T}{3} [\bar{F}(n+1) + 4 F(n) + F(n-1)] \end{aligned} \quad (4.16)$$

It can be noted that the procedure developed in this section requires solutions at the preceeding steps. This is a handicap for the first few steps. It can be overcome by using a smaller interval for the first few steps and by use of lower order methods that do not require the solutions at the preceeding steps. For example let

$$\frac{d}{dt} \begin{bmatrix} x \\ o \end{bmatrix} = \begin{bmatrix} f(x, y, t) \\ g(x, y, t) \end{bmatrix} \quad (4.17)$$

The first three steps $x(1)$, $x(2)$, $x(3)$ are obtained from the first degree approximation with a temporary (local) step size of $\frac{1}{2}h$, thus,

$$x\left(\frac{1}{2}\right) = x(0) + \frac{h}{2} f[x(0), y(0), 0] \quad (4.18)$$

$$y\left(\frac{1}{2}\right) = y(0) - \frac{g[x(\frac{1}{2}), y(0), \frac{h}{2}]}{\frac{\partial}{\partial y} g[x(\frac{1}{2}), y(0), \frac{h}{2}]}$$

$$x(1) = x\left(\frac{1}{2}\right) + \frac{h}{2} f[x(\frac{1}{2}), y(\frac{1}{2}), \frac{h}{2}] \quad (4.19)$$

$$y(1) = y\left(\frac{1}{2}\right) - \frac{g[x(1), y(\frac{1}{2}), \frac{h}{2}]}{\frac{\partial}{\partial y} g[x(1), y(\frac{1}{2}), h]}$$

V. EXAMPLE SOLUTIONS

In this section the computer results for several simple cases have been given for the purpose of comparing computation times with some of the other methods.

5.1 Numerical Solution of a Linear System in Normal Form

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad (5.1)$$

Using the procedure of section 2.1

$$\begin{bmatrix} x_1(n+1) \\ x_2(n+1) \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix} \begin{bmatrix} x_1(n) \\ x_2(n) \end{bmatrix} + \begin{bmatrix} \beta_{11} \\ \beta_{21} \end{bmatrix} \quad (5.2)$$

where

$$\alpha_{11} = 1 - h + \frac{1}{2} h^2 - \frac{1}{6} h^3 + \frac{1}{24} h^4$$

$$\alpha_{12} = 0$$

$$\alpha_{21} = h - \frac{3}{2} h^2 + \frac{3}{2} h^3 - \frac{5}{8} h^4$$

$$\alpha_{22} = 1 - 2h + 2h^2 - \frac{4}{3} h^3 + \frac{2}{3} h^4$$

$$\beta_{11} = h - \frac{1}{2} + \frac{1}{6} h^3 - \frac{1}{24} h^4$$

$$\beta_{21} = \frac{1}{2} h^2 - \frac{1}{2} h^3 + \frac{1}{24} h^4$$

The numerical results for the above linear system as obtained by three different methods are given in Table (5.1) along with the computation time. Calculations were made with a time increment of 0.001 units, with results printed out every 10th interval.

TABLE 5.1
SOLUTION OF LINEAR SYSTEM OF EQUATION
Recurrative Method of Section 2.1

Time Units	x1	x2
0.00	2.00000	3.00000
0.10	1.90484	2.63293
0.20	1.81873	2.32421
0.30	1.74082	2.06404
0.40	1.67032	1.84431
0.50	1.60653	1.65835
0.60	1.54881	1.50060
0.70	1.49659	1.36648
0.80	1.44933	1.25217
0.90	1.40657	1.15432
1.00	1.36788	1.07088

Computation time: 2 minutes, 45 seconds

Predict-Correct Method of Appendix B

Time Units	x1	x2
0.00	2.00000	3.00000
0.10	1.90484	2.63293
0.20	1.81873	2.32421
0.30	1.74082	2.06404
0.40	1.67032	1.84431
0.50	1.60653	1.65835
0.60	1.54881	1.50060
0.70	1.49659	1.36648
0.80	1.44933	1.25217
0.90	1.40657	1.15432
1.00	1.36788	1.07088

Computation time: 8 minutes 5 seconds

TABLE 5.1 Cont'd.

Runge-Kutta Method

Time Units	x1	x2
0.00	2.00000	3.00000
0.10	1.90434	2.63293
0.20	1.81373	2.32421
0.30	1.74032	2.06404
0.40	1.67032	1.84431
0.50	1.60653	1.65836
0.60	1.54330	1.50061
0.70	1.49653	1.36650
0.80	1.44931	1.25219
0.90	1.40655	1.15434
1.00	1.36735	1.07091

Computation time: 3 minutes 35 seconds

5.2 Numerical solution to mixed linear and nonlinear system-closed linear set.

The mathematical model under consideration is

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_1 x_2 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (5.3)$$

The numerical solution as obtained by the two procedures developed in section 3.1 and by the Runge-Kutta method along with the computation time for each is given in Table (5.2). The calculations were made with a time increment of 0.001 units. The results were printed out at every 10th step with every 10th printed result being given in Table (5.2).

TABLE 5.2
SOLUTION OF MIXED LINEAR AND NONLINEAR EQUATIONS WITH CLOSED LINEAR SET
Procedure 1

Time Units	x1	x2
0.00	+ 1.00000	+ 1.00000
0.01	+ 1.01015	+ 1.02010
0.02	+ 1.02061	+ 1.04040
0.03	+ 1.03139	+ 1.06091
0.04	+ 1.04250	+ 1.08162
0.05	+ 1.05394	+ 1.10254
0.06	+ 1.06574	+ 1.12367
0.07	+ 1.07790	+ 1.14502
0.08	+ 1.09043	+ 1.16657
0.09	+ 1.10334	+ 1.18835
0.10	+ 1.11665	+ 1.2.034

Computation time: 6 minutes, 50 seconds

Procedure 2

Time Units	x1	x2
0.00	+ 1.00000	+ 1.00000
0.01	+ 1.01015	+ 1.02010
0.02	+ 1.02061	+ 1.04040
0.03	+ 1.03139	+ 1.06091
0.04	+ 1.04250	+ 1.08162
0.05	+ 1.05395	+ 1.10254
0.06	+ 1.06574	+ 1.12367
0.07	+ 1.07790	+ 1.04502
0.08	+ 1.09043	+ 1.16657
0.09	+ 1.10335	+ 1.18835
0.10	+ 1.1666	+ 1.21034

Computation time: 6 minutes, 5 seconds

TABLE 5.2 Continued
Runge-Kutta Method

Time Units	x1	x2
0.00	+ 1.00000	+ 1.00000
0.01	+ 1.01015	+ 1.02016
0.02	+ 1.02061	+ 1.04040
0.03	+ 1.03139	+ 1.06091
0.04	+ 1.04250	+ 1.09162
0.05	+ 1.05395	+ 1.10254
0.06	+ 1.06574	+ 1.12467
0.07	+ 1.07790	+ 1.14502
0.08	+ 1.09043	+ 1.16657
0.09	+ 1.10335	+ 1.18835
0.10	+ 1.11666	+ 1.21034

Computation time: 3 minutes 15 seconds.

5.3 Mixed Linear-Nonlinear Mathematical Model

Consider the set of equations

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -\sin x_1 + x_2^2 + \sin 100t \\ -x_1 - x_2 \end{bmatrix} \quad (5.4)$$

$$x_1(0) = 0 \quad x_2(0) = +1$$

In the notations of Section 3.2

$$f(x_1, x_2, t) = -\sin x_1 + x_2^2 + \sin 100t \quad (5.5)$$

$$A_1 = -1 \quad A_2 = -1 \quad (5.6)$$

$$\begin{aligned} \alpha &= [U + \frac{1}{2} h A_2 + \frac{1}{6} h^2 A_2^2 + \frac{1}{24} h^3 A_2^3] \\ &= [1 - \frac{1}{2} h + \frac{1}{6} h^2 - \frac{1}{24} h^3] \end{aligned} \quad (5.7)$$

$$\begin{aligned} \beta &= \frac{1}{6} [4 U + h A_2 + \frac{1}{4} h^2 A_2^2] \\ &= \frac{1}{6} [4 - h + \frac{1}{4} h^2] \end{aligned} \quad (5.8)$$

The numerical solutions as obtained by the procedure of Section 3.2 and by the Runge-Kutta scheme are given in Table 5.3.1 along with computation times. The time increment used is 0.0001 units and results are printed out at every 10th step; with every 10th printed results given in Table 5.3.

SOLUTION OF A CLASS OF MIXED LINEAR AND NONLINEAR SYSTEM

MODIFIED RUNGE-KUTTA PROCEDURE

Time Units	x1	x2
0.00	0.000000	1.000000
0.01	0.0103473	0.981999
0.02	0.0213633	0.979991
0.03	0.0329704	0.969969
0.04	0.0450701	0.959930
0.05	0.0576121	0.949067
0.06	0.0700247	0.939700
0.07	0.0823374	0.929671
0.08	0.0945375	0.919540
0.09	0.106440	0.909350
0.10	0.118013	0.899224

Computation time: 10 minutes, 15 seconds

Standard Runge-Kutta Procedure

Time Units	x1	x2
0.00	0.000000	1.000000
0.01	0.0103473	0.981999
0.02	0.0213633	0.979991
0.03	0.0329704	0.969970
0.04	0.0450701	0.959930
0.05	0.0576122	0.949063
0.06	0.0700247	0.939702
0.07	0.0823375	0.929672
0.08	0.0945376	0.919542
0.09	0.106510	0.909392
0.10	0.118013	0.899226

Computation time: 12 minutes, 55 seconds

5.4 Mixed Linear and Nonlinear Mathematical Model Given in Equation (3.3)

The mathematical model under consideration is

$$\frac{d}{dt} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -10 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{5}{3} x_1^3 \end{bmatrix} \quad (5.9)$$

By notations of section 3.3;

$$A = \begin{bmatrix} 0 & 1 \\ -10 & 0 \end{bmatrix} \quad \text{and} \quad G(x) = \begin{bmatrix} 0 \\ \frac{5}{3} x_1^3 \end{bmatrix} \quad (5.10)$$

Then

$$\alpha_{11} = 1 - 5h^2 + \frac{25}{6} h^4$$

$$\alpha_{12} = h - \frac{5}{3} h^3 + \frac{5}{6} h^5$$

$$\alpha_{21} = -10 \left(h - \frac{5}{3} h^3 + \frac{5}{6} h^5 \right)$$

$$\alpha_{22} = 1 - 5h^2 + \frac{25}{6} h^4$$

The numerical solution to (5.2) by Modified Runge-Kutta Method and Standard Runge-Kutta Method is given in Table 5.4. The time increment used is 0.001 time units and results are printed out at every 10th step; with every 10th printed result given in Table 5.4.

TABLE 5.4
SOLUTIONS OF ANOTHER CLASS OF LINEAR AND NONLINEAR SYSTEMS

Modified Runge-Kutta Procedure of Section 3.3

Time Units	x1	x2
0.00	+ 0.500000	0.000000
0.10	+ 0.476216	- 0.472181
0.20	+ 0.406958	- 0.902567
0.30	+ 0.298456	- 1.250510
0.40	+ 0.160822	- 1.479810
0.50	+ 0.007335	- 1.564410
0.60	- 0.146878	- 1.494180
0.70	- 0.286602	- 1.277550
0.80	- 0.398322	- 0.939304
0.90	- 0.471588	- 0.514959
1.00	- 0.499789	- 0.044926

Computation time: 7 minutes, 10 seconds

Standard Runge-Kutta Procedure

Time Units	x1	x2
0.00	+ 0.500000	0.000000
0.10	+ 0.476216	- 0.472181
0.20	+ 0.406958	- 0.902567
0.30	+ 0.298456	- 1.250510
0.40	+ 0.160822	- 1.479810
0.50	+ 0.007335	- 1.564410
0.60	- 0.146878	- 1.494180
0.70	- 0.286602	- 1.277550
0.80	- 0.398322	- 0.939303
0.90	- 0.471589	- 0.514958
1.00	- 0.499790	- 0.044925

Computation time: 10 minutes, 50 seconds

5.5 Mathematical Models with Nonlinear Algebraic Equations

The mathematical model considered is

$$\frac{d}{dt} \begin{bmatrix} x \\ 0 \end{bmatrix} = \begin{bmatrix} -x + \cos y \\ x - \sin y \end{bmatrix} \quad (5.11)$$

$$x(0) = 0.5, \quad y(0) = 0.5236$$

According to the method of differentiation (5.11) reduces to

$$\frac{d}{dt} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -x + \cos y \\ (-x + \cos y) / \cos y \end{bmatrix}$$

$$= \begin{bmatrix} -x + \cos y \\ 1 - (x / \cos y) \end{bmatrix} \quad (5.12)$$

$$x(0) = 0.5, \quad y(0) = 0.5236$$

The solutions of (5.11) by the two procedures of section 4, along with the computation time are given in Table 5.5. The time increment of 0.001 units is used with every 10th result printed out at every 10th result and is given in Table 5.5.

TABLE 5.5
SOLUTION OF SYSTEM MODEL WITH NONLINEAR ALGEBRAIC EQUATIONS

Method of Differentiation

Time	x	y
0.00	+ 0.500000	+ 0.523600
0.10	0.533832	0.563127
0.20	0.562574	0.597498
0.30	0.586874	0.627193
0.40	0.607325	0.652691
0.50	0.624470	0.674454
0.60	0.638793	0.692929
0.70	0.650721	0.708534
0.80	0.660628	0.721656
0.90	0.668837	0.732645
1.00	0.675627	0.741816

Computation time: 12 minutes, 30 seconds

Alternate Procedure

Time	x	y
0.00	0.500000	0.523600
0.10	0.533832	0.563126
0.20	0.562575	0.597497
0.30	0.586874	0.627192
0.40	0.607325	0.652689
0.50	0.624470	0.674453
0.60	0.638793	0.692928
0.70	0.650721	0.708533
0.80	0.660628	0.721655
0.90	0.668838	0.732644
1.00	0.675627	0.741815

Computation Time: 7 minutes, 5 seconds

VI. CONSLUSION

Several procedures for numerical solutions of mathematical models are given in this thesis. In section two, a recursive formula is developed for linear systems. The computation time by this formula is one fourth of that by the Runge-Kutta method. Moreover, any given degree of accuracy can be maintained without increasing the computation time per step. In the case of linear systems, the expression for the nth stage solution vector in terms of the initial conditions vector and input vectors has a striking similarity to the mathematical model of discrete state Markov Process. Thus the "Dynamic Programming" techniques developed by Howard and others for optimizing the systems also apply to the control systems.

In section 3, mixed linear and nonlinear systems are considered. The procedures developed for the various types of linear and nonlinear combinations are essentially modified Runge-Kutta Procedures, in which the computation time has been reduced considerably. This saving in time is achieved by applying the results of section 2 to the subsets in the system.

In section 4, systems containing nonlinear algebraic equations are considered. Two procedures are given for the solution of such a system. The method of differentiation as given in section 4.1 gives better results, but is relatively slower as compared to the alternate procedure of section 4.2. Both the procedures, however, seem to converge rapidly.

In section 5, several examples of solutions by the procedures developed in this thesis are given and compared with the solutions obtained by several other methods. These comparisons, indeed, are consistent with the theoretical results.

REFERENCES

1. WIRTH J.L., Time Domain Models of Physical Systems and Existence of Solutions, Technical Report No. 1, NSF, G-20949, Michigan State University, East Lansing, 1962.
2. MILNE W.E., Numerical Solution of Differential Equations, John Wiley and Sons Inc., 1953.
3. HOUSEHOLDER, A.S., Principles of Numerical Analysis, McGraw Hill, 1953
4. HARTREE, D.R., Numerical Analysis, Oxford University Press, 1953.
5. HAMMING, R.W., Numerical Methods for Scientists and Engineers, McGraw Hill, 1962.
6. HOWARD, R.A., Dynamic Programming and Markov Processes, The Technology Press and John Wiley and Sons Inc. 1960.
7. BELLMAN R., Stability Theory of Differential Equations, McGraw Hill, 1953.
8. BUCKINGHAM, R.A., Numerical Methods, Pitman, 1957.
9. KUNZ, K.S., Numerical Analysis, McGraw Hill, 1957.
10. BENNETT A.A., Milne W.E and Bateman H., Numerical Integration of Differential Equations, Dover Publications, 1956.

Appendix A

Recurssive formula for various degrees of accuracy.

See Section 2.1.

A.1. Third degree accuracy.

$$X(n+1) = [\alpha] X(n) + \beta_1 E(n) + \beta_2 E(n + \frac{1}{2}) + \beta_3 E(n+1) \quad (A.1)$$

where

$$[\alpha] = U + h [A] + \frac{1}{2} h^2 [A]^2 + \frac{1}{6} h^3 [A]^3 \quad (A.2)$$

$$[\beta_1] = h [\frac{2}{3} U + \frac{1}{6} hA + \frac{1}{6} h^2 A^2] \quad (A.3)$$

$$[\beta_2] = h [-\frac{1}{3} U + \frac{1}{3} hA] \quad (A.4)$$

$$[\beta_3] = h [\frac{2}{3} U] \quad (A.5)$$

A.2 Fourth degree accuracy.

$$X(n+1) = [\alpha] X(n) + \beta_1 E(n) + \beta_2 E(n + \frac{1}{2}) + \beta_3 E(n+1) \quad (A.6)$$

$$[\alpha] = U + hA + \frac{1}{2} h^2 A^2 + \frac{1}{6} h^3 A^3 + \frac{1}{24} h^4 A^4 \quad (A.7)$$

$$\beta_1 = h [\frac{1}{2} U + \frac{1}{3} hA + \frac{1}{12} h^2 A^2 + \frac{1}{24} h^3 A^3] \quad (A.8)$$

$$\beta_2 = h [\frac{1}{12} h^2 A^2] \quad (A.9)$$

$$\beta_3 = h [\frac{1}{2} U + \frac{1}{6} hA] \quad (A.10)$$

A.3 Fifth degree accuracy.

$$X(n+1) = \alpha X(n) + \beta_1 E(n) + \beta_2 E(n + \frac{1}{2}) + \beta_3 E(n+1) \quad (A.11)$$

where

$$[\alpha] = [U + hA + \frac{1}{2} h^2 A^2 + \frac{1}{6} h^3 A^3 + \frac{1}{24} h^4 A^4 + \frac{1}{120} h^5 A^5] \quad (A.12)$$

$$[\beta_1] = h [\frac{5}{15} U + \frac{3}{10} hA + \frac{7}{60} h^2 A^2 + \frac{1}{40} h^3 A^3 + \frac{1}{120} h^4 A^4] \quad (A.13)$$

$$[\beta_2] = h [-\frac{1}{15} U + \frac{1}{15} hA + \frac{1}{60} h^2 A^2 + \frac{1}{60} h^3 A^3] \quad (A.14)$$

$$[\beta_3] = h [\frac{3}{15} U + \frac{2}{15} hA + \frac{1}{30} h^2 A^2] \quad (A.15)$$

E.4 Sixth degree accuracy.

$$X(n+1) = \alpha X(n) + \beta_1 E(n) + \beta_2 E(n + \frac{1}{2}) + \beta_3 E(n+1) \quad (A.16)$$

$$[\alpha] = [U + hA + \frac{1}{2} h^2 A^2 + \frac{1}{6} h^3 A^3 + \frac{1}{24} h^4 A^4 + \frac{1}{120} h^5 A^5 + \frac{1}{720} h^6 A^6] \quad (A.17)$$

$$[\beta_1] = h [\frac{19}{36} U + \frac{11}{36} hA + \frac{1}{9} h^2 A^2 + \frac{11}{360} h^3 A^3 + \frac{1}{180} h^4 A^4 + \frac{1}{720} h^5 A^5] \quad (A.18)$$

$$[\beta_2] = h [-\frac{1}{18} U + \frac{1}{18} hA + \frac{1}{36} h^2 A^2 + \frac{1}{180} h^3 A^3 + \frac{1}{360} h^4 A^4] \quad (A.19)$$

$$[\beta_3] = h [\frac{19}{36} U + \frac{5}{36} hA + \frac{1}{36} h^2 A^2 + \frac{1}{180} h^3 A^3] \quad (A.20)$$

APPENDIX B

B.1 Runge Kutta Method.

$$\text{Let } \frac{d}{dt} X(t) = F [X(t), t] \quad (\text{B.1})$$

Then $X(n+1)$ is calculated from $X(n)$ and B.1 by the recursion equation

$$X(n+1) = X(n) + \frac{1}{6} [K_1 + 2K_2 + 2K_3 + K_4] \quad (\text{B.2})$$

where

$$K_1 = hF [X(n), nh] \quad (\text{B.3})$$

$$K_2 = hF [X(n) + \frac{1}{2} K_1, nh + \frac{h}{2}] \quad (\text{B.4})$$

$$K_3 = hF [X(n) + \frac{1}{2} K_2, nh + \frac{h}{2}] \quad (\text{B.5})$$

$$K_4 = hF [X(n) + K_3, nh + h] \quad (\text{B.6})$$

For a linear system the numerical solution by the Runge-Kutta method is equivalent to the fourth order Taylor series expansion when the function F is not a direct function of the independent variable t . When, F is a direct function of t , the truncation error factor is much higher than the exact expansion.

B.2 A Predictor - Corrector Method.

A very commonly used procedure for the numerical solution to linear differential equations is the predictor-corrector method. There are many variations of this method, but the following one suggested by Milne and modified by Hamming is used as a comparison with the procedure of section 2.1, and so is included here.

The details of the general predictor-corrector methods can be found in Reference 5.

Let the differential equation be

$$\frac{d}{dt} X(t) = f(X, t) \quad (B.7)$$

Then

$$X(n+1) = X(n) + \frac{h}{2} [P'(n+1) + X'(n)] + \frac{h^2}{12} [X''(n) - P''(n+1)] \quad (B.8)$$

where

$$P(n+1) = X(n-1) + 2h X'(n+1) + h^2 \left[\frac{4}{3} X''(n) + \frac{2}{3} X''(n-1) \right] \quad (B.9)$$

ROOM USE ONLY

ROOM USE ONLY
