



**NUMERICAL METHODS IN THE TIME-DOMAIN
ANALYSIS OF SYSTEMS**

**Thesis for the Degree of M. S.
MICHIGAN STATE UNIVERSITY
John Charles Eidson
1960**

72-1

This is to certify that the

thesis entitled

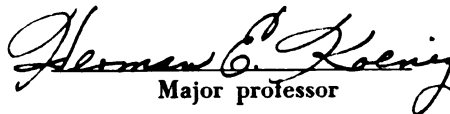
NUMERICAL METHODS IN THE TIME-DOMAIN
ANALYSIS OF SYSTEMS

presented by

JOHN CHARLES EIDSON

has been accepted towards fulfillment
of the requirements for

MASTER OF SCIENCE degree in ELECTRICAL ENGINEERING


Major professor

Date February 16, 1960

O-169



NUMERICAL METHODS IN THE TIME-DOMAIN
ANALYSIS OF SYSTEMS

By

John Charles Eidson

AN ABSTRACT

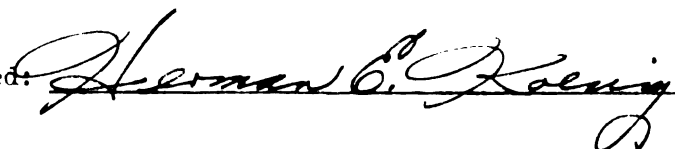
Submitted to the College of Engineering
Michigan State University of Agriculture and
Applied Science in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE

Department of Electrical Engineering

1960

Approved:

A handwritten signature in dark ink, appearing to read "Herman C. Koenig", is written over a horizontal line.

ABSTRACT

The two approaches to the problem of solving the algebraic and differential equations describing systems are techniques such as the Laplace transform, and time-domain techniques such as analog computer methods or classical differential equation theory. This thesis demonstrates another type of time-domain technique, numerical methods.

One of the major problems in the use of numerical methods is that of accuracy. As an illustration of the technique of determining the accuracy of a numerical method, a detailed error analysis of the Runge-Kutta fourth order method is given. A second problem is that of determining initial conditions in higher order systems. A numerical technique that partially circumvents this problem is presented.

The ultimate goal of system study is design. Numerical methods can be used effectively in system design. One such method that can be applied to certain control system problems is presented along with an illustrative example.

NUMERICAL METHODS IN THE TIME-DOMAIN
ANALYSIS OF SYSTEMS

By
John Charles Eidson

A THESIS

Submitted to the College of Engineering
Michigan State University of Agriculture and
Applied Science in partial fulfillment of
the requirements for the degree of

MASTER OF SCIENCE

Department of Electrical Engineering

1960

ACKNOWLEDGMENT

The author is indebted to his major professor, Dr. H. E. Koenig, and to Dr. G. P. Weeg of the Computer Laboratory, for their helpful advice and guidance throughout the preparation of this thesis.

CONTENTS

	Page
I. INTRODUCTION	1
II. THE EQUATIONS OF PHYSICAL SYSTEMS	3
III. NUMERICAL METHODS OF SOLVING ALGEBRAIC AND ORDINARY DIFFERENTIAL EQUATIONS	6
IV. APPLYING NUMERICAL TECHNIQUES TO SYSTEM ANALYSIS	15
V. OPTIMIZATION BASED ON SUCCESSIVE ANALYSES	20
VI. CONCLUSION	25

I. INTRODUCTION

The development of the high-speed electronic computer has made possible the practical use of numerical methods in system analysis. The main objective of this thesis is to discuss the use of numerical methods in the study of lumped-constant systems on the basis of the time-domain solutions to the algebraic and differential equations describing these systems.

Current methods of system analysis are usually based on the properties of transfer functions obtained by taking the Laplace transform of the differential equations describing the system. If the time-domain solution of the equations is desired, it must be obtained by techniques such as partial fraction expansion. Since a partial fraction expansion yields a sum of exponential time functions, a great deal of work is involved in obtaining a plot of the system variables as a function of time.

Another common practice in system analysis is to obtain a set of simultaneous differential equations which are equivalent to a block diagram by treating each s in the transfer functions as an operator representing the time derivative. The time-domain solution to these equations is then obtained with an analog computer. Design is carried out by observing the effects of changing the system constants. If the number of potentiometers which must be adjusted for each change of system constants is not too large, the analog computer affords a rapid method of observing system response.

Certain types of nonlinearity can be handled by analog computers with such devices as servos and electronic multipliers, but techniques based on the Laplace transform are limited to linear systems.

Numerical techniques of solution have at least two characteristics which make them useful in system analysis: (1) the time-domain solution is obtained directly and it is comparatively easy to study the effects of variations of the system parameters, and (2) nonlinear systems can be analyzed.

The discussion of these numerical techniques is in four sections. First a brief description of the equations that describe physical systems is given. Next the numerical methods are presented. The application of the numerical techniques to the solution of the system equations is presented, and finally a method of optimization based on successive analyses is discussed.

II. THE EQUATIONS OF PHYSICAL SYSTEMS

The problem of formulating the equations of physical systems has been discussed at great length by several authors, for example Koenig and Blackwell [5]. Only the results and steps which are influenced by the proposed use of numerical methods will be presented here. The notation used follows that of Koenig and Blackwell.

There are three distinct sets of equations which describe the physical systems under consideration. The first is the set of $e-v+1$ circuit equations

$$BX=0 \quad (2.1)$$

This set relates the across variables of the system. The second set consists of $v-1$ equations relating the system through variables and is called the cut-set.

$$SY=0 \quad (2.2)$$

Equations (2.1) and (2.2) together form a set of e independent equations based on the topology or configuration of the system. Since there are $2e$ system variables, e more independent equations are needed to obtain a solution to the system.

The remaining e equations are the terminal equations which describe the components of the system. These equations in general are of the following form.

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} X_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} A_{13} & A_{14} \\ A_{23} & A_{24} \end{bmatrix} \begin{bmatrix} Y_1 \\ X_2 \end{bmatrix} \quad (2.3)$$

$$\begin{aligned} X_1 &= K_1(t) \\ Y_2 &= K_2(t) \end{aligned} \quad (2.4)$$

where $K_1(t)$ and $K_2(t)$ are specified time functions and the A_{ij} are coefficient matrices. X_k and Y_k are partitionings of X and Y of equations (2.1) and (2.2). For linear systems the entries in the A matrices are of the form

$$a_{ij} = b_n p^n + b_{n-1} p^{n-1} + \dots + b_0 \quad (2.5)$$

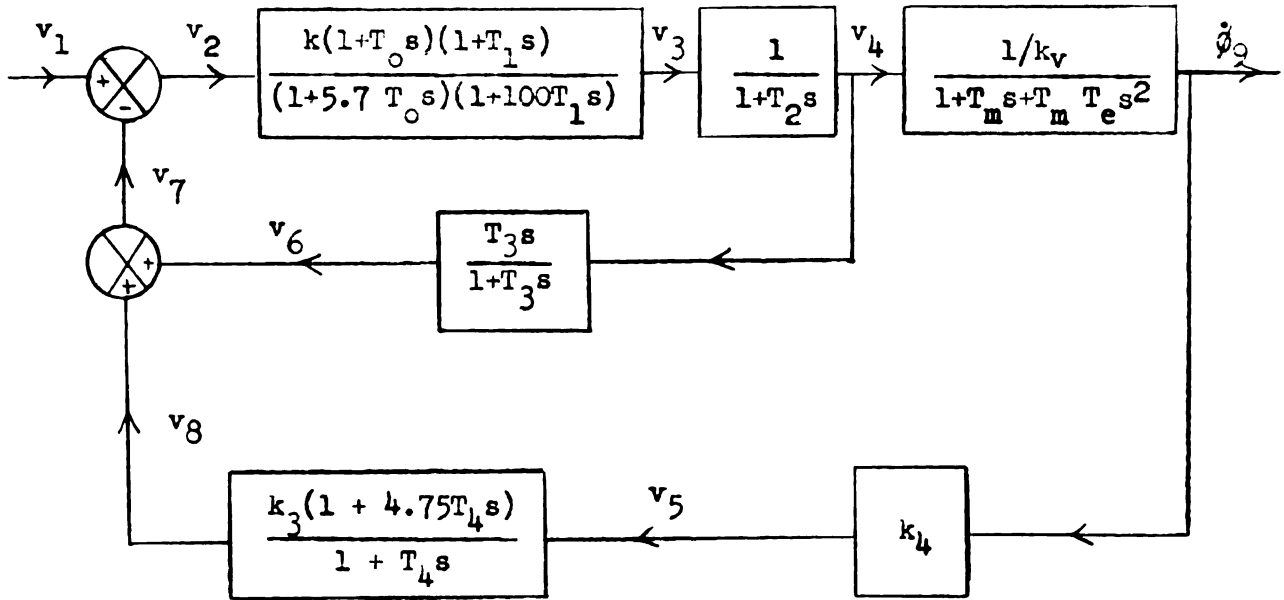
where the b_n are constants and $p^n \equiv d^n/dt^n$. Normally n is two or less.

The numerical techniques described in section III permit more freedom in the form of the terminal equations. Specifically, the techniques allow the b_n to be functions of the system variables and of time. With this freedom it is possible to describe nonlinear components by equations in the form of equation (2.3).

Using the methods discussed in [5], it is possible to combine equations (2.1) through (2.4) to obtain a reduced set of algebraic and differential equations which must be solved simultaneously.

If a block diagram of a system is available, a set of differential equations describing the system can be obtained by treating each s in the transfer function as a derivative operator. Example (2.1) demonstrates this method.

Example 2.1



$$\begin{aligned}
 .002 < T_1 < 1 & \quad T_e = .026 & \quad k_v = 1.83 \\
 T_2 = .03 & \quad T_m = .2 & \quad k_3 = .209 \\
 0 < T_3 < .05 & \quad T_o = .047 & \quad k_4 = 1.9 \\
 0 < T_4 < .09 & \quad k = 10,000 &
 \end{aligned} \tag{2.6}$$

$$\begin{aligned}
 (k + k[T_o + T_1]p + kT_o T_1 p^2)v_2 &= (1 + 5.7T_o p + 100T_1 p + 570T_o T_1 p^2)v_3 \\
 v_3 &= (1 + T_2 p)v_4 \\
 T_3 p v_4 &= (1 + T_3 p)v_6 \\
 (1 + T_m p + T_m T_e p^2)\ddot{\phi}_9 &= v_4/k_v \\
 v_5 &= k_4 \ddot{\phi}_9 \\
 (1 + T_4 p)v_8 &= k_3(1 + 4.75 T_4 p)v_5 \\
 v_7 &= v_6 + v_8 \\
 v_2 &= v_1 - v_7
 \end{aligned} \tag{2.7}$$

where $\rho \equiv \frac{d}{dt}$.

III. NUMERICAL METHODS OF SOLVING ALGEBRAIC AND ORDINARY DIFFERENTIAL EQUATIONS

This section of the thesis is devoted to a discussion of methods of solving the algebraic and differential equations described in the previous section. In any discussion of numerical methods, some mention should be made of existence and accuracy of solutions. For most of the methods discussed, treatments of error analysis are given in references [2], [3], and [6], and will not be repeated here. In the case of the Runge-Kutta fourth order formula a detailed error analysis will be given.

There are numerous methods available for obtaining the solutions to simultaneous algebraic equations [1, 2, 3, 6]. The method used in a given problem depends on the number of equations to be solved and the required degree of accuracy.

One of the simplest methods to code for a digital computer is the Gauss reduction method. This method consists of upper triangularizing the matrix A of equation (3.1).

$$AX = C \quad (3.1)$$

This triangularization is accomplished by dividing the first equation by the main diagonal coefficient and then eliminating the first variable from the succeeding equations. This process is repeated on each equation. The solution is then obtained by a back substitution beginning with the last equation of the modified system of equations. Although this method is simple to code, it is susceptible to round-off errors particularly when the number of equations is large.

A procedure for reducing the effects of round-off error on the solution of algebraic equations is shown in equations (3.2) and (3.4). Let X_1 be the values calculated in the preceding method.

$$AX_1 = C_1 \quad (3.2)$$

Subtracting (3.1) from (3.2) gives

$$A(X_1 - X) = C_1 - C \quad (3.3)$$

$$A\delta_1 = C_1 - C \quad (3.4)$$

where δ_1 is the error in X_1 . Solving equation (3.4) would yield the δ_1 if there were no round-off errors. Since there generally are round-off errors, the calculated δ_1 are only approximate. This process may be repeated to attempt to obtain better values of X_1 although there is no assurance that the process will converge. The Crout modification of the Gauss reduction is well adapted to this procedure as pointed out by Hildebrand [2].

The previous methods are applicable only to linear equations. Methods which can be used to solve both linear and nonlinear algebraic equations generally make use of an iterative technique, (see for example, Householder [3]).

Suppose the coefficients of A in equation (3.1) are a function of X. To solve this equation using an iterative technique, it is written in the form

$$X = F(X) \quad (3.5)$$

The iterative scheme is then defined by

$$X_{k+1} = F(X_k) \quad (3.6)$$

In general, $F(X_k)$ can be written from equation (3.1) in several different forms. The convergence of the iterative process depends not only on the form used, but also on the initial value of X_k .

Another method that can be used to solve sets of nonlinear algebraic equations is apparent when equation (3.1) is written as

$$f_i(X) = 0 \quad i = 1, 2, 3, \dots, n \quad (3.7)$$

Define the function $M(X)$ as

$$M(X) = \sum_{i=1}^n |f_i(X)| \quad (3.8)$$

Clearly M is always greater than or equal to zero and will be equal to zero only when the vector X is a solution to (3.7). Therefore the solution to (3.7) can be obtained by finding the vector X for which the function $M(X)$ is an absolute minimum. Further comments on the process of minimization will be given in section V.

As in the case of algebraic equations there are many numerical methods available for obtaining the solutions of simultaneous ordinary differential equations [2, 3, 6]. Before these methods are discussed, several remarks on the form of the differential equations and the existence of their solutions are necessary.

The differential equations describing physical systems are generally a set of n^{th} order simultaneous differential equations. Since n is usually greater than one and since many of the numerical methods are applicable only to systems of first order equations, it is convenient to reduce the n^{th} order system to a system of first order equations. The following example illustrates this procedure.

Example 3.1

Reduce the n^{th} order differential equation (3.9) to a system of n first order equations.

$$a_n y^{(n)} + a_{n-1} y^{(n-1)} + \dots + a_0 y + f(y, y', \dots, y^{(n-1)}, t) = 0 \quad (3.9)$$

$$y^i(0) = k_i \quad i = 0, 1, \dots, n-1 \quad (3.10)$$

$$\begin{aligned}
\text{Let } y' &= z_1 \\
y'' &= z_2 = z_1' \\
&\vdots \\
y^{(n-1)} &= z_{n-1} = z_{n-2}' \\
y^{(n)} &= z_{n-1}'
\end{aligned} \tag{3.11}$$

Then the n first order equations obtained from (3.10) are

$$\begin{aligned}
a_n z_{n-1}' &= -a_{n-1} z_{n-1} - a_{n-2} z_{n-2} - \dots - a_1 z_1 \\
&\quad - a_0 y - f(y, z_1, z_2, \dots, z_{n-1}, t)
\end{aligned} \tag{3.12}$$

$$\begin{aligned}
z_i' &= z_{i+1} \quad i = 1, 2, \dots, n-2 \\
y' &= z_1
\end{aligned}$$

The initial conditions of the variables in (3.12) are

$$\begin{aligned}
z_{i+1}(0) &= k_{i+1} \quad i = 0, 1, 2, \dots, n-2 \\
y(0) &= k_0
\end{aligned} \tag{3.13}$$

The procedure illustrated can be applied to a system of higher order equations to yield the set of first order equations

$$AX' = G(X, t) \tag{3.14}$$

$$X(t_0) = K \tag{3.15}$$

To solve equation (3.14) using the techniques to be described later, it is necessary to write (3.14) in the form

$$X' = F(X, t) \tag{3.16}$$

This requires that A be nonsingular. Further discussion of this problem will be given in the next section.

Once the set of differential equations (3.16) and the initial conditions (3.15) are established, the question of the existence of a unique

solution must be considered. This is necessary since numerical methods will give an answer whether a solution exists or not and because the conditions required for existence play an important role in the error analysis of the numerical methods. A complete discussion and proof of the following theorem can be found in Ince [4].

Theorem:

Given a system of ordinary differential equations (3.16) and a set of initial conditions (3.15), there exists a unique set of continuous solutions which assume the values specified by (3.15) when $t = t_0$ and satisfies the differential equation (3.16) in a region defined by

$|t - t_0| \leq d$ provided the following conditions are met:

a) $F(X, t)$ is single valued and continuous in all its arguments

in a domain D defined by

$$\begin{aligned} |t - t_0| &\leq a \\ |X - X(t_0)| &\leq B \end{aligned} \quad (3.17)$$

where if $Y = (y_{ij})$ then $|Y| = (|y_{ij}|)$.

$$b) \quad \left| F(\bar{X}, t) - F(X, t) \right| < \left(\sum_{i=1}^n c_i \left| \bar{x}_i - x_i \right| \right) = C(X) \quad (3.18)$$

where the x_i are the arguments of F in the domain D .

c) if M is the greatest upper bound of F in D then

$$d = \min (a, b_{ij} / M). \quad (3.19)$$

This theorem gives a set of sufficient conditions for the existence of unique continuous solutions to the differential equations. The condition of inequality (3.18) is known as a Lipschitz condition and will be used in the error analysis of the Runge-Kutta method.

Once the existence of a solution to a given set of differential equations is assured, there are two distinct types of numerical methods which

may be used. One type is based on a Taylor series approximation and the other on quadrature formulas. In the following discussion X_1 represents the vector X when $t = t_1$. The quantities h , t_1 , and t_0 are related by

$$t_1 = t_0 + 1h \quad (3.20)$$

An example of a method based on quadrature formulas is Milne's method. This method involves two separate formulas, a predictor (3.21) based on Q_{24} and a corrector (3.22) based on Q_{22} , (see Kunz [6]).

$$X_{n+1} = X_{n-3} + \frac{4h}{3} (2X'_{n-2} - X'_{n-1} + 2X'_n) \quad (3.21)$$

$$X_{n+1} = X_{n-1} + \frac{h}{3} (X'_{n-1} + 4X'_n + X'_{n+1}) \quad (3.22)$$

Assuming X_{n-3} , X'_{n-2} , X'_{n-1} , and X'_n are given, the predictor (3.21) is used to calculate an estimate of X_{n+1} . This estimate is then used in the corrector (3.22) to obtain a better value of X_{n+1} . The corrector may be applied several times to the new value of X_{n+1} to yield better values of X_{n+1} . These formulas have accuracies comparable with the Runge-Kutta formulas and have the advantage that only one evaluation of $F(X, t)$, equation (3.16), is necessary for each evaluation of the predictor and corrector formulas. Furthermore, if h is small enough, only one application of the corrector will be necessary.

The Milne formulas have the disadvantage of requiring four starting values of X . To obtain these starting values an alternate method such as the Runge-Kutta method must be used. This alternate method must also be used if it becomes necessary to reduce h at some point in the calculation to decrease truncation error. Since the Milne method actually requires two methods to be coded and stored in a digital computer, it is not practical in many situations.

A method which requires only the initial conditions (3.15) for starting values is the Runge-Kutta fourth order approximation. Although this method does not require several starting values, it has the disadvantage that the function $F(X, t)$ must be evaluated four times in calculating X_{n+1} from X_n . The fact that only one set of formulas is required makes the Runge-Kutta method more practical in most situations. A common Runge-Kutta fourth order scheme is given by the following equations.

$$X_{i+1} - X_i = \Delta X_i \quad (3.23)$$

$$\Delta X_i = \frac{1}{6} (\Delta' X_i + 2\Delta'' X_i + 2\Delta''' X_i + \Delta^{IV} X_i) \quad (3.24)$$

$$\Delta' X_i = h F(X_i, t_i) \quad (3.25)$$

$$\Delta'' X_i = h F(X_i + \frac{\Delta'}{2} X_i, t_i + \frac{h}{2}) \quad (3.26)$$

$$\Delta''' X_i = h F(X_i + \frac{\Delta''}{2} X_i, t_i + \frac{h}{2}) \quad (3.27)$$

$$\Delta^{IV} X_i = h F(X_i + \Delta''' X_i, t_i + h) \quad (3.28)$$

As an example of the procedure used in determining the accuracy of a numerical method, a detailed error analysis will be given. In the following equations a bar over a quantity indicates that it differs from the correct value due to some type of error, (e.g., round-off error).

The Runge-Kutta formula is an approximation to a Taylor series. Assume that the solution to the differential equation, $X(t)$, can be expanded in a convergent Taylor series around the point t_i .

$$X_{i+1} - X_i = X_i' h + X_i'' \frac{h^2}{2!} + X_i''' \frac{h^3}{3!} + X_i^{(IV)} \frac{h^4}{4!} + R_i \quad (3.29)$$

$$R_i = X^{(V)}(Z_i, \tau_i) \frac{h^5}{5!} = F^{(IV)}(Z_i, \tau_i) \frac{h^5}{5!} \quad (3.30)$$

where Z_i is in the interval (X_i, X_{i+1}) and τ_i is in the interval (t_i, t_{i+1}) .

If equations (3.25) through (3.28) are expanded in Taylor series and substituted into (3.24), ΔX_i will agree exactly with the right hand side of (3.29) through terms involving F'''' . Therefore the ΔX_i defined by (3.24) differs from the true value of ΔX_i in (3.23) by an amount

$$\phi_i = R_i + \frac{1}{6} (2R'' + 4R''' + R^{IV}) \quad (3.31)$$

where R'' , R''' , and R^{IV} are higher order terms in the expansions of equations (3.26) through (3.28) and are of the form

$$R^n = \frac{h^5}{4!} F^{(IV)}(Z, \tau) \quad (3.32)$$

where Z and τ are in the appropriate sub-intervals of $(X_i, t_i; X_{i+1}, t_{i+1})$. ϕ_i is called the truncation error.

In general the arguments used to compute the Δ^n of equations (3.25) through (3.28) are in error, with the result that the Δ^n are in error as indicated by the following equations.

$$\overline{\Delta'X_i} = h F(\bar{X}_i, t_i) \quad (3.33)$$

$$\overline{\Delta''X_i} = h F(\bar{X}_i + \frac{\Delta'}{2} X_i, t_i + \frac{h}{2}) \quad (3.34)$$

$$\overline{\Delta'''X_i} = h F(\bar{X}_i + \frac{\Delta''}{2} X_i, t_i + \frac{h}{2}) \quad (3.35)$$

$$\overline{\Delta^{IV}X_i} = h F(\bar{X}_i + \Delta''' X_i, t_i + h) \quad (3.36)$$

Note that the errors in each Δ (as indicated by the bar) are due only to errors in the arguments of F and not to errors in calculating F itself.

Let δ_i be defined by the following equations.

$$\delta_i = \overline{\Delta X_i} - \Delta X_i = \frac{1}{6} (\delta_i' + 2\delta_i'' + 2\delta_i''' + \delta_i^{IV}) \quad (3.37)$$

$$\delta_i' = \overline{\Delta'X_i} - \Delta'X_i \quad (3.38)$$

$$\delta_i'' = \overline{\Delta''X_i} - \Delta''X_i \quad (3.39)$$

$$\delta_i''' = \overline{\Delta''''X_i} - \Delta''''X_i \quad (3.40)$$

$$\delta_i^{IV} = \overline{\Delta^{IV}X_i} - \Delta^{IV}X_i \quad (3.41)$$

Using the Lipschitz condition (3.18), a bound can be placed on δ_i .

$$\delta_i < h C(Z_i) \quad (3.42)$$

where Z_i is in the interval (X_i, X_{i+1}) . Note that δ_i is zero if X_i is exact and if there is no computation or round-off error in the evaluation of (3.24) through (3.28).

In general, each Δ in equations (3.24) through (3.28) will also be in error as a result of round-off and other computation errors. Let this error in each Δ^n be symbolized by η^n and the round-off error made in evaluating (3.24) by η^0 , then the resulting error in ΔX_i is called η_i and is defined in (3.43).

$$\eta_i = \eta_i^0 + \frac{1}{6}(\eta_i' + 2\eta_i'' + 2\eta_i''' + \eta_i^{IV}) \quad (3.43)$$

Notice that the error in $\Delta^n X_i$ due to η_i^{n-1} is included in δ_i^n .

The total error in computing the $(i+1)^{st}$ point from the i^{th} point is given by

$$X_{i+1} - X_i = \Delta X_i + E_i \quad (3.44)$$

where

$$E_i \leq \phi_i + \delta_i + \eta_i \quad (3.45)$$

Although E_i represents the upper bound of the error, the actual error will rarely approach this figure.

IV. APPLYING NUMERICAL TECHNIQUES TO SYSTEM ANALYSIS

Using the numerical methods discussed in section III, it is possible to solve the system of equations presented in section I. If the equations are all algebraic or all differential, there is very little to do except choose the appropriate numerical method and solve them.

In the case of mixed algebraic and differential equations several more problems must be solved before the solution can be obtained. In particular, the problem of solving for X' mentioned in connection with (3.16) must be resolved. The question of initial conditions in mixed and in pure differential systems must also be considered.

The problem of mixed systems and of solving for X' can be illustrated best by preparing (2.7) for numerical solution.

Example 4.1

The first step is to reduce the second order differential equations to first order equations. Equation (2.7) then can be written as the first order system (4.1).

$$\begin{aligned}
 k(T_o + T_1)pv_2 + kT_o T_1 p_2 - (100T_1 + 5.7T_o)pv_3 - 570T_o T_1 p_3 &= -kv_2 - v_3 \\
 pv_2 &= \alpha_2 \\
 pv_3 &= \alpha_3 \\
 T_2 pv_4 &= v_3 - v_4 \\
 T_3 pv_4 - T_3 pv_6 &= v_6 \\
 T_m p\ddot{\phi}_9 + T_m T_e p\ddot{\phi}_9 &= v_4/k_v - \ddot{\phi}_9 \\
 p\ddot{\phi}_9 &= \beta_9 \\
 -4.75 T_4 k_3 pv_5 + T_4 pv_8 &= -v_8 \\
 v_5 &= k_4 \ddot{\phi}_9 \\
 v_7 &= v_6 + v_8 \\
 v_2 &= v_1 - v_7
 \end{aligned} \tag{4.1}$$

It is impossible to solve for the derivative terms in (4.1). One procedure for bringing up the rank of the differential equations is to differentiate the algebraic equations and possibly some of the differential equations. The resulting system is then reduced to a system of first order equations in which it is possible to solve for the derivative terms.

By eliminating v_7 from the algebraic equations and then taking the second derivative of the remaining algebraic equations and the first derivative of the fourth, fifth, and eighth differential equations, a new system of mixed second and first order equations is obtained. When this system is reduced to a system of first order equations, it is possible to solve for the derivative terms. The result is shown in (4.2).

$$\begin{aligned}
 p\beta_9 &= v_4 / (T_m T_e k_v) - (\dot{\phi}_9 + T_m \beta_9) / (T_m T_e) \\
 p\dot{\phi}_9 &= \beta_9 \\
 pv_5 &= \alpha_5 \\
 p\alpha_5 &= k_4 (p\beta_9) \\
 pv_8 &= \alpha_8 \\
 p\alpha_8 &= (1/T_4) [-\alpha_8 + k_3 \alpha_5 + 4.75k_3 T_4 (p\alpha_5)] \\
 pv_4 &= \alpha_4 \\
 p\alpha_4 &= (1/T_2) (\alpha_3 - \alpha_4) \\
 pv_6 &= \alpha_6 \\
 p\alpha_6 &= (1/T_3) [-\alpha_6 + T_3 (p\alpha_4)] \\
 pv_2 &= \alpha_2 \\
 p\alpha_2 &= p^2 v_1 - (p\alpha_8) - (p\alpha_6) \\
 pv_3 &= \alpha_3 \\
 p\alpha_3 &= -[1/(570T_o T_1)] [-kv_2 - k(T_o + T_1)\alpha_2 + v_3 \\
 &\quad + (5.7T_o + 100T_1)\alpha_3 - kT_o T_1 (p\alpha_2)]
 \end{aligned} \tag{4.2}$$

It should be noted that the derivative terms appearing on the right side of some of the equations can be written in terms of the variables. For coding and computation purposes it is easier to use the equations as shown in (4.2) since terms common to several equations are emphasized.

Probably the most difficult problem connected with solving the differential equations describing a system is determining the required initial conditions. The only reason for analyzing a system on the basis of the differential equations is to determine the response of the system to transients. If the steady state characteristics are desired, the system will be described by algebraic equations. In the case of a.c. systems the algebraic equations will have complex coefficients. These equations can be solved by the methods of section III. The Crout method can be used to solve systems of equations with complex coefficients [2].

Since it is generally very difficult to determine the correct initial conditions of the variables for a given driving function, the following method of attack is useful. The "steady state" values of the variables are determined by algebraic means. The driving functions are then approximated by functions whose derivatives permit the use of the "steady state" initial conditions. For example, the driving function v_1 in (4.2) would need to have a differentiable first derivative and a second derivative equal to zero for t equal to zero.

In order to approximate discontinuous functions such as step functions, the rise and fall times of the approximating functions are made short with respect to the system response times. For example, the polynomial (4.3) may be used to approximate the leading edge of a step function when the first and second derivatives must be zero at the beginning and end of the leading edge.

$$v(t) = \frac{1}{2} \left(\frac{t^5}{5} - \frac{t_1 t^4}{2} + \frac{t_1^2 t^3}{3} \right) a + v(0) \quad (4.3)$$

where $v(0)$ is the initial value of v , t_1 is the length of the leading edge, and $\frac{t_1^5}{60} a$ is the height of the step. This technique will be used in solving (4.2) in connection with section V.

If the system being studied has nonlinear elements, then $F(X, t)$ in equation (3.16) will have coefficients which are functions of one or more of the variables. The only difference this will make in the numerical treatment is that at each evaluation of $F(X, t)$, the appropriate values of the coefficients must be determined on the basis of the values of the system variables.

In the case of the Runge-Kutta method, the errors caused by nonlinear coefficients appear in (3.25) through (3.28). For example, in computing $\Delta^{IV} X_1$, the value of X which must be used in determining the values of the coefficients is $X_1 + \Delta''' X_1$. Since $X_1 + \Delta''' X_1$ generally is not the correct value of X at $t_1 + h$, namely X_{1+1} , the coefficients computed on the basis of $X_1 + \Delta''' X_1$ may be in error. This error is included in δ_1 of equation (3.45). This error can be minimized by using small values of h since the nonlinear coefficients are more nearly constant over small changes in the variables. The coefficients must be single-valued functions of the system variables.

In the case of algebraic equations with nonlinear coefficients, more than one solution may be possible. Since an iterative or a minimizing technique must be used, the solution obtained will depend on the starting point of the iteration or minimization. For certain starting points it may not be possible to obtain a solution. This problem will be discussed more in section V.

The methods of computing the values of nonlinear coefficients are also of some interest. If the exact functional relationship is known, it should be used since it will yield the greatest accuracy. For many components the relationship will be specified by curves. In this case some sort of approximation must be used [2, 3, 6, 7]. One method would be to store values from the curves and interpolate between them. A more practical solution from the standpoint of computer memory requirements is to obtain an approximating polynomial. In general a least squares polynomial will be used. In theory, polynomials can be obtained to approximate functions of any number of variables, but it becomes increasingly difficult to obtain sufficient accuracy as the number of variables is increased. The only satisfactory way of determining whether a given polynomial gives the required accuracy is to check the true value with the value produced by the polynomial at several points in the operating region.

V. OPTIMIZATION BASED ON SUCCESSIVE ANALYSES

In the preceding sections the methods of obtaining a numerical solution to the differential equations describing systems have been outlined. In this section a method of finding values of system constants which give a better system response will be presented.

Specifically, the values of system constants are desired which will give a specified response to a given input. For instance, in Example 2.1, the values of T_1 , T_3 , and T_4 which give the minimum rise time of ϕ_9 for a step function applied to v_1 might be desired. The technique used in this type of problem is to define a function, which is a measure of the deviation of the actual system performance from the desired performance, and minimize this function with respect to the system constants. This is the same type of process suggested for obtaining the solution to the nonlinear algebraic equation (3.7).

A typical form of a deviation function is

$$D(c_k) = \sum_{i=1}^n w(x_i, y_i, t_i) |x_i - y_i| \quad (5.1)$$

where $w(x_i, y_i, t_i)$ is a weighting function greater than zero, x_i is the value of the output variable at t_i , y_i is the desired value of x at t_i , and the c_k are the variable system constants. The weighting function can be used to give greater emphasis to certain types of deviations, such as overshoot. The summation on i running from 1 to n corresponds to a solution of the differential equations from $t=0$ to $t=nh$. Since D is always positive, it has a lower bound.

There are several ways of minimizing $D(c_k)$ with respect to the c_k , given starting values of the c_k . One technique is to repeatedly change each c_k by an amount equal to the negative of the k component of the unit

gradient vector of D as defined by

$$\Delta c_k = - \frac{\frac{\partial D}{\partial c_k}}{\left(\sum_{j=0}^n \left| \frac{\partial D}{\partial c_j} \right|^2 \right)^{1/2}} \quad (5.2)$$

The minimization is considered complete when the magnitude of the gradient of D is less than a given amount. This method has two shortcomings: (1) a great deal of computation is required to determine the components of the gradient vector, and (2) a saddle point in the function D will cause the gradient of D to be zero.

An alternative scheme which requires less computation and will not be troubled by saddle points, is minimization on one variable at a time. In this method c_1 is increased by an amount Δ and D is computed. If D is smaller, c_1 is increased by 2Δ and so forth. If $c_1 + \Delta$ increases D , then $c_1 - \Delta$ is tried. When no further reduction of D can be obtained by changing c_1 , the process is repeated with each of the other variables. After this process is completed, Δ can be reduced and the minimization continued. The computation is terminated when no further reduction of D is possible using values of Δ greater than a given amount.

In general, a finite minimization process can not yield a true minimum. Consider the problem of minimizing the function shown in Figure 5.1 with respect to x .

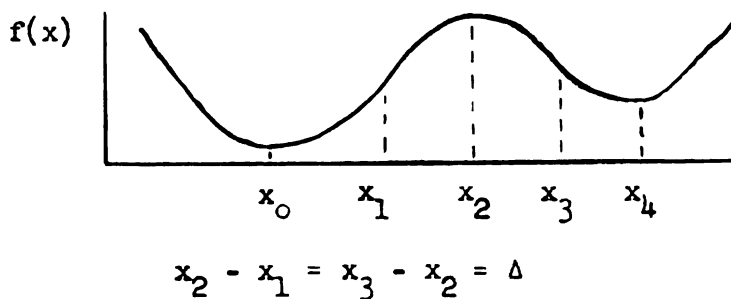


Figure 5.1

It can be seen that if the starting point was greater than x_3 a relative minimum would be found near x_4 , and if the process started with x less than x_1 , a value near x_0 would be indicated as the minimum. Since the minimization is of necessity a finite process, x_0 would be found for a minimum, in the absence of computation and round-off errors, only if x_0 happened to be one of the points at which $f(x)$ was evaluated.

In the case of solving algebraic equations, the minimum of the function (3.8) is known to be zero if a solution exists. In the case of a deviation function for improving system performance, the minimum will not usually be known. Therefore the only statement that can be made about a set of system constants found by this process is that they are the best set that can be obtained using this process and the given starting values. Different starting values or a different order of minimization (in the case of the one variable at a time technique) could yield a different minimum.

As an example of this technique, the system of Example 2.1 was optimized to give minimum rise time of $\dot{\phi}_9$ to the step function in v_1 which results in a 1% change in the output variable, $\dot{\phi}_9$. The system was optimized with respect to the constants T_1 , T_3 , and T_4 . The step function was approximated using polynomial (4.3) with $t_1 = .05$. The D function was defined as

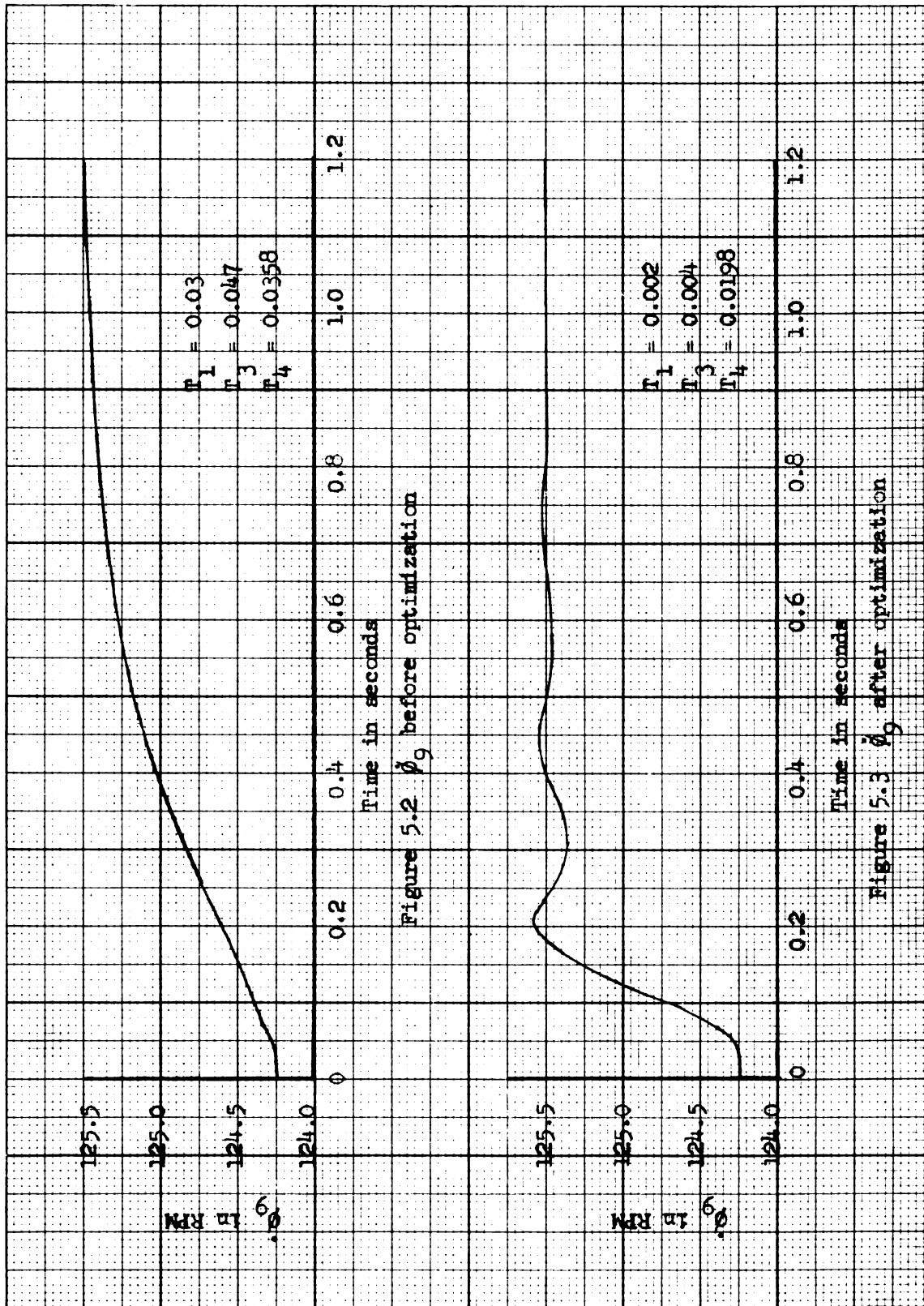
$$D = \int_0^{.6} (125.5 - \dot{\phi}_9)^2 dt. \quad (5.3)$$

The Runge-Kutta method was used with $h = .001$ from $t=0$ to $t= .1$ and $h = .0025$ for t greater than $.1$. The starting values of T_1 , T_3 , and T_4 and the corresponding solution of $\dot{\phi}_9$ are shown in Figure 5.2. The result of the optimization and the new solution of $\dot{\phi}_9$ is shown in Figure 5.3. The

initial conditions of the variables and the values of the variables for t greater than 1.5 are shown in Table 5.1.

Table 5.1

Initial conditions:		Values for $t > 1.5$	
$\dot{\phi}_9 = 124.25$	$\beta_9 = 0$	$\dot{\phi}_9 = 125.5$	$\beta_9 = 0$
$v_1 = 49.3624$	$\alpha_2 = 0$	$v_1 = 49.8591$	$\alpha_2 = 0$
$v_2 = .022738$	$\alpha_3 = 0$	$v_2 = .022967$	$\alpha_3 = 0$
$v_3 = 227.378$	$\alpha_4 = 0$	$v_3 = 229.665$	$\alpha_4 = 0$
$v_4 = 227.378$	$\alpha_5 = 0$	$v_4 = 229.665$	$\alpha_5 = 0$
$v_5 = 236.075$	$\alpha_6 = 0$	$v_5 = 238.45$	$\alpha_6 = 0$
$v_6 = 0$	$\alpha_8 = 0$	$v_6 = 0$	$\alpha_8 = 0$
$v_8 = 49.3397$		$v_8 = 49.8361$	



VI. CONCLUSION

A few additional comments on the practicality of this method of analysis are in order. One rather obvious disadvantage is the difficulty of studying the stability properties of a system using time domain analysis. For linear systems, stability can be studied by computing the eigenvalues of F , equation (3.16), although the computation is difficult.

The other major problem connected with numerical methods in the time domain is that of machine time and coding time. In all but a few cases, these methods will have to be carried out using a floating point scheme since the scaling difficulties in a problem of this size prohibit the use of the faster fixed point operations. Even so, running time is not completely unreasonable. For example, the solution to the fourteen simultaneous equations (4.2) was obtained at the rate of about thirty points per minute using a floating point routine with an add time of approximately two milliseconds.

The most serious problem is that of coding time. For numerical methods to be used effectively, general programs must be written that can perform a certain type of analysis on any one of a given class of systems. For example, a program was written in connection with this thesis which will perform the optimization discussed in section V with respect to any number of system constants. The only thing the programmer must provide is a routine to calculate $F(X, t)$, equation (3.16), and the appropriate initial conditions and other required constants. The techniques are almost useless unless these general programs are written since coding time for a problem of this kind is at least several days.

REFERENCES

1. Hestenes, M. R. and Stiefel, E., "Method of Conjugate Gradients for Solving Linear Systems," National Bureau of Standards Report 1659, (1952)
2. Hildebrand, F. B., "Introduction of Numerical Analysis," McGraw-Hill Book Company, Inc., New York (1956)
3. Householder, A. S., "Principles of Numerical Analysis," McGraw-Hill Book Company, Inc., New York (1953)
4. Ince, E. L., "Ordinary Differential Equations," Dover Publications, Inc. (1956)
5. Koenig, H. E. and Blackwell, W. A., "Introduction to System Analysis," Prentice-Hall, New York (to be published)
6. Kunz, K. S., "Numerical Analysis," McGraw-Hill Book Company, Inc., New York (1957)
7. Milne, W. E., Arntzen, W., Reynolds, N. and Wheelock, J., "Mathematics for Digital Computers, Volume I, Multivariate Interpolation," WADC Technical Report 57-556, Volume I (1958)

ROOM USE ONLY

ROOM USE ONLY