1963

A modified algorithm for Henrici's solution of y' ' = f (x,y)

Frank Garnett Walters

A MODIFIED ALGORITHM FOR HENRICI'S SOLUTION OF $y'' = f(x,y)$

BY

FRANK GARNETT WALTERS

A

THESIS

submitted to the faculty of the

SCHOOL OF MINES AND METALLURGY OF THE UNIVERSITY OF MISSOURI

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE, APPLIED MATHEMATICS

Rolla, Missouri

1963

Approved by

Charles A Johnson (advisor) J. P. Govier

Ralph E. Lee Aaron J Miles

ACKNOWLEDGEMENT

The author wishes to express his deep and sincere thanks to Dr. Charles A. Johnson for his assistance in all phases of this thesis, both in the preparation and editing. Also, the author is grateful for the assistance given to him by the Computer Center Staff and the Director of the Computer Center, Professor Ralph E. Lee.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

INTRODUCTION

Boundary value problems are of wide interest and computer-oriented solutions are of great value in applied mathematics. For instance, it may be desired to determine the trajectory of a ballistic missile which travels from one specified point on the surface of the earth to another in a given amount of time. Such a problem leads to the solution of either an ordinary or a partial differential equation depending on the approach to the solution. Boundary value problems involve differential equations of at least the second order (or systems of at least two equations of the first order). It is well known that it is theoretically always possible to reduce the solution of a boundary value problem to the solution of a sequence of initial value problems.

It has been shown by other authors that boundary value problems can be solved by standard methods such as regula falsi or Newton's method. It is the author's intention to apply Newton's method to the nonlinear equations, and to apply an adaptation of the Gaussian algorithm to the linear equations in the same manner as Henrici [1]. It is interesting to note that Albert Einstein, in his work pertaining to the orbital motion of the planets under the assumption of general relativity, that is, in the problem of perihelion shift, had to solve a nonlinear differential equation of the type $y'' = f(x,y)$.

It should be pointed out that even the simplest of all boundary value problems may have an infinite number of solutions or no solution. It must also be noted that the mathematical theory of the boundary value problem of this study is quite complicated, including the theory that would be involved in the numerical solution of the problem. Discussion of the theoretical foundation of the problem treated in this study may be found in the references cited.

It should also be noted that most nonlinear and some linear differential equations of the second order must be solved by some numerical analysis method. As will be pointed out shortly, some of our most useful second order differential equations that result from common physical conditions have no simple analytical solutions, as one would see if he were to consult the literature, for instance, Hildebrand (2), Struble (3), or Davis (4). There are many numerical methods for solving this type of second order linear and nonlinear differential equation. In devising more tractable methods, some mathematicians have developed their own functions to obtain analytic solutions of particular boundary value problems; but many boundary value problem solutions can be expressed in terms of the well-known Elliptic or Theta functions. Three of the sets of functions which the author has encountered that were developed to solve particular equations are Mathieu Functions, Bessel Functions and the First and Second Painlevé Transcendents. One can find a complete discussion of these functions in Struble [3] and Davis [4].

It has been proved that the differential equation of the type $y'' = f(x,y)$ with boundary conditions $y(a) = A$ and $y(b) = B$ has a unique solution for a wide class of functions, $f(x,y)$. This type of boundary value problem may, for example, result from a mathematical formulation of any or all of the following physical problems:

1. Trajectory of a ballistic missile
2. Vibration of a pendulum
3. Curve of pursuit
4. Orbital motion of the planets.

Some of the better known differential equations (boundary value problems) which arise from such problems are:

1. Duffing's Equation
2. Mathieu's Equation
3. Painlevé's Equation
4. Thomas-Fermi's Equation.

The previous examples were cited because they fit the exact form of the type of equation which is to be discussed in this thesis.

The purpose of the present study is to discuss the boundary value problems leading to equations exemplified by the four equations previously mentioned and to write a program for an I.B.M. computer in Fortran II language as a contribution to the numerical solution of an important class of linear and nonlinear differential equations.

CHAPTER II

REVIEW OF LITERATURE

There is extensive literature pertaining to the subject of linear and nonlinear differential equations and to numerical processes for solving them. To begin, it is convenient to discuss some methods that can be found in Hildebrand [2], Struble [3], Davis [4], and Fox [5].

There are basically two methods for solving boundary value problems. These are (a) methods which transform the boundary value problem to an initial value problem and (b) direct methods. Either or both of these methods may be such that the process is iterated upon until convergence is attained if the iteration converges. Whether or not an iteration process must be used is dependent on the method chosen and sometimes on the linearity of the function.

A method which transforms the boundary value problem to an initial value problem is the method of superposition. If the boundary value problem is linear and of the form

$$y'' + P(x)y' + Q(x)y = F(x) \quad (a \prec x \prec b) \qquad \underline{2.1}$$

with boundary conditions

$$y(a) = A, \quad y(b) = B$$

where A and B are finite constants, then this method can be used to obtain a solution. To describe superposition, assume $u(x)$ is any solution of

$$u'' + Pu' + Qu = F$$

that satisfies the initial condition

$$u(a) = A$$

and $v(x)$ is any nontrivial solution of

$$v'' + Pv' + Qv = 0$$

that satisfies the initial condition

$$v(a) = 0$$

Then the function

$$y(x) = u(x) + cv(x)$$

is a solution of equation (2.1) with one boundary condition $y(a) = A$ for any constant c. If the functions $P(x)$, $Q(x)$, and $F(x)$ are continuous in the interval $a < x < b$, then there exist no other functions with the above mentioned property. If the value of c can be determined from the equation

$$u(b) + cv(b) = B$$

then one has a solution. However, it is possible to have no solution, infinitely many solutions, or an unique solution, depending on the values of $u(b)$ and $v(b)$.

The corresponding nonlinear problem that has the more general form

$$y'' = G(x,y,y') \qquad \underline{2.2}$$

with boundary conditions

$$y(a) = A, \quad y(b) = B$$

is generally such that superposition is not valid. There are not very many methods that can be used to obtain a solution for the nonlinear problem. It is possible to reduce equation (2.2) to the initial value problem

$$u'' = G(x,u,u')$$

$$u(a) = A, \quad u'(a) = \alpha$$

which has as its solution $u(x,\alpha)$, and then attempt to determine α such that

$$u(b,\alpha) = B$$

The procedure is to find $u(b)$ for two or more trial values of α and then iterate, using inverse interpolation, to find an acceptable value of α. The disadvantages in using this method are many. Mainly, the process is apt to be tedious. Also, a small change in α does not necessarily correspond to a small change in $u(b,\alpha)$ and furthermore, there are the questions of uniqueness and existence of a solution. It appears that there is no completely satisfactory general method for dealing with the nonlinear problem. If the problem does not contain a y' term, then the method to be discussed in this thesis can be applied.

There are several methods for solving the initial value problem that arises from a transformation of the boundary value problem. Some of the better known methods for solving initial value problems are Stormer's method, Milne's method, Adams' method, and continuous analytic continuation. These and other methods can be found in Hildebrand [2], Struble [3], and Davis [4].

A direct method called the "γ-process", which could be used to solve the ordinary differential equation $y'' = -f(x,y,y')$ with the boundary conditions $g_1(y,y') = 0$ at $x = x_1$ and $g_2(y,y') = 0$ at $x = x_2$, is found in Fox [5].

A class of direct methods, which usually apply only when the problem is linear, is the one that approximates the differential equation with a difference equation which then requires solving a set of simultaneous algebraic equations. In this group is the method to be discussed in this thesis, taken from Henrici [1], who used a set of implicit difference equations.
tridiagonal and column matrices. Since this is the basis of the method dealt with in the present study, it will be discussed in detail in the next chapters.

One of the objectives of this study was to enhance the number of ordinates which can be found by the method of Henrici. Furthermore, Henrici's method was modified so that computer storage space necessary for a numerical solution is small as compared to that which would have been required. This was primarily accomplished by reducing each n x n tridiagonal matrix to a set of three n x 1 column matrices, so that instead of using n^2 storage spaces, only 3n spaces are required.

The manner in which these two objectives were attained will be discussed at length in the main body of this study.

CHAPTER III

DISCUSSION OF BASIC ALGORITHM

The problem that is to be discussed in this thesis is best described by the equations and conditions as follows:

$$y'' = f(x,y), \quad y(a) = A, \quad y(b) = B \qquad \underline{3.1}$$

where $-\infty < a < b < +\infty$, A and B are finite arbitrary constants and the function $f(x,y)$, in addition to satisfying the conditions of the existence theorem (theorem 1.1 found on pages 15-16 of Henrici [(1)]), is such that $f_y(x,y)$ is continuous and satisfies $f_y(x,y) \geq 0$, $a \leq x \leq b$, $-\infty < y < +\infty$.

The fact that the type of problem described by equations (3.1) has a unique solution has been proved by Henrici. The proof of this theorem can be found on pages 347-348 of Henrici [(1)].

It is the contention of this author that a numerical solution for equations (3.1) can be written in the Fortran II language for an I.B.M. computer which combines into one program the linear and nonlinear cases.

When attempting to find a numerical solution to the boundary value problem (3.1), it is desirable to divide the task into three principal cases:

<u>Case 1.</u> This case will be employed whenever the <u>function</u>, $f(x,y)$, can be expressed in the form $f(x,y) = yG(x) + P(x)$.

Case 2. This case will be employed whenever the function, $f(x,y)$, can be expressed in the form $f(x,y) = P(x) + T(x)H(y)$ where $H(y)$ is nonlinear in y and there is no linear term of y.

Case 3. This case will be employed whenever the function, $f(x,y)$, contains both linear and nonlinear terms of y, such that the function can be expressed in the form $f(x,y) = yG(x) + P(x) + T(x)H(y)$ where $H(y)$ is nonlinear in y.

It is the author's intention to discuss in some detail the process, in general, used to solve all three cases, and then to discuss in greater detail the individual cases. The procedure begins by replacing the given problem with a set of implicit difference equations, namely, the difference equations:

$$- y_{n-1} + 2y_n - y_{n+1} + h^2(\beta_0 f_{n-1} + \beta_1 f_n + \beta_2 f_{n+1}) = 0 \qquad \underline{3.2}$$

where $\beta_0 + \beta_1 + \beta_2 = 1$. The two most commonly used forms of equations (3.2) deal with only the sets of β's as follows:

$$\beta_0 = 0, \quad \beta_1 = 1, \quad \beta_2 = 0$$
$$\beta_0 = 1/12, \quad \beta_1 = 10/12, \quad \beta_2 = 1/12$$

These are by no means the only sets of values that could be used, only the sets of values to which this thesis will pertain. For more elaborate schemes, one should examine the references cited, such as Fox [(5)]. The difference equations (3.2) will be expressed as a system of matrix equations involving tridiagonal and column matrices. This set of matrix equations is easily solved by using an adaptation of the Gaussian algorithm if the system is linear, however, if the system is nonlinear, then the Newton-Raphson method for systems of transcendental equations needs to be applied.

It would now be appropriate to discuss the notation and the two cases as found in Henrici [1]. Henrici first introduced and defined the vectors (column matrices)

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \cdot \\ \cdot \\ \cdot \\ y_{N-1} \end{bmatrix}, \quad f(y) = \begin{bmatrix} f(x_1,y_1) \\ f(x_2,y_2) \\ \cdot \\ \cdot \\ \cdot \\ f(x_{N-1},y_{N-1}) \end{bmatrix}, \quad a = \begin{bmatrix} A - \beta_0 h^2 f(x_0,A) \\ 0 \\ \cdot \\ \cdot \\ 0 \\ B - \beta_2 h^2 f(x_N,B) \end{bmatrix}$$

and the tridiagonal matrices

$$J = \begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix}, \quad B = \begin{bmatrix} \beta_1 & \beta_2 & & & & \\ \beta_0 & \beta_1 & \beta_2 & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ & & & \beta_0 & \beta_1 & \beta_2 \\ & & & & \beta_0 & \beta_1 \end{bmatrix}$$

(where all the elements not on the main diagonal or on the diagonals adjacent to it are zero). The difference equations (3.2) can then be written as

$$Jy + h^2 Bf(y) = a \qquad \underline{3.3}$$

where h is a constant increment for the variable x. Defining the function $f(x,y) = yg(x) + k(x)$ which is linear in y, and $g(x)$ and $k(x)$ are given functions of x, it is also necessary to define the diagonal matrix

$$G = \begin{bmatrix} g(x_1) & & & & \\ & g(x_2) & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ & & & & & g(x_{N-1}) \end{bmatrix}$$

and the vector

$$k = \begin{bmatrix} k(x_1) \\ k(x_2) \\ \cdot \\ \cdot \\ \cdot \\ k(x_{N-1}) \end{bmatrix}$$

where $f(y) = Gy + k$. The system (3.3) can be expressed as

$$Ay = b \tag{3.4}$$

where

$$A = J + h^2BG, \quad b = a - h^2Bk \tag{3.5}$$

The solution of the system (3.3) is relatively easy due to the fact that A is a tridiagonal matrix. The elements of the tridiagonal matrix A can be expressed as

$$\begin{aligned} a_{n,n} &= 2 + h^2\beta_1 g_n, \quad a_{n,n-1} = -1 + h^2\beta_0 g_{n-1}, \\ a_{n,n+1} &= -1 + h^2\beta_2 g_{n+1}, \quad n = 1,2,\cdots, N-1 \end{aligned} \tag{3.5A}$$

and $a_{n,m} = 0$ for $|n - m| > 1$. Assume that two nonsingular matrices, $L = (l_{mn})$ and $U = (u_{mn})$, have been found where L is of the form

$$L = \begin{bmatrix} 1 & & & & & 0 \\ l_{21} & 1 & & & & \\ & l_{32} & 1 & & & \\ & & & \cdot & & \\ & & & & \cdot & \\ 0 & & & & l_{N-1,N-2} & 1 \end{bmatrix} \quad \begin{array}{l} (l_{mm} = 1; \ l_{mn} = 0 \\ \text{for } n>m \text{ or } n<m-1) \end{array}$$

and U is of the form

$$U = \begin{bmatrix} u_{11} & u_{12} & & & & 0 \\ & u_{22} & u_{23} & & & \\ & & \cdot & & & \\ & & & \cdot & & \\ & & & & u_{N-2,N-2} & u_{N-2,N-1} \\ 0 & & & & & u_{N-1,N-1} \end{bmatrix} \quad \begin{array}{l} (u_{mn} = 0 \text{ for } n<m \\ \text{or } n>m+1) \end{array}$$

and such that $LU = A$. To solve (3.4), it is first necessary to determine a vector z such that $Lz = b$ and second a vector y such that $Uy = z$. Because $y = U^{-1}z$ and $z = L^{-1}b$ and $U^{-1}L^{-1} = A^{-1}$, the vector y satisfies equation (3.4).

The procedure can be summarized as follows:

(I) Compute u_{nn} and z_n, starting with $n = 1$, using the relations:

$$u_{11} = a_{11}, \quad z_1 = b_1$$
$$l_{n,n-1} = a_{n,n-1}/u_{n-1,n-1}$$
$$u_{nn} = a_{nn} - l_{n,n-1}a_{n-1,n}$$
$$z_n = b_n - l_{n,n-1}z_{n-1}$$
$$n = 2,3,\cdots,N-1$$

(II) Compute y_n, starting with $n = N-1$, using the relations:

$$y_{N-1} = z_{N-1}/u_{N-1,N-1}$$
$$y_n = (z_n - a_{n,n+1}y_{n+1})/u_{nn}$$
$$n = N-2,\ldots,2,1$$

There are numerous tedious details which have been omitted that are necessary in order to go from equation (3.4) to the end result. If the reader wishes to refer to the details that were omitted, they can be found on page 353 of Henrici [1]. What has just been computed is the y that would satisfy the linear case of (3.3). This is what the author will refer to as Case 1.

If the function f(x,y) is nonlinear in y, then one cannot hope to solve equation (3.4) by any algebraic method. For this reason, Henrici used the Newton-Raphson method for systems of transcendental equations to solve equation (3.4). Suppose that equation (3.3) is now written as

$$Jy + h^2Bf(y) - a = 0 \qquad \underline{3.6}$$

and assume that $y^{(0)}$ is close to the actual solution of equation (3.6), so that the residual vector

$$r(y^{(0)}) = Jy^{(0)} + h^2Bf(y^{(0)}) - a \qquad \underline{3.7}$$

is small. The increments of the functions

$$r(y) = r(y^{(n+1)}) - r(y^{(n)})$$

are replaced by their differentials at the points $y = y^{(0)}$. The resulting linear system of equations is solved for the increment of the vector y, which is referred to as Δy. Then all that is necessary is that the system (3.6) be linearized. Jy is already linear and the remainder of the equation (3.6) can be linearized by defining the diagonal matrix

$$F(y) = \begin{bmatrix} f_y(x_1,y_1) & & & & 0 \\ & f_y(x_2,y_2) & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ 0 & & & & f_y(x_{N-1},y_{N-1}) \end{bmatrix}$$

where the differential of the vector r(y) at $y = y^{(0)}$ is $F(y^{(0)})\Delta y$. The equation (3.6) then reads

$$r(y^{(0)}) + (J + h^2 BF(y^{(0)}))\Delta y = 0 \qquad \underline{3.8}$$

which has as its solution

$$\Delta y = \Delta y^{(0)} = - A(y^{(0)})^{-1} r(y^{(0)}) \qquad \underline{3.9}$$

if the inverse of the matrix

$$A(y) = J + h^2 BF(y) \qquad \underline{3.10}$$

exists for $y = y^{(0)}$. Equation (3.8) is now linear. If $y^{(1)} = y^{(0)} + \Delta y^{(0)}$, then $y^{(1)}$ will be a better approximation to the exact solution, and repeating the process by letting $y^{(1)}$ take the place of $y^{(0)}$, etc., convergence can be attained.

It has been proved that equation (3.6) has a unique solution and that Newton's method produces a sequence of vectors $y^{(0)}, y^{(1)}, y^{(2)}, \cdots$, which converges rapidly to y provided the first approximation $y^{(0)}$ was well chosen.

To attain the solution of equation (3.9), it is necessary only to solve the system of linear equations

$$A(y^{(0)})\Delta y = - r(y^{(0)}) \qquad \underline{3.11}$$

for the Δy components.

Because the matrix $A(y^{(0)})$ is a tridiagonal matrix, the solution of the system (3.11) is relatively easy. The elements of the matrix $A(y^{(0)})$ can be written as

$$\begin{aligned} a_{n,n-1} &= -1 + h^2\beta_0 f_y(x_{n-1}, y_{n-1}^{(0)}), \quad n = 2,\cdots,N-1 \\ a_{n,n} &= 2 + h^2\beta_1 f_y(x_n, y_n^{(0)}), \quad n = 1,\cdots,N-1 \\ a_{n,n+1} &= -1 + h^2\beta_2 f_y(x_{n+1}, y_{n+1}^{(0)}), \quad n = 1,\cdots,N-2 \end{aligned} \qquad \underline{3.12}$$

where all other elements are zero. The procedure described in the discussion pertaining to the linear case is now applicable to this nonlinear case. This is what the author will refer to as Case 2.

It was assumed in the nonlinear case that $y^{(0)}$ was an approximation to the exact solution. The simplest way in which $y^{(0)}$ can be found is to solve the differential equation $y'' = Q(x,y)$ where $Q(x,y)$ is the part of $f(x,y)$ remaining after the nonlinear terms in y have been omitted. As one will notice in Case 2, this resulting differential equation can be solved by elementary methods. Solutions by algebraic methods of differential equations of the type $y'' = P(x)$ can be found in any elementary differential equations book, such as, Kells [(6)]. However, if the differential equation is of the type considered in Case 3, then it is not easy to find a first approximation to the exact solution. In terms of the notation used in Case 3, it is necessary to solve $y'' = yG(x) + P(x)$ to find $y^{(0)}$. It is readily seen that this is exactly like Case 1. However, Henrici has not discussed how to find $y^{(0)}$ in this case. Chapter IV of this thesis will point out how the author proposes to find the approximation $y^{(0)}$ and the notation changes necessary to enable the programming of the ideas set forth in Chapter I.

It is interesting to note that on page 227 of Hildebrand [2], a change of variables is found which enables one to transform any linear equation of the second order, of the form $Y'' + P(x)Y' + Q(x)Y = F(x)$, to the form $y'' + yf(x) = g(x)$, where $g(x) = e^{1/2\int P(x)dx}F(x)$ and $f(x) = 1/4(4Q(x) - 2P'(x) - (P(x))^2)$. The change of variables necessary to accomplish this is

$$Y(x) = e^{-1/2\int P(x)dx}y(x)$$

As one can readily see if this form is rewritten as $y'' = - yf(x) + g(x)$, then it corresponds exactly to the form $y'' = yG(x) + P(x)$ (Case 1) where $G(x) = - f(x)$ and $P(x) = g(x)$. From this it follows that the ideas developed in the present study extend to the solution of a class of linear second order differential equations containing a y' term.

CHAPTER IV

MODIFIED ALGORITHM

The author, in beginning work on the I.B.M. program for this thesis, decided to adopt a different manner of designating the constants, variables and functions, so that he could reduce the storage space. The following is a list of the changes:

A. The elements of matrix B were designated by β_0 = BO, β_1 = B1, and β_2 = B2 and used as constants.

B. The elements of matrix J were designated by TNM1 = - 1, TN = 2, and TNP1 = - 1 and used as constants.

C. The boundary conditions were designated by a = XO, b = XN, A = AA, and B = BB.

D. The function $f(x,y)$ was designated by $f(x,y) = yG(x) + P(x) + T(x)H(y)$.

E. The elements of matrix A were designated by $a_{n,n-1}$ = ANM1, $a_{n,n}$ = AN, and $a_{n,n+1}$ = ANP1.

F. The first approximation was designated by $y^{(0)}$ = YO.

G. The elements of matrix U were designated by u_{mn} = UN.

H. The elements of matrix L were designated by l_{mn} = EN.

I. The components of vector y were designated by y_n = YN if f(x,y) was linear and by y_n = YOO if f(x,y) was nonlinear.

J. The function G(x) was designated by GN.

In discussing Case 1, using the notation of the preceding paragraph, the elements of the tridiagonal matrix A can be expressed as

$$\begin{aligned} \text{AN(I)} &= \text{TN} + h^2*\text{B1}*\text{GN(I)}, \quad \text{I} = 1,\cdots,\text{N-1} \\ \text{ANM1(I)} &= \text{TNM1} + h^2*\text{BO}*\text{GN(I)}, \quad \text{I} = 1,\ldots,\text{N-2} \\ \text{ANP1(I)} &= \text{TNP1} + h^2*\text{B2}*\text{GN(I)}, \quad \text{I} = 2,\cdots,\text{N-1} \end{aligned} \qquad \underline{4.1}$$

which correspond to equations (3.5A). Equations (4.1) represent the elements of three n x 1 column matrices. To compute the values of the column matrices described by equations (4.1), it is necessary to define the n x 1 matrix

$$\text{GN(I)} = \text{G}(x), \quad \text{I} = 1,\cdots,\text{N-1}$$

where G(x) is the coefficient of the y term of the function f(x,y) (see subheading D). From the relationship of the functions F(1), F(N-1), A(I) and A(N-1) (which are defined in the same fashion as on page 11 of this thesis) and P(1) = P(N+1) = 0 and P(I) = P(x), I = 2,$\cdots$,N where P(x) is part of f(x,y) (see subheading D), the following function can be computed

$$\text{B(I)} = \text{A(I)} - h^2*(\text{BO}*\text{P(I)} + \text{B1}*\text{P(I+1)} + \text{B2}*\text{P(I+2)})$$
$$\text{I} = 1,\cdots,\text{N-1} \qquad \underline{4.2}$$

where the function B(I) corresponds to the b term of equation (3.4). It is now possible to summarize the procedure corresponding to the summary on pages 13-14, in the following manner in terms of the new notation:

(I) Compute UN(I) and Z(I), starting with I = 2, using the relations:

$$UN(1) = AN(1), \quad Z(1) = B(1)$$

$$EN(I-1) = ANM1(I-1)/UN(I-1)$$

$$UN(I) = AN(I) - EN(I-1)*ANP1(I)$$

$$A(I) = B(I) - EN(I-1)*Z(I-1)$$

$$I = 2,\cdots,N-1$$

(II) Compute YN(I), starting with I = N-1, using the relations:

$$YN(N-1) = Z(N-1)/UN(N-1)$$

$$YN(M) = (Z(M) - ANP1(M+1)*YN(M+1))/UN(M)$$

$$M = N-2,\cdots,1$$

The YN(M) from the above equation is then the solution of the problem as outlined in Case 1. As one can observe from the previous paragraph, there are few changes in the procedure.

To discuss Case 2, it is necessary to find a first approximation to the exact solution which is YO in terms of the new notation. The function described in Case 2 has no linear term in y, therefore, it is possible to find YO by some elementary method. Assume that YO has been found, it is then necessary to define the function GN(I) as the partial derivative of $f_y(x,y)$ (see subheading D) evaluated at the point (x,YO) as I goes from 1 to N-1, inclusive. With this function defined, it is now possible to write equations analogous to equations (3.12) using the new notation. The equations are of the same form as equations (4.1). The functions F(1), F(N-1), A(I) and A(N-1) are defined in the same fashion as on page 11 of this thesis. The function P(1) = P(N+1) = 0 and P(I) = f(x,y), I = 2,···,N is the function f(x,y) (see subheading D). In order that both linear and nonlinear functions can be computed with a minimum amount of difficulty, the residual vector $r(y^{(0)})$ has been referred to as -B(I) in this thesis. Since it is now necessary to compute $r(y^{(0)})$ in order that equation (3.11) can be solved, the function B(I) is computed from the relationship

$$B(I) = (S(I) + T(I) - A(I))$$
$$I = 1,\cdots,N-1$$

where

$$S(I) = (TNM1*YO(I) + TN*YO(I+1) + TNP1*YO(I+2))$$
$$T(I) = h^2*(BO*P(I) + B1*P(I+1) + B2*P(I+2))$$

The procedure as described on page 20 of this thesis is then immediately applicable and one can proceed to find the value of the components of y which are referred to as YOO.

In the nonlinear cases, it should be pointed out that an iteration procedure is used and continued until the components of $r(y^{(0)}) = 0$ to the number of digits as specified by the individual using this program.

The remaining idea to be discussed in this study is Case 3 of the problem outlined in Chapter III. As was pointed out in Chapter III, the approximation YO is not readily found by elementary methods. It is important to note that in Case 3, if the coefficient of y is a constant and P(x) is any function of x, then the first approximation YO can be found by elementary methods. If the above statement is true, then Case 3 can be solved using ideas set forth in Case 2. However, there are numerical methods for finding the approximation YO, such as the methods found in Chapter II, if the coefficients are all functions of x.

The author has not written an I.B.M. program for the methods found in Chapter II, except for Henrici's method. It is the author's intention to find the approximation YO (when it is necessary to use a numerical procedure) by employing the ideas set forth in Case 1. It has already been shown that the resulting differential equation to be solved by neglecting the nonlinear term in y in Case 3, is precisely Case 1. For this reason, the author has written the program as found in Appendix I and the subroutine subprograms as found in Appendix II.

CHAPTER V

RESULTS AND CONCLUSIONS

In numerical methods for solving any type of problem, the error analysis is a very important and necessary feature. The two types of error that occur in this thesis problem are the round-off error and the discretization error. The round-off error is defined by the equation

$$- \tilde{y}_{n-1} + 2\tilde{y}_n - \tilde{y}_{n+1} + h^2(\beta_0 f(x_{n-1},\tilde{y}_{n-1}) + \beta_1 f(x_n,\tilde{y}_n) + \beta_2 f(x_{n+1},\tilde{y}_{n+1})) = \varepsilon_n, \quad n = 1,2,\cdots,N-1 \qquad \underline{5.1}$$

where $\tilde{y}_n$ is the solution obtained by applying the algorithm as described in this thesis. The local round-off error (as defined by Henrici) may not only be due to the limitations of the particular computer but also to the fact that Newton's method was terminated after a finite number of terms. Some round-off error may result from the fact that the system (3.6) was not solved accurately for other reasons. The value of ε_n will be the round-off error in the literal sense and can be computed from equation (5.1).

The discretization error is defined to be $e_n = y_n - y(x_n)$, where y_n is the exact solution of equations (3.6) (without round-off) and $y(x_n)$ is the exact solution of the boundary value problem. As it will not be possible in the general case to find the exact solution of the boundary value problem, it is necessary to write the discretization error in another form. In accordance with Theorem 7.8, as well as the conditions implied in Theorem 7.7 of Henrici [1], the discretization error can be written in the form

$$|e_n| \leq ((x_n - a)(b - x_n)/2)(\bar{G}Zh^p + Kh^q)$$
$$n = 1,2,\cdots,N-1 \qquad \underline{5.2}$$

The $\bar{G}$ term is defined by $\bar{G} = \int_0^k |\bar{G}(s)|ds$ where $\bar{G}(s)$ is the kernel of the integral equation obtained from an integration of the differential equation. The major problem that arises now is in determining the kernel, $\bar{G}(s)$. If the function $f(x,y)$ is linear, then the difficulty can be overcome by solving for the kernel as described in Hildebrand [7]. In terms of the boundary value problem $y'' = yG(x) + P(x)$, $y(XO) = AA$ and $y(XN) = BB$, the kernel is found to be

$$\bar{G}(s) = K(x,\xi) = \begin{matrix} ((\xi - a)(x - b)/(b - a))G(\xi), & \xi < x \\ ((a - x)(b - \xi)/(b - a))G(\xi), & \xi > x \end{matrix}$$

which is clearly continuous when $\xi = x$.

The author has applied the modified algorithm to a boundary value problem that fit each of the cases as described in Chapter III. Each boundary value problem was chosen so that it had an analytic solution. The exact solution and the approximation to the solution were computed and compared. For the limited number of cases examined, the discretization error seems to be small.

However, before more confidence could be placed in the discretization error without applying equation (5.2), a larger number of problems with known solutions would have to be computed and the approximation to the solution compared with the exact solution. In any case, to determine a thoroughly reliable approximation to the discretization error, it is necessary to find the kernel of the associated integral equation.

Table I and Table II show the amount of compilation and computing time necessary to obtain solutions to a problem from each of the three cases outlined in Chapter III, using the method described in this thesis and in accordance with the flow chart on page 28. As one can notice, the times are small for finding 19 ordinates between two end points. Because only 40,000 positions of core storage are available, the author has been able to compute only 49 ordinates between two end points. The maximum number of ordinates that could be obtained with the present number of storage spaces is 57.

There are many other methods, such as those described in Chapter II, that could be used to find the first approximation YO when one is dealing with problems as described in Case 3. The author feels that if one would combine some of those methods with Henrici's, it would be possible to obtain better results. It is

TABLE I

COMPILATION TIMES

Nature of Program	Time
Load Fortran II Pass I	3 minutes
Load Fortran II Pass II	2 minutes
Mainline Program	8 minutes
Subprograms for linear case	6 minutes
Subprograms for nonlinear case with a y term	6 minutes
Subprograms for nonlinear case without a y term	6 minutes

TABLE II

COMPUTING TIMES

Boundary Value Problem	Number of Points Found	Time
$y'' = -y + 6\cos^2 x$		
$y(0) = 0, \quad y(\pi/2) = 0$	19	5 minutes
$y'' = -y + 2\cos x - x^2\sin^2 x + y^2$		
$y(0) = 0, \quad y(\pi/2) = \pi/2$	19	5 minutes
$y'' = 2 + x(x^2 - 1)^2 - xy^2$		
$y(0) = -1, \quad y(1) = 0$	19	5 minutes

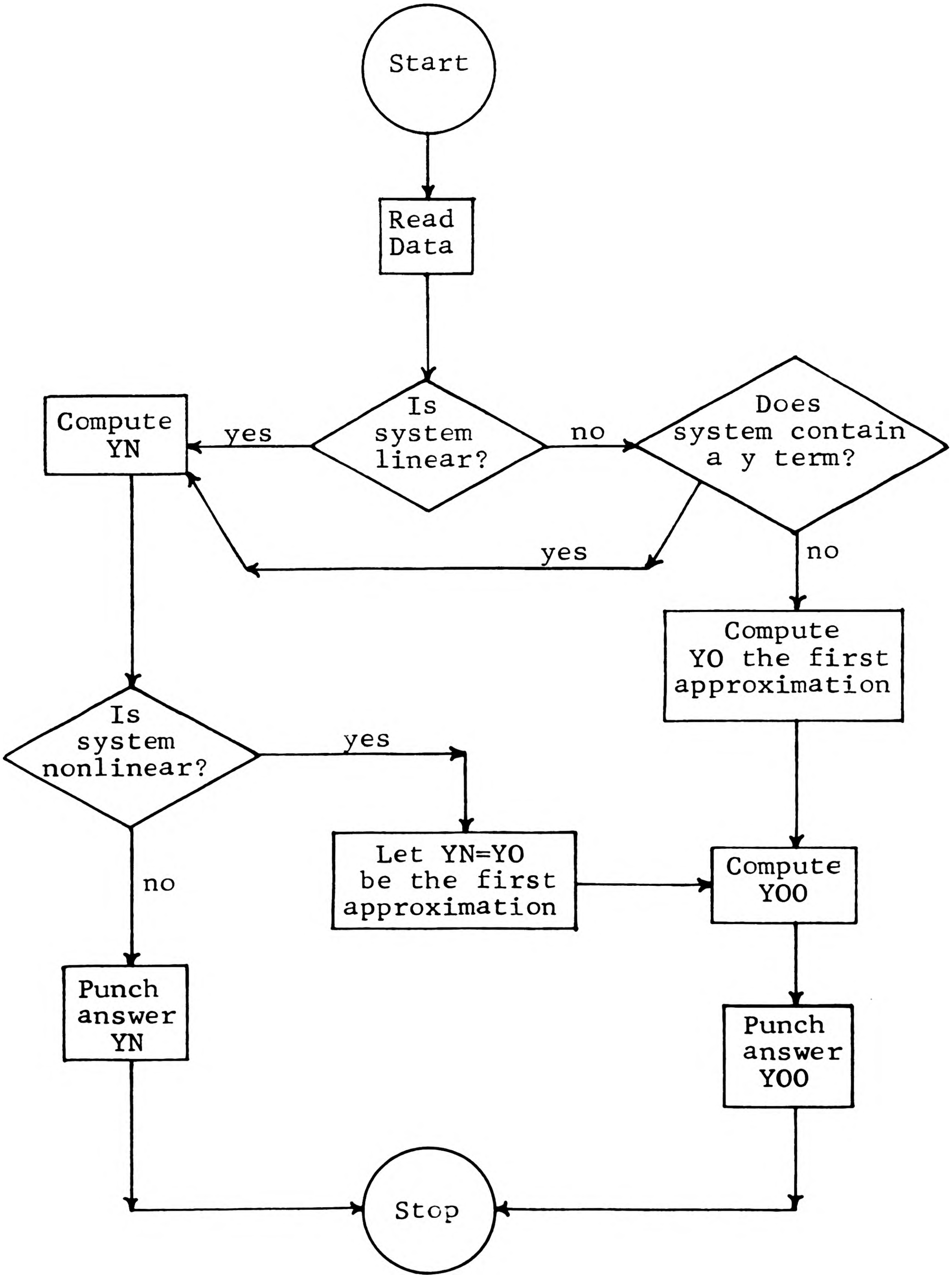

Fig. 1 Flow Chart

the opinion of the author, that further investigation into these possibilities would be of great value in extending the work pertaining to linear and nonlinear differential equations and that a more detailed study of the error analysis, particularly the error associated with the nonlinear differential equation, would benefit the study of boundary value problems.

BIBLIOGRAPHY

1. Henrici, Peter, (1962) Discrete Variable Methods in Ordinary Differential Equations, John Wiley, New York, p.345-388.

2. Hildebrand, F. B., (1956) Introduction to Numerical Analysis, McGraw-Hill, New York, p.188-257.

3. Struble, Raimond A., (1962) Nonlinear Differential Equations, McGraw-Hill, New York, 267 pages.

4. Davis, Harold T., (1960) Introduction to Nonlinear Differential and Integral Equations, United States Atomic Energy Commission, 566 pages.

5. Fox, L., (1957) The Numerical Solution of Two-Point Boundary Problems in Ordinary Differential Equations, Oxford University Press, London, p.67-105.

6. Kells, Lyman M., (1954) Elementary Differential Equations, McGraw-Hill, New York, 266 pages.

7. Hildebrand, F. B., (1952) Methods of Applied Mathematics, McGraw-Hill, New York, p.381-461.

APPENDIX I

THE SOURCE PROGRAM

```
C     Use format (4E18.8) for floating point data.
C     Use format (4I3) for fixed point data.
C     Load AA, BB, XO and XN, which are determined by
      y(XO) = AA and y(XN) = BB.
C     Load N, ISW1, ISW2, and ISW3 where N is the number
      of intervals.
C     ISW1 = 2 if BO = 0, B1 = 1 and B2 = 0 and ISW1 = 1
      if BO = 1/12, B1 = 10/12 and B2 = 1/12.
C     ISW2 = 1 if f(x,y) is linear and 2 if f(x,y) is
      nonlinear in y.
C     ISW3 = 1 if no y term and 2 if it contains a y
      term and 0 if linear.
C     Load TOL where TOL is tolerance set by individual
      if f(x,y) is nonlinear.
C     Values in dimension statement are dependent on
      storage space of computer.
      Dimension AN(xxx), ANP1(xxx), ANM1(xxx), GN(xxx+1)
      Dimension UN(xxx), B(xxx), A(xxx), F(xxx), P(xxx+1)
      Dimension Z(xxx), EN(xxx), YN(xxx), BL(xxx)
      Dimension YO(xxx+1), S(xxx), T(xxx), YOO(xxx+1)
      Read 5, AA, BB, XO, XN
      Read 25, N, ISW1, ISW2, ISW3
      XOO = XO
```

APPENDIX I (Con't.)

```
   ISW1 = (ISW1 - 1)
   ISW2 = (ISW2 - 1)
   ISW3 = (ISW3 - 1)
   If (ISW1) 1, 1, 2
 2 BO = 0.0
   B1 = 1.0
   B2 = 0.0
   Go to 3
 1 BO = 1.0/12.0
   B1 = 10.0/12.0
   B2 = 1.0/12.0
 3 FN = N
   TN = 2.0
   TNM1 = -1.0
   TNP1 = -1.0
   H = (XN - XO)/FN
   K = N - 1
   J = N - 2
   If (ISW2) 4, 4, 62
62 If (ISW3) 6, 6, 60
60 Call DIFF1 (GN, XO, H, K)
   XO = XOO
   Do 41 I = 1, K
41 AN(I) = TN + H*H*B1*GN(I)
```

APPENDIX I (Con't.)

```
      Do 45 I = 2, K
45    ANP1(I) = TNP1 + H*H*B2*GN(I)
      Do 46 I = 1, J
46    ANM1(I) = TNM1 + H*H*BO*GN(I)
      Call DIFF2 (P, XO, H, N)
      XO = XOO
      Call DIFF3 (F, XO, AA, 1)
      Call DIFF3 (F, XN, BB, K)
      P(1) = 0.0
      P(N+1) = 0.0
      Do 55 I = 2, J
      F(I) = 0.0
55    A(I) = 0.0
      A(1) = AA - BO*H*H*F(1)
      A(K) = BB - B2*H*H*F(K)
      Go to 7
 6    Call DIFF4 (YO, XO, H, N)
      XO = XOO
29    YO(1) = 0.0
      YO(N+1) = 0.0
      Read 5, TOL
      Call DIFF5 (GN, XO, H, YO, K)
      XO = XOO
      Go to 48
```

APPENDIX I (Con't.)

```
  4 Do 47 I = 1, N
 47 YO(I) = 0.0
    YO(N+1) = 0.0
    Call DIFF1 (GN, XO, H, K)
    XO = XOO
 48 Do 10 I = 1, K
 10 AN(I) = TN + H*H*B1*GN(I)
    Do 12 I = 2, K
 12 ANP1(I) = TNP1 + H*H*B2*GN(I)
    Do 13 I = 1, J
 13 ANM1(I) = TNM1 + H*H*BO*GN(I)
    If (ISW2) 14, 14, 99
 99 Call DIFF6 (P, XO, H, YO, N)
    XO = XOO
    Go to 246
 14 Call DIFF2 (P, XO, H, N)
    XO = XOO
246 P(1) = 0.0
    P(N+1) = 0.0
    If (ISW2) 72, 72, 73
 72 Call DIFF3 (F, XO, AA, 1)
    Call DIFF3 (F, XN, BB, K)
    Go to 74
 73 Call DIFF7 (F, XO, AA, 1)
```

APPENDIX I (Con't.)

```
   Call DIFF7 (F, XN, BB, K)
74 Do 30 I = 2, J
   F(I) = 0.0
30 A(I) = 0.0
   A(1) = AA - BO*H*H*F(1)
   A(K) = BB - B2*H*H*F(K)
   If (ISW2) 7, 7, 8
 7 Do 28 I = 1, K
28 B(I) = A(I) - (H*H)*(BO*P(I) + B1*P(I+1) + B2*P(I+2))
44 Z(1) = B(1)
   UN(1) = AN(1)
   Do 17 I = 2, K
   EN(I-1) = ANM1(I-1)/UN(I-1)
   UN(I) = AN(I) - EN(I-1)*ANP1(I)
17 Z(I) = B(I) - EN(I-1)*Z(I-1)
   YN(K) = Z(K)/UN(K)
   M = N - 2
18 YN(M) = (Z(M) - ANP1(M+1)*YN(M+1))/UN(M)
   M = M - 1
   If (M) 19, 19, 18
19 If (ISW2) 42, 42, 40
40 If (ISW3) 56, 56, 57
57 Do 59 I = 2, N
59 YO(I) = YN(I-1)
```

APPENDIX I (Con't.)

```
   ISW3 = (ISW3 - 1)
   Go to 29
42 Punch 5, (YN(I), I = 1, K)
   Go to 92
 8 Do 16 I = 1, K
   S(I) = (TNM1*YO(I) + TN*YO(I+1) + TNP1*YO(I+2))
   T(I) = H*H*(BO*P(I) + B1*P(I+1) + B2*P(I+2))
16 B(I) = - 1.0*(S(I) + T(I) - A(I))
   Go to 44
56 Do 20 I = 2, N
20 YOO(I) = YO(I) + YN(I-1)
   YOO(1) = 0.0
   YOO(N+1) = 0.0
   Call DIFF8 (P, XO, H, YOO, N)
   XO = XOO
   Do 22 I = 1, K
   S(I) = (TNM1*YOO(I) + TN*YOO(I+1) + TNP1*YOO(I+2))
   T(I) = H*H*(BO*P(I) + B1*P(I+1) + B2*P(I+2))
22 BL(I) = - 1.0*(S(I) + T(I) - A(I))
   Do 35 I = 1, K
   If (ABSF(B(I) - BL(I)) - TOL) 35, 35, 24
35 Continue
   Go to 23
24 Do 33 I = 2, N
```

APPENDIX I (Con't.)

```
33 YO(I) = YOO(I)
   Go to 48
23 Punch 5, (YOO(I), I = 2, N)
 5 Format (4E18.8)
25 Format (4I3)
92 Stop
   End
```

APPENDIX II

SUBROUTINE SUBPROGRAM

```
C      For linear case, define DIFF1 - DIFF3 as specified.
C      Define all other subprograms as zero.
C      For nonlinear case with no y term, define
       DIFF4 - DIFF8 as specified.
C      Define all other subprograms as zero.
C      For nonlinear case with y term, define
       DIFF1 - DIFF8 as specified.
C      All terms of subprogram refer to equation
       f(x,y) = G(x)*y + P(x) + T(x)*H(y)
C      where H(y) is nonlinear in y.
C      Read each comment card preceding each subprogram
       before starting.
C      GN(I) is coefficient of y term in f(x,y) and in
       terms of x.
       Subroutine DIFF1 (GN, X, H, N)
       Dimension GN(xxx+1)
       Do 102 I = 1, N
       X = X + H
   102 GN(I) = G(x)
       Return
       End
C      P(I) is the function P(x) in f(x,y) in terms of
       x when f(x,y) is linear.
```

APPENDIX II (Con't.)

```
C     P(I) = 0 if f(x,y) is nonlinear, with or without
      a y term.
      Subroutine DIFF2 (P, X, H, N)
      Dimension P(xxx+1)
      Do 103 I = 2, N
      X = X + H
  103 P(I) = P(x)
      Return
      End
C     F(N) = G(x)*y + P(x) in terms of x and y.
      Subroutine DIFF3 (F, X, Y, N)
      Dimension F(xxx)
      F(N) = G(x)*y + P(x)
      Return
      End
C     YO(I) is approximation to the second derivative of
      f(x,y) without the
C     nonlinear term of y if G(x) = 0, and in terms of x.
C     If G(x) is not zero, then YO(I) is zero.
      Subroutine DIFF4 (YO, X, H, N)
      Dimension YO(xxx+1)
      Do 104 I = 2, N
      X = X + H
  104 YO(I) = YO(x), approximation to solution.
```

APPENDIX II (Con't.)

```
      Return
      End
C     GN(I) is partial derivative of f(x,y) with respect
      to y.
C     If f(x,y) has a nonlinear term, then GN(I) is in
      terms of x and YO(I+1).
C     If f(x,y) has no nonlinear term, then GN(I) = 0.
      Subroutine DIFF5 (GN, X, H, YO, N)
      Dimension GN(xxx+1), YO(xxx+1)
      Do 105 I = 1, N
      X = X + H
  105 GN(I) = ∂f(x,y)/∂y
      Return
      End
C     P(I) = f(x,y) in terms of x and YO(I) if f(x,y) is
      nonlinear.
C     P(I) = 0 if f(x,y) is linear.
      Subroutine DIFF6 (P, X, H, YO, N)
      Dimension P(xxx+1), YO(xxx+1)
      Do 106 I = 2, N
      X = X + H
  106 P(I) = f(x,y)
      Return
      End
```

APPENDIX II (Con't.)

```
C     F(N) = f(x,y) in terms of x and y if f(x,y) is
      nonlinear.
C     F(N) = 0 if f(x,y) is linear.
      Subroutine DIFF7 (F, X, Y, N)
      Dimension F(xxx)
      F(N) = f(x,y)
      Return
      End
C     P(I) = f(x,y) in terms of x and YOO(I) if nonlinear
      and P(I) = 0 if linear.
      Subroutine DIFF8 (P, X, H, YOO, N)
      Dimension P(xxx+1), YOO(xxx+1)
      Do 107 I = 2, N
      X = X + H
  107 P(I) = f(x,y)
      Return
      End
```

VITA

The author was born on September 28, 1937, in Lake View Heights, Missouri. He received his primary and secondary education in Sedalia, Missouri; his college education at Central Missouri State College in Warrensburg, Missouri, the University of Missouri in Columbia, Missouri, and the University of Missouri School of Mines and Metallurgy, Rolla, Missouri. In May 1958, he received a Bachelor of Science degree in Education, Major in Mathematics, from Central Missouri State College.

Since September, 1958, he has been employed as an Instructor in Mathematics at the University of Missouri School of Mines and Metallurgy.