SOME ASPECTS OF THE SOLUTION OF NON-LINEAR DIFFERENTIAL EQUATIONS

by

G. R. SLATER

Submitted for the Degree of Master
of Philosophy in Computer Science of
the University of Aston in Birmingham.

October 1972.

C O N T E N T S

SUMMARY.

Nonlinear differential equations can be solved by a variety of methods. One such technique is to transform the differential equations into a set of algebraic equations.

The thesis describes the methods available at present for solving large systems of algebraic equations. The methods described are found to be ~~deficient~~ not to be entirely satisfactory and therefore a new technique for solving algebraic equations is presented. The technique is specifically developed for solving large systems of equations whilst simplifying the computation and minimizing the storage requirements.

Next, a study has been made to provide a suitable standard of comparisons between the various methods.

Finally, an industrial problem is solved using the new technique. Comparisons show that the above technique provides an adequate solution at reduced computational effort.

ACKNOWLEDGEMENTS.

I wish to thank Mr.R.S.S.Wee, my supervisor for his ready and willing assitance, valuable advice and encouragement throughout the course of the work.

Mr.R.H.Bannister who was extremely helpful in providing the rigid rotor problem together with test data ~~and his own calculated solution~~.

Mrs.M.Hasleton who typed the whole of this thesis.

LIST OF SYMBOLS.

$\underset{\sim}{A}$	-	Auxiliary Function
$\underset{\sim}{B}$	-	Jacobian Estimate
$\underset{\sim}{b}$	-	back substitution vector
C	-	Comparison Index
$\underset{\sim}{f}$	-	function vector
$\underset{\sim}{g}$	-	function vector in Brown's Method
h	-	increment
$\underset{\sim}{H}$	-	Inverse Jacobian Estimate
$\underset{\sim}{J}$	-	Jacobian
n	-	number of variables
$\underset{\sim}{p}$	-	increment vector
σ	-	Inverse weighted storage ratio ($^1/S$)
S	-	weighted storage ratio
x	-	variable

C H A P T E R 1.

INTRODUCTION.

1.1) Outline of Problem.

The study of nonlinear systems has been increasing in importance as engineers design equipment to perform at greater efficiency than had been hitherto possible. New and expensive materials as well as a greater understanding of the physical system are contributory factors in this direction. The physical characteristics of a number of nonlinear systems have been examined and recorded by the ingenious instrumentation techniques of engineers. This, in turn, has lead mathematicians to suggest techniques to solve a number of problems related to nonlinear systems. However, in many instances, the methods developed cannot be readily applied to the general class of nonlinear problems.

In view of this gap that exists between theoretical analysts and engineers, it is felt that this present thesis could serve a useful function by co-ordinating an aspect of the range of mathematical methods available with the basic requirement of simple applicability to a class of nonlinear problems.

The class of nonlinear problems examined is taken from the field of nonlinear vibrations and, in particular, the harmonic responses resulting from such a system. Predominantly harmonic responses are most often required from even highly nonlinear problems. All the methods examined were considered with the view of its ease of application and suitability on a digital computer. This would certainly fulfill some, if not all, of the requirements demanded by practising engineers.

1.2) Method of Approach.

There is a large variety of methods that exist in the

1.2) contd.

field of optimization and root-finding for a system of algebraic equations. Thus it is felt that it would certainly be desirable to exploit these techniques for the solution of the set of differential equations which describe a nonlinear system. ~~Thus~~ The method of approach to solve a nonlinear vibration problem falls into two phases. The first phase is to transform the set of nonlinear differential equations into a corresponding algebraic equivalent and the next phase is to solve the resulting algebraic system by a variety of root finding methods.

In addition the thesis suggests a standard by which comparisons may be made between the various methods in spite of their different requirements. The suggestion of a standard available for comparative purposes is considered desirable and could very well provoke other standards to be offered.

1.3) Structure of Thesis.

Chapter 1 outlines a basic problem in the field of nonlinear vibration problems and gives a macro view of the method of approach to obtain a harmonic response resulting from a nonlinear vibration problem. The structural content of the thesis is also presented.

Chapter 2 gives a survey of the class of methods used in nonlinear vibration problems. A detailed description of a number of existing root finding methods are presented and discussed.

Chapter 3 specifies the class of vibration problems examined and the particular approach adopted by the author.

Chapter 4 presents a new method of root finding. The method is described in detail and applied to a number of test examples.

1.3) contd.

Chapter 5 makes a comparative study of the relative merits and demerits of the root finding methods. Test examples are used to indicate how the various methods performed in the light of the suggested standard for comparative evaluations.

Chapter 6 is concerned essentially with the phase of transforming the nonlinear differential equations into its algebraic equivalent. The Ritz-Averaging method, the Harmonic Balancing method and the Energy Balancing method were used to transform a nonlinear rotor dynamic problem into its algebraic equivalent.

Chapter 7 is concerned with solving an industrial problem and the performance of the method presented.

Chapter 8 discusses the type of stability that exist in nonlinear systems. Reference is made to the particular problem considered.

Chapter 9 suggests future developments for the method presented and also recent developments in the field. Finally future developments on the solution of nonlinear differential equations are commented on.

C H A P T E R 2.

SURVEY OF PUBLISHED LITERATURE.

2.1) Introduction.

Many physical systems can often be represented by sets of simultaneous differential equations. These differential equations can be divided into those that are linear and those that are nonlinear. When the equations are linear analytic solutions are often possible while nonlinear systems are invariably solved by non-analytic or numerical methods. In the linear case the solution for the system is an exact and general solution. However, the solution for a nonlinear system is not a general solution but only a particular solution in a specific range of values.

If the nonlinearities in the differential equation are small and the equations are of a special type, analytic methods may be used to yield sufficiently accurate results. Often assumptions can be made which lead to simplification and pseudo-linearization of these equations. In such cases, classical methods may be applied to obtain a solution. While the simplified equations do not describe the problem exactly they do, however, represent the salient features of the physical system. The study of these modified equations will give information which can be used to give starting values for a more comprehensive study of the nonlinear equations.

If the nonlinearities are large then solution may only be possible by non-analytic methods. These methods of solution vary considerably in their approach to the problem and they can be separated into two distinct categories. Continuous methods of solution fall into the first category whilst discrete methods of solution form this second category. Analogue computer simulations and phase-plane methods are examples of the first type of method giving solutions in the complete field of operation. The phase-plane methods are

2.1) contd.

essentially graphical in their implementation and are suitable for system consisting of one or two parameters. For large systems the construction of the graphs becomes cumbersome and the interpretation of the graphs becomes difficult. The philosophy of analogue computer methods of solution is one of continuous integration. Computing elements in analogue machines are inherently only accurate to within certain limits. Care has to be taken to ensure that these elements operate within certain specified values to avoid inaccuracies that will creep in at both the low and high end of the voltage range. In addition, as the steady state solution is the required solution, the analogue computer has to be run until the transient solution has disappeared. This may cause drift errors. However with proper time scaling these drift difficulties can be minimized. Analogue computers themselves may be simulated on digital computers using programming languages such as Slang[1] or Kaldas[2]. These languages and others like them do not suffer from limitations of equipment or drift errors. The simulation of the integration process of such methods do, however, pose a problem. The integration process on a digital machine has to be performed in a discrete manner and in certain cases instability can ari se due to the integration process rather than the instability of the system. However, this problem is not exclusive to languages such as Slang and Kaldas but an inherent problem in all digital simulations.

Nonlinear differential equations can be reduced to a set of algebraic equations by using the Ritz-Balancing method, Harmonic Balancing and others. The set of algebraic equations can then be solved by a variety of methods but most

2.1) contd.

of these methods will only generate particular solutions.

Solution to these algebraic equations can be found by finding the maximum or minimum of the objective function subject to the constraint that all the functions are zero at the solution. General methods of optimization which find these maxima or minima do not ~~usually~~ necessarily locate the roots of these algebraic equations. In this context they are considered to be less efficient than the methods now presented.

A Taylor series expansion for a simple function of a single variable takes the form

$$f(x+h) = f(x) + h\, f'(x) + \frac{h^2}{2!} f''(x) + \ldots$$

and ignoring terms of second and higher order gives

$$f(x+h) = f(x) + h\, f'(x) \tag{2.1}$$

Now at the solution x^{k+1} say, $f(x^{k+1})$ is zero so that if $x^{k+1} = x+h$, then using equation (2.1) with $x = x^k$ gives

$$x^{k+1} = x^k - \frac{f(x^k)}{f'(x^k)} \tag{2.2}$$

which is Newton's one dimensional method for solving an algebraic equation. The derivative in equation (2.2), $f'(x^k)$, can be expressed analytically or can be evaluated using function approximations given by equation (2.1). This means that equation (2.2) can now be written as

$$x^{k+1} = x^k - \left[\frac{f(x^{k+1}) - f(x^k)}{h}\right]^{-1} f(x^k) \tag{2.3}$$

where $h = x^{k+1} - x^k$. It is possible to choose h differently than above, for if $h = x^{k-1} - x^k$ then

2.1) contd.

$$x^{k+1} = x^k - \left[\frac{f(x^{k-1}) - f(x^k)}{x^{k-1} - x^k}\right]^{-1} f(x^k) \qquad (2.4)$$

which is the secant method. The n-dimensional generalizations to these one dimensional methods and the modifications of these generalizations are now to be considered.

2.2) Newton's method[3].

Suppose a set of n functions in n unknowns takes the form

$$\begin{aligned}
f_1(x_1,x_2, \ldots ,x_i, \ldots ,x_n) &= 0\\
f_2(x_1,x_2, \ldots ,x_i, \ldots ,x_n) &= 0\\
&\vdots\\
f_i(x_1,x_2, \ldots ,x_i, \ldots ,x_n) &= 0\\
&\vdots\\
f_n(x_1,x_2, \ldots ,x_i, \ldots ,x_n) &= 0
\end{aligned}$$

If the first function is expanded using a Taylor series expansion about the point $(x_1,x_2, \ldots,x_n)$ the following equation is obtained

$$f_1(x_1+h_1,x_2+h_2,\ldots x_i+h_i, \ldots x_n+h_n) = f_1(x_1,x_2,\ldots x_i \ldots x_n)+h_1\frac{\partial f_1}{\partial x_1} + h_2 \frac{\partial f_1}{\partial x_2} + \ldots h_i \frac{\partial f_1}{\partial x_i} + \ldots + h_n \frac{\partial f_1}{\partial x_n}$$

if terms of the second and higher order are ignored. Similarly if we expand $f_2,\ldots f_i \ldots f_n$ the following system is generated

$$\begin{aligned}
f_1(x_1+h_1,\ldots x_i+h_i \ldots x_n+h_n) &= f_1(x_1 \ldots x_i \ldots x_n)+h_1\frac{\partial f_1}{\partial x_1}+\ldots h_i\frac{\partial f_1}{\partial x_i}+\ldots h_n \frac{\partial f_1}{\partial x_n}\\
&\vdots\\
f_i(x_1+h_1,\ldots x_i+h_i \ldots x_n+h_n) &= f_i(x_1 \ldots x_i \ldots x_n)+h_1 \frac{\partial f_i}{\partial x_1}+\ldots h_i\frac{\partial f_i}{\partial x_i}+\ldots h_n \frac{\partial f_i}{\partial x_n}\\
&\vdots\\
f_n(x_1+h_1,\ldots x_i+h_i \ldots x_n+h_n) &= f_n(x_1 \ldots x_i \ldots x_n)+h_1 \frac{\partial f_n}{\partial x_1}+\ldots h_i \frac{\partial f_n}{\partial x_i}+\ldots h_n \frac{\partial f_n}{\partial x_n}
\end{aligned}$$

2.2) contd.

or in vector form

$$\begin{bmatrix} f_1 \\ f_2 \\ \cdot\cdot \\ f_i \\ \cdot\cdot \\ f_n \end{bmatrix}_{(x+h)} = \begin{bmatrix} f_1 \\ f_2 \\ \cdot\cdot \\ f_i \\ \cdot\cdot \\ f_n \end{bmatrix}_{x} + \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_i} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \cdot\cdot & \cdot & \cdot\cdot & \cdot & \cdot\cdot \\ \frac{\partial f_i}{\partial x_1} & \cdots & \frac{\partial f_i}{\partial x_i} & \cdots & \frac{\partial f_i}{\partial x_n} \\ \cdot\cdot & \cdot & \cdot\cdot & \cdot & \cdot\cdot \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_i} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \cdot\cdot \\ h_i \\ \cdot\cdot \\ h_n \end{bmatrix}$$

The $(n \times n)$ matrix is the Jacobian matrix, $\underset{\sim}{J}$, so that the above can be rewritten

$$\underset{\sim}{f}(\underset{\sim}{x} + \underset{\sim}{h}) = \underset{\sim}{f}(\underset{\sim}{x}) + \underset{\sim}{J}(\underset{\sim}{x})\underset{\sim}{h} \qquad (2.5)$$

where $\underset{\sim}{f}$ is the vector of functions $f_1, f_2 \ldots f_n$, $\underset{\sim}{x}$ is the vector of variables $x_1, x_2 \ldots x_n$ and $\underset{\sim}{h}$ is the vector of differences $h_1, h_2 \ldots h_n$.

Again if the solution is at the point $\underset{\sim}{x}^{k+1}$, say, so that $\underset{\sim}{f}(\underset{\sim}{x}^{k+1})$ is zero and $\underset{\sim}{x}^{k+1} = \underset{\sim}{x} + \underset{\sim}{h}$ then using equation (2.5) with $\underset{\sim}{x} = \underset{\sim}{x}^{k}$ then

$$\underset{\sim}{x}^{k+1} = \underset{\sim}{x}^{k} - \underset{\sim}{J}^{-1}(\underset{\sim}{x}^{k})\, \underset{\sim}{f}(\underset{\sim}{x}^{k}) \qquad (2.6)$$

which is Newton's n-dimensional method. Newton's method as expressed in equation (2.6) suffers from serious disadvantages from the point of view of practical calculation. Major problem centre round the calculation of the Jacobian matrix and its inverse. Lohr and Rall[4] suggested that the Jacobian should be evaluated only once every few iterations instead of an evaluation at every iterative step as is strictly required. This variant, however, requires the complete evaluation of the Jacobian matrix. Broyden[5] describes a class of methods in which the partial derivatives are not estimated or evaluated directly, but corrections to an approximate inverse of the Jacobian matrix are computed from values of the vector function $\underset{\sim}{f}$.

2.3) Broyden's Method.

Broyden makes a simple modification to Newton's method so that there is a greater circle of convergence. For each increment in Newton's method it is necessary to calculate the Jacobian matrix and obtain its inverse. To avoid a full matrix inversion Broyden suggests that it was more convenient to estimate an estimate for the inverse Jacobian, $\underset{\sim}{J}^{-1}$, and shows he can obtain it relatively simply. If a vector $\underset{\sim}{\rho}$ is defined such that

$$\underset{\sim}{\rho} = - \underset{\sim}{B}^{-1} \underset{\sim}{f} \qquad (2.7)$$

where $\underset{\sim}{f}$ is the vector of functions and $\underset{\sim}{B}$ is an estimate for the Jacobian $\underset{\sim}{J}$, then Newton's method can be rewritten as

$$\underset{\sim}{x}^{i+1} = \underset{\sim}{x}^{i} + \underset{\sim}{\rho}^{i} \qquad (2.8)$$

Convergence will only occur if we are close enough to the solution so a simple modification to equation (2.8) will give $\underset{\sim}{x}^{i+1}$ by

$$\underset{\sim}{x}^{i+1} = \underset{\sim}{x}^{i} + t^{i} \underset{\sim}{\rho}^{i}$$ where

t^{i} is a scalar chosen to prevent the process diverging. Let us now define the variable $\underset{\sim}{x}$ as

$$\underset{\sim}{x} = \underset{\sim}{x}^{i} + t \underset{\sim}{\rho}^{i}$$

where $\underset{\sim}{x}^{i}$, $\underset{\sim}{\rho}^{i}$ will have particular values and t is a variable quantity. The vector $\underset{\sim}{f}$ will now be functions of the variable t and Broyden shows that it is possible to use the functions to obtain an estimate of the Jacobian. The Jacobian matrix contains terms which take the form $\frac{\partial f_j}{\partial x_k}$ but the $\underset{\sim}{f}$'s are now functions of the single variable t so that only derivatives of the form $\frac{df_j}{dt}$ are available. Using the chain rule[6]

2.3) contd.

$$\frac{df_j}{dt} = \sum_{k=1}^{n} \frac{\partial f_j}{\partial x_k} \frac{dx_k}{dt} \qquad j = 1 \ldots n$$

we now have the following relationship

$$\frac{d\underset{\sim}{f}}{dt} = \underset{\sim}{J}\, \underset{\sim}{\rho}^i \tag{2.9}$$

It is now required to obtain an approximation to $\underset{\sim}{J}$ at the point $\underset{\sim}{x}^{i+1}$, so if each f_j is expanded as a Taylor series about the point t^i

$$f_j(t^i - s) \simeq f_j(t^i) - s \frac{df_j(t^i)}{dt} + O(s^2)$$

and since $\underset{\sim}{f}^{i+1} = \underset{\sim}{f}(\underset{\sim}{x}^{i+1}) = f(\underset{\sim}{x}^i + t^i \underset{\sim}{\rho}^i) = \underset{\sim}{f}(t^i)$

and the vector $\underset{\sim}{f}$ are functions of t alone, then

$$\underset{\sim}{f}(t^i - s) = \underset{\sim}{f}^{i+1} - s \frac{d\underset{\sim}{f}}{dt} \tag{2.10}$$

From equations (2.9) and (2.10) we obtain

$$\underset{\sim}{f}(t^i - s) \simeq \underset{\sim}{f}^{i+1} - s\, \underset{\sim}{J}\, \underset{\sim}{\rho}^i \tag{2.11}$$

At the i^{th} iterate Broyden uses the notation $\underset{\sim}{B}^i$ as the approximate Jacobian $\underset{\sim}{J}$. To improve on the approximation Broyden uses equation (2.11) and suggests that a better approximation to the Jacobian $\underset{\sim}{J}$ is now $\underset{\sim}{B}^{i+1}$. This results in the following equation

$$\underset{\sim}{f}(t^i - s^i) = \underset{\sim}{f}^{i+1} - s^i\, \underset{\sim}{B}^{i+1}\, \underset{\sim}{\rho}^i \tag{2.12}$$

where s^i is a particular value of s chosen at each iteration to minimize the error of the estimate of $\frac{d\underset{\sim}{f}}{dt}$. Broyden stated that his philosophy was to find an estimate to the inverse Jacobian $\underset{\sim}{J}^{-1}$ and not an estimate to the Jacobian. If we now

2.3) contd.

define an estimate to the inverse Jacobian $\underset{\sim}{H}$, say, as

$$\underset{\sim}{H}^i = (\underset{\sim}{B}^i)^{-1}$$

and we also define a vector, $\underset{\sim}{y}^i$, such that

$$\underset{\sim}{y}^i = \underset{\sim}{f}^{i+1} - \underset{\sim}{f}(t^i - s^i)$$

then using equation (2.12) we obtain the following relationship

$$\underset{\sim}{y}^i = s^i \underset{\sim}{B}^{i+1} \underset{\sim}{\rho}^i \qquad (2.13)$$

Using equations (2.7) and (2.13) we can now obtain the relationships

$$\underset{\sim}{\rho}^i = \underset{\sim}{H}^i \underset{\sim}{f}^i$$

and

$$\underset{\sim}{H}^{i+1} \underset{\sim}{y}^i = - s^i \underset{\sim}{\rho}^i$$

which defines a class of methods, based upon Newton's method for solving nonlinear algebraic equations.

Broyden now describes a particular class of methods which results when different assumptions are made on $\underset{\sim}{H}^{i+1}$ or $\underset{\sim}{B}^{i+1}$, t^i and s^i. He makes comparisons between these various methods using standard test examples and concludes that one particular method is superior to the other variations. He calls this particular method the full step reducing variation. In this t^i is chosen to reduce the norm. If the first value of t^i chosen results in the norm being reduced then it is the only possible value for s^i if there are going to be no further function evaluations. If the first t^i is not chosen then a choice for s^i is possible but experience shows that $s^i = t^i$ is the best choice for a reducing step method. When $s^i = t^i$ the increment is called a "full-step". It remains to place restrictions on $\underset{\sim}{B}^{i+1}$ so that it can be defined uniquely in equation (2.13). As no information is available about the

2.3) contd.

change in $\underset{\sim}{f}$ when $\underset{\sim}{x}$ is changed in a direction different from that of vector $\underset{\sim}{\rho}^i$, $\underset{\sim}{B}^{i+1}$ is chosen so that the change in $\underset{\sim}{f}$ predicted by $\underset{\sim}{B}^{i+1}$ in the direction $\underset{\sim}{q}^i$ which is orthogonal to $\underset{\sim}{\rho}^i$, is the same as that which would be predicted by $\underset{\sim}{B}^i$. That is to say

$$\underset{\sim}{B}^{i+1}\,\underset{\sim}{q}^i = \underset{\sim}{B}^i\,\underset{\sim}{q}^i$$

where $$(\underset{\sim}{q}^i)^T\,\underset{\sim}{\rho}^i = 0$$

together with equation (2.12) defines $\underset{\sim}{B}^{i+1}$ uniquely as

$$\underset{\sim}{B}^{i+1} = \underset{\sim}{B}^i + \frac{(\underset{\sim}{y}^i - s^i\,\underset{\sim}{B}^i\,\underset{\sim}{\rho}^i)(\underset{\sim}{\rho}^i)^T}{s^i\,(\underset{\sim}{\rho}^i)^T\,\underset{\sim}{\rho}^i}$$

It is now possible to use Householder's formula[7] to express $\underset{\sim}{H}^{i+1}$, the inverse of $\underset{\sim}{B}^{i+1}$, in the same terms as above giving

$$\underset{\sim}{H}^{i+1} = \underset{\sim}{H}^i - \frac{(s^i\,\underset{\sim}{\rho}^i + \underset{\sim}{H}^i\,\underset{\sim}{y}^i)(\underset{\sim}{\rho}^i)^T\,\underset{\sim}{H}^i}{(\underset{\sim}{\rho}^i)^T\,\underset{\sim}{H}^i\,\underset{\sim}{y}^i} \qquad (2.14)$$

Equation (2.14) now defines the full step reducing method which can be expressed in the following algorithm:-

1. Obtain an initial estimate $\underset{\sim}{x}^0$ to the solution.
2. Obtain an initial value $\underset{\sim}{H}^0$, the iteration matrix.
3. Compute $\underset{\sim}{f}^i = \underset{\sim}{f}(\underset{\sim}{x}^i)$
4. Computer $\underset{\sim}{\rho}^i = \underset{\sim}{H}^i\,\underset{\sim}{f}^i$
5. Select a value t^i such that the norm of $\underset{\sim}{f}(\underset{\sim}{x}^i + t^i\,\underset{\sim}{\rho}^i)$ is less than the norm of $\underset{\sim}{f}(\underset{\sim}{x}^i)$.

   $\underset{\sim}{f}(\underset{\sim}{x}^{i+1})$ will be calculated during this step.
6. Test if $\underset{\sim}{x}^{i+1}$ is a solution point. Yes, go to step 10.
7. Computer $\underset{\sim}{y}^i = \underset{\sim}{f}^{i+1} - \underset{\sim}{f}^i$

2.3) contd.

8. Compute $H^{i+1} = H^i - \dfrac{(\underset{\sim}{H}^i \underset{\sim}{y}^i + t^i \underset{\sim}{\rho}^i)(\underset{\sim}{\rho}^i)^T \underset{\sim}{H}^i}{(\underset{\sim}{\rho}^i)^T \underset{\sim}{H}^i \underset{\sim}{y}^i}$

9. Return to step 4.

10. Stop. Solution has been found.

2.4) Wolfe's n-dimensional secant method.

The one dimensional secant method derived previously is

$$x^{k+1} = x^k - \left[\frac{f(x^{k-1}) - f(x^k)}{x^{k-1} - x^k}\right]^{-1} f(x^k)$$

In 1959, Wolfe[8], developed a secant method based on the property of a straight line. If we consider the one dimensional method as shown in Fig.1.

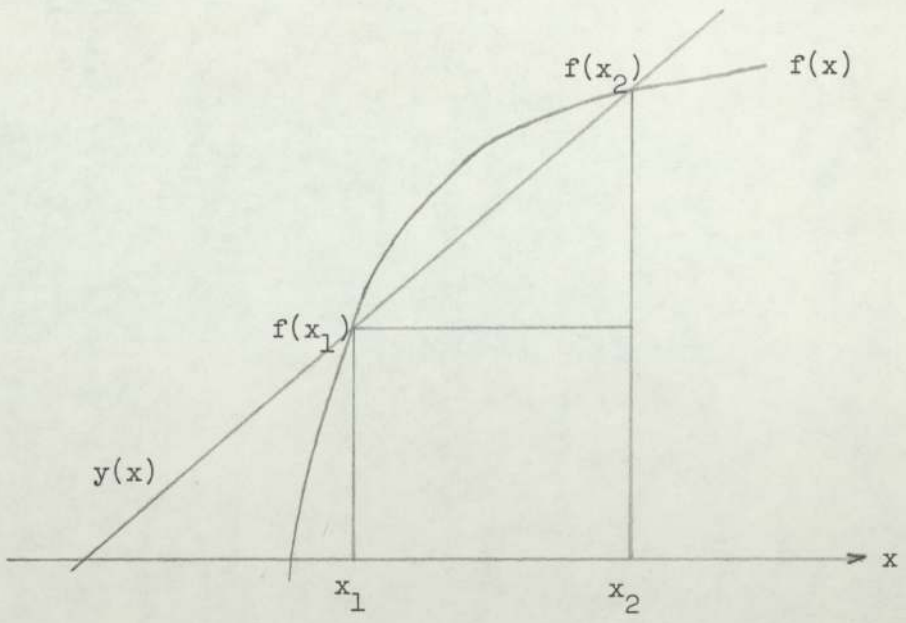

Fig: 1

then the next estimate to x is obtained by drawing the secant through $f(x_1)$ and $f(x_2)$ and evaluating the point at which this line crosses the x-axis. The general form of this line is $y = mx + c$ where m is the gradient and c is a constant. Using

2.4) contd.

the above equation with the two starting values x_1 and x_2 it can be easily shown that

$$x = \frac{f(x_2)}{f(x_2) - f(x_1)}\, x_1 \; - \frac{f(x_1)}{f(x_2) - f(x_1)}\, x_2$$

or $\quad x = \pi_1 x_1 + \pi_2 x_2$

where $\quad \pi_1 = \dfrac{f(x_2)}{f(x_2) - f(x_1)}$ and $\pi_2 = \dfrac{-f(x_1)}{f(x_2) - f(x_1)}$

In addition we can also derive, the following relationships

$$\pi_1 + \pi_2 = 1$$

and $\qquad \pi_1 f(x_1) + \pi_2 f(x_2) = 0$

Wolfe now generalizes these relationships to obtain his n-dimensional secant method. This means he defines a vector $\underset{\sim}{\pi}$ such that

$$\sum_{j=1}^{n+1} \pi_j = 1 \qquad (2.15)$$

and $\qquad \sum_{j=1}^{n+1} \pi_j \underset{\sim}{f}(x^j) = 0$

which will lead to a new set of trial solutions given by

$$\underset{\sim}{x}^* = \sum_{j=1}^{n+1} \pi_j \underset{\sim}{x}^j$$

where some $\underset{\sim}{x}^j$ is to be replaced by $\underset{\sim}{x}^*$ for which $\| \underset{\sim}{x}^j \|$ is maximal according to

$$\| \underset{\sim}{x}^j \| = \sum_{i=1}^{n} |f_i(x^j)|^2 \qquad \text{for } j = 0 \ldots n$$

2.4) contd.

Computationally Wolfe expressed this in a simple scheme of four steps. The first step is to form the (n+1) by (n+1) matrix $\underset{\sim}{A}$ given by

$$A_j = \begin{bmatrix} 1 \\ f_1(\underset{\sim}{x}^j) \\ \vdots \\ f_n(\underset{\sim}{x}^j) \end{bmatrix} \qquad j = 1, \ldots (n+1)$$

noting the column with the largest norm. Solve for $\underset{\sim}{A}^{-1}$ and use the fact

$$\underset{\sim}{A}\,\underset{\sim}{\pi} = (1, 0, \ldots 0)^T \qquad \text{from (2.15)}$$

calculate $\underset{\sim}{\pi} = \underset{\sim}{A}^{-1}(1,0 \ldots 0)^T$

It is now possible to calculate a new point $\underset{\sim}{x}^* = \sum_{j=1}^{n+1} \pi_j \underset{\sim}{x}^j$ as the second step and test if the norm is less than the tolerance. The third step is to form a column vector $\underset{\sim}{\rho}$, of the form $\underset{\sim}{A}_j$, to be inserted in $\underset{\sim}{A}$ in place of the column with the largest norm. It is possible to calculate the new inverse, $\underset{\sim}{A}^{*-1}$, required by using the relationships

$$(A^{*-1})_{\rho j} = (A^{-1})_{\rho j} \Big/ q_\rho \qquad j = 1, \ldots (n+1)$$

$$(A^{*-1})_{ij} = (A^{-1})_{ij} - (A^{*-1})_{\rho j}\, q_j \qquad i \neq \rho$$

$$j = 1 \ldots (n+1)$$

where $\underset{\sim}{q} = \underset{\sim}{A}^{-1}\,\underset{\sim}{\rho}$. If we now note the column with the largest norm in $\underset{\sim}{A}^*$ we can return to calculate a new $\underset{\sim}{\pi}$ and continue the iteration.

The three methods that had just been presented are

2.4) contd.

all simple applications of a Taylor series expansion. They all approach the solution by considering a vectorial increment of the dependent variables. A method in which each variable is considered separately is going to be considered.

2.5) Brown and Conte's Method.

Brown and Conte[9] suggest a method which considers each variable and generates each increment to that variable separately. They calculate expressions for increments using Taylor series expansions. When expressions for each increment become available, the method uses the Gauss-Seidel technique to solve for the independent variables. The value of a norm at this new point, $\underset{\sim}{x}^{i+1}$, can now be calculated and tested for convergence. The method is described by the following algorithm:-

Step 1.

Expand $f_1(\underset{\sim}{x})$, where $\underset{\sim}{x}$ is the solution point, using a Taylor series expansion about the point $\underset{\sim}{x}^n$ retaining only first order terms and thus obtaining the linear approximation

$$f_1(\underset{\sim}{x}) \sim f_1(\underset{\sim}{x}^n) + \frac{\partial f_1(\underset{\sim}{x}^n)}{\partial x_1} h_1^n + \frac{\partial f_1(\underset{\sim}{x}^n)}{\partial x_2} h_2^n + \ldots + \frac{\partial f_1(\underset{\sim}{x}^n)}{\partial x_N} h_N^n \qquad (2.16)$$

As $\underset{\sim}{x}$ is the solution point and the nearest approximation is $\underset{\sim}{x}^n$ then

$$\underset{\sim}{x} = \underset{\sim}{x}^n + \underset{\sim}{h}^n$$

which means that (2.16) can be rewritten as

$$f_1(\underset{\sim}{x}) \sim f_1(\underset{\sim}{x}^n) + \frac{\partial f_1(\underset{\sim}{x}^n)}{\partial x_1}(x_1 - x_1^n) + \frac{\partial f_1(\underset{\sim}{x}^n)}{\partial x_2}(x_2 - x_2^n) + \ldots + \frac{\partial f_1(\underset{\sim}{x}^n)}{\partial x_N}(x_N - x_N^n) \quad (2.17$$

Equate the right hand side of (2.17) to zero and solve for that variable (x_N say) whose corresponding partial derivative is largest in absolute value. Thus

2.5) contd.

$$x_N = x_N^n - \sum_{j=1}^{N-1} \frac{\frac{\partial f_1(\underline{x}^n)}{\partial x_j}}{\frac{\partial f_1(\underline{x}^n)}{\partial x_N}} (x_j - x_j^n) - \frac{f_1(\underline{x}^n)}{\partial f_1(\underline{x}^n)/\partial x_N} \qquad (2.18)$$

The constant terms $\frac{\partial f_1/\partial x_j}{\partial f_1/\partial x_N}$, $j = 1,..,N-1$ and $\frac{f_1}{\partial f_1/\partial x_N}$ are saved for future use. The left hand side of (2.18) is now renamed as b_N a function of $(x_1, x_2 \ \ldots \ x_{N-1})$ giving

$$b_N(x_1 \ldots x_{N-1}) = x_N^n - \sum_{j=1}^{N-1} \frac{\partial f_1(\underline{x}^n)/\partial x_j}{\partial f_1(\underline{x}^n)/\partial x_N} (x_j - x_j^n) - \frac{f_1(\underline{x}^n)}{\partial f_1(\underline{x}^n)/\partial x_N} \qquad (2.19)$$

and also

$$x_N = b_N(x_1, x_2 \ \ldots \ x_{N-1})$$

Step 2.

If this expression for x_N is substituted into the second equation f_2 a new function g_2 say, of (N-1) variables $x_1, \ldots x_{N-1}$ is defined as

$$g_2 = f_2(x_1 \ \ldots \ x_{N-1}, b_N)$$

If the procedure described in the first step is now followed and g_2 is expanded, using a Taylor series, this time about the point $(x_1^n \ \ldots\ x_{N-1}^n)$ then solving for the variable, x_{N-1} say, whose corresponding partial derivative is largest in magnitude we obtain the expression

$$x_{N-1} = x_{N-1}^n - \sum_{j=1}^{N-2} \frac{\partial g_2/\partial x_j}{\partial g_2/\partial x_{N-1}} (x_j - x_j^n) - \frac{g_2}{\partial g_2/\partial x_{N-1}}$$

Again if we rewrite the left hand side this time as b_{N-1}, a function of N-2 variables, we have

$$b_{N-1}(x_1 \cdots x_{N-2}) = x_{N-1}^{n} - \sum_{j=1}^{N-2} \frac{\partial g_2/\partial x_j}{\partial g_2/\partial x_{N-1}} (x_j - x_j^{n}) - \frac{g_2}{\partial g_2/\partial x_{N-1}} \qquad (2.20)$$

and $x_{N-1} = b_{N-1}(x_1 \cdots x_{N-2})$.

Again the ratios formed, $\frac{\partial g_2/\partial x_j}{\partial g_2/\partial x_{N-1}}$, $j = 1, \ldots, N-2$

and $\frac{g_2}{\partial g_2/\partial x_{N-1}}$ are saved for future use.

Step 3.

The expression for x_{N-1} is now substituted into f_3 to give a definition to g_3, at function of $x_1, \ldots x_{N-2}$, as

$$g_3 = f_3(x_1, x_2 \cdots x_{N-2}, b_{N-1}, b_N)$$

where b_{N-1}, b_N are defined by (2.20), (2.19). This process is repeated until the N^{th} step is reached. At this stage there is

$$g_N = f_N(x_1, b_2, b_3 \cdots b_N)$$

where the b_i's are obtained by back substitution in the (N-1) triangularized linear system which takes the form

$$b_i = x_i^{n} - \sum_{j=1}^{N-i} \frac{\partial g_i/\partial x_j}{\partial g_i/\partial x_{N-i+1}} (x_j - x_j^{n}) - \frac{g_i}{\partial g_i/\partial x_{N-i+1}} \qquad (2.21)$$

Now expanding, linearizing and solving for x_1 yields

$$x_1 = x_1^{n} - \frac{g_N}{\partial g_N/\partial x_1}$$

We use the point x_1 thus obtained as the improved approximation x_1^{n+1} to the first component of the root x_1 and call it b_1. Back-solve the b_i system (2.12) to obtain approximations to the other components. We note that the most recent information available is used immediately in the construction of

2.5) contd.

the next component, as in the Gauss-Seidel process[10]. Having calculated a new point we can test for convergence or if further iterations are required return to Step 1.

Brown has shown that under the hypothesis for Newton's method this process is well defined and it is considered that this method offers a suitable alternative to the other methods which have been presented.

A summary of the general methods of solution for nonlinear differential equations is given in Fig.2.

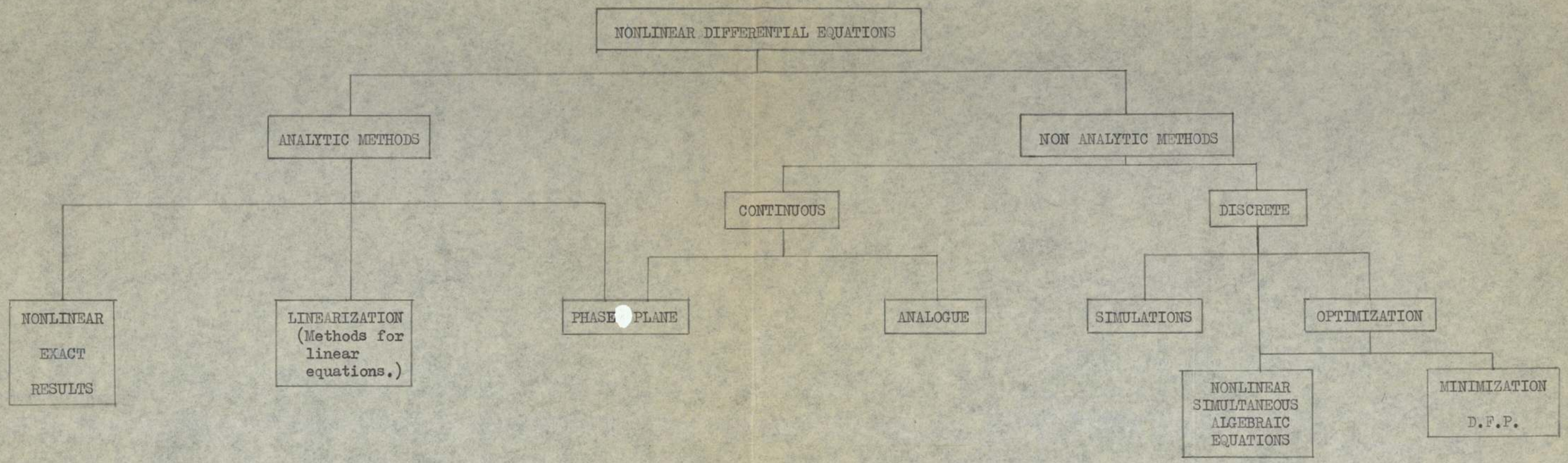
NONLINEAR DIFFERENTIAL EQUATIONS
ANALYTIC METHODS
NON ANALYTIC METHODS
NONLINEAR EXACT RESULTS
LINEARIZATION (Methods for linear equations.)
PHASE PLANE
CONTINUOUS
ANALOGUE
DISCRETE
SIMULATIONS
OPTIMIZATION
NONLINEAR SIMULTANEOUS ALGEBRAIC EQUATIONS
MINIMIZATION
D.F.P.

FIGURE 2

CHAPTER 3

SPECIFICATION OF PROBLEM.

The differential equations which describe a vibration problem will in general contain inertia, damping, restoring forces and excitation terms. When the stiffness and damping terms are linear the equations of motion can be solved by the classical analytic methods. However, if the stiffness and/or damping terms are nonlinear then the resulting nonlinear equations of motion can, in general, only be solved by approximate methods. In a few special cases, however, these can be solved by conventional analytic techniques. The general type of nonlinear differential equation expressing these problems can be expressed in general mathematical terms as

$$\underset{\sim}{F}(\underset{\sim}{x}, \dot{\underset{\sim}{x}}, \ddot{\underset{\sim}{x}}) = \underset{\sim}{0}$$

where F, x, $\dot{x}$, $\ddot{x}$ are all vectors.

Typical methods for solving equations of this type is to use the Ritz-Averaging method or Harmonic Balancing method to generate a set of corresponding algebraic equations which take the form

$$\underset{\sim}{f}(\underset{\sim}{x}) = \underset{\sim}{0}$$

The number of independent variables in the algebraic equations is dependent upon the order of accuracy required in the solution. To obtain a solution to these algebraic equations one can apply a number of root finding techniques some of which have been described in the previous chapter. Consideration from the general engineering viewpoint to these methods of solution is such that a considerable effort has to be put into the understanding and application of these algebraic methods which is not necessarily justified from the practical standpoint. These criticisms are relevant to the methods previously described.

Newton's method is regarded as the traditional method

of solution for these algebraic equations. The method cannot guarantee convergence to a solution and if a solution is successfully found progress towards the solution could be erratic. The computational effort required by the method is prohibitive as every step in the iteration process requires the evaluation of a Jacobian and its inversion, both of which are very costly numerical processes. Generally, although Newton's method suffers from these drawbacks, it is still used because of the simplicity of application.

Broyden made several modifications to Newton's method to ensure convergence and at the same time reduce the computational effort. A Jacobian is uneconomic to evaluate at every step of the iteration so Broyden, after having obtained an initial estimate to the inverse of the Jacobian, improves his previous estimate to the inverse of the Jacobian rather than calculate a new one. It is important to note that while Broyden avoided evaluating a new inverse Jacobian his method still requires a significant computational effort in making his estimate to the inverse of the Jacobian. Nevertheless, this still represents a reduction in the computational effort with respect to a full inversion. Noting that Newton's method cannot guarantee convergence and its path towards finding a solution can be unpredictable, Broyden decided that his method must converge to a solution in a more uniform manner. This is achieved by calculating a norm value at every iteration, this norm having to satisfy the condition that it has been reduced after every step of the iteration.

Wolfe's method of solution is similar to Broyden's being the development of the secant method where Broyden is the development of Newton's method. Wolfe, in his method, does not calculate the inverse of the Jacobian, but merely updates the inverse Jacobian from its previous value. Wolfe's method has an inherent norm reduction facility since it is based on simple interpolation techniques. Although Wolfe's method has desirable characteristics it

cannot be considered as good as Broyden's method. In the first place Wolfe's method requires a number of starting values rather than one starting value as with the other methods. Secondly its rate of convergence is slow compared to Broyden's method because the secant method on which Wolfe is based has a significantly slower rate of convergence than Newton's method on which Broyden is based.

Brown's method is another variant of Newton's method but this particular method does not require the evaluation of a Jacobian matrix. In view of the fact that Brown's method does not require the use of a Jacobian matrix a considerable amount of storage space has been saved. However, Brown's method, requires the construction of intermediate or axuiliary functions. The construction of these functions is complicated, involving the removal and substitution of a variable in every stage of the iteration process. It does not seem possible to program Brown's algorithm conveniently in a high-level scientific language like Fortran IV though it may be a feasible proposition in PL1. However PL1 language is not readily available and is confined to only a small family of computers. This, of course, is not an attractive alternative to an engineer. When all the intermediate functions have been constructed a Gauss-Seidel iteration is used to solve for the variables. Back substitution is used in the Gauss process so half the storage saved in having no Jacobian will be used for the back substitution. Convergence to a solution is not guaranteed but care is taken in the construction of the intermediate function to reduce the erratic tendancies of Newton's method.

In this thesis an alternative method of solution for these nonlinear simultaneous algebraic equations is presented. The method of solution is a modification of Newton's method and the desirable characteristics of the methods already discussed so far

have been incorporated into the structure of the method. As in Brown's method the method presented does not require the evaluation of the Jacobian matrix. This type of approach means there is a significant reduction in the storage requirements and that there are no lengthy calculations centred around the construction and inversion of a Jacobian matrix. Linear interpolation is used to improve the iterated values at every stage of the iterative process thus inducing a steady convergence towards the solution. These improved values are used at the time they are calculated unlike Newton's method which does not use improved values until all the variables have an improved value. Again this contributes to improving the convergence to the solution. Norm reduction, as in Wolfe and Broyden's methods at the end of every iterative step is considered to be desirable as it ensures a predictable convergence to the required solution.

Every individual section of the algorithm presented was developed separately, thus enabling comparisons as to the effectiveness of each modification to be undertaken. Following this comparisons were made between the method presented and the other four methods. Four specific test examples were used, all of which had been designed to test the characteristics and convergence factors of the method. These results were used to show the general characteristic of the method presented. An industrial problem was then solved using the Ritz method to generate the algebraic equations and then solving these equations by the method presented. This application enables a comprehensive appreciation of the technique to be undertaken.

C H A P T E R 4.

NEW METHOD

4.1) Introduction.

The methods already described for solving nonlinear simultaneous algebraic equations can, in theory, be used for very large systems. These Quasi-Newton methods attempt to increase their circle of convergence by modifications to Newton's method. In these methods every itration generates an update for the vector $\underset{\sim}{x}$ using the elements of the inverse Jacobian or a suitable estimate to the inverse Jacobian. This matrix which is used only once per iterative step causes considerable problems in the large amount of storage it requires.

The object of this chapter is to develop a method which is reliable and which possesses a rate of convergence equal to that of existing methods whilst attempting to minimize on storage requirements for a large system.

4.2) Description of the method.

It is required to solve the system of equations $\underset{\sim}{f}(\underset{\sim}{x}) = \underset{\sim}{0}$. The method makes use of auxiliary functions which take the form

$$A_i^* = A_{i-1} + f_i \qquad i = 1, \ldots, n \qquad (4.1)$$

where $A_0 = 0$ and where A_i^* is an auxiliary function, n is the number of variables and A_{i-1} is the most recent value of the previous auxiliary function, A_{i-1}^*. If we expand (4.1) then:-

$$\begin{aligned} A_1^* &= f_1 \\ A_2^* &= A_1 + f_2 \\ A_3^* &= A_2 + f_3 \\ &\cdots \\ A_n^* &= A_{n-1} + f_n \end{aligned}$$

At the solution $\underset{\sim}{f}(\underset{\sim}{x}) = \underset{\sim}{0}$, $\underset{\sim}{A}(\underset{\sim}{x})$ will also be zero. Thus we can solve the equations

4.2) contd.

$$\underset{\sim}{A}(\underset{\sim}{x}) = \underset{\sim}{0}$$

instead of $\underset{\sim}{f}(\underset{\sim}{x}) = \underset{\sim}{0}$. The following algorithm describes the method.

Step 1.

Expand $A_1^*(\underset{\sim}{x})$ using a Taylor series expansion about the point $\underset{\sim}{x}^n$ to obtain a unidirectional expansion in x_1. If we retain only first order terms then

$$A_1^*(\underset{\sim}{x}) \sim A_1^*(\underset{\sim}{x}^n) + \frac{\partial A_1^*(\underset{\sim}{x})}{\partial x_1} \; h_1^n$$

where $\underset{\sim}{x} = \underset{\sim}{x}^n + \underset{\sim}{h}^n$ so we can rewirte the above as

$$A_1^*(\underset{\sim}{x}) \sim A_1^*(\underset{\sim}{x}^n) + \frac{\partial A_1^*(\underset{\sim}{x}^n)}{\partial x_1} (x_1 - x_1^n) \qquad (4.2)$$

Equating the right hand side of (4.2) to zero and solving for the variable x_1 gives

$$x_1 = x_1^n - \frac{A_1^*(\underset{\sim}{x}^n)}{\partial A_1(\underset{\sim}{x}^n)/\partial x_1}$$

This gives an improved value for x_1 which gives a vector $(x_1^{n+1}, x_2^n \ldots x_n^n)$. This vector is used to calculate a new value for A_1. If there is no change in the sign of the auxiliary function after the iterative step or the auxiliary function has been successfully reduced to zero then it is possible to proceed to the next step. On the other hand if there has been a change in sign so that

$$\text{sign}[A_1(x_1^{n+1}, x_2^n)] \neq \text{sign}[A_1^*(x_1^n, x_2^n \ldots x_n^n)]$$

then linear interpolation is performed such that

4.2) contd.

$$x_1^{(n+1)}_{\text{new}} = x_1^{(n+1)}_{\text{old}} - \frac{|A_1(x_1^{n+1}, x_2^{n} \ldots x_n^{n})|(x_1^{(n+1)} - x_1^{n})_{\text{old}}}{|A_1(x_1^{n+1}, x_2^{n} \ldots x_n^{n})| + |A_1^{*}(x_1^{n} \ldots x_n^{n})|}$$

and this new value for x_1 is used to replace x_1^{n}. The use of linear interpolation is an attempt to prevent the ~~solution point~~ overshoot of the variable x_1 ~~overshooting~~. At the end of the first step two values will be available:-

(i) a new value x_1^{n+1}

(ii) a new value $A_1(x_1^{n+1}, x_2^{n} \ldots x_i^{n} \ldots x_n^{n})$

Step 2.

It is now required to construct the auxiliary function A_2^{*}. This will take the form

$$A_2^{*}(x_1^{n+1}, x_2^{n} \ldots x_n^{n}) = A_1(x_1^{n+1}, x_2^{n+1}, x_2^{n} \ldots x_n^{n}) + f_2(x_1^{n+1}, x_2^{n} \ldots x_n^{n})$$

It is now possible to expand $A_2^{*}(\underset{\sim}{x})$ using a Taylor series about the point $(x_1^{n+1}, x_2^{n} \ldots x_n^{n})$ to obtain a uni-directional expansion in x_2. Retaining first order terms only

$$A_2^{*}(\underset{\sim}{x}) \sim A_2^{*}(x_1^{n+1}, x_2^{n} \ldots x_n^{n}) + \frac{\partial A_2^{*}(x_1^{n+1}, x_2^{n} \ldots x_n^{n})}{\partial x_2} h_2^{n}$$

where $\underset{\sim}{x} = \underset{\sim}{x}^{n} + \underset{\sim}{h}^{n}$ so we can rewrite the above as

$$A_2^{*}(\underset{\sim}{x}) \sim A_2^{*}(x_1^{n+1}, x_2^{n} \ldots x_n^{n}) + \frac{\partial A_2^{*}(x_1^{n+1}, x_2^{n} \ldots x_n^{n})}{\partial x_2} (x_2 - x_2^{n}) \quad (4.3)$$

Equating the right hand side of (4.3) to zero and solving for the variable x_2 gives

4.2) contd.

$$x_2 = x_2^{n} - \frac{A_2^*(x_1^{n+1}, x_2^{n} - x_n^{n})}{\partial A_2^*/\partial x_2}$$

This gives an improved value for x_2 giving a vector $(x_1^{n+1}, x_2^{n+1}, \ldots x_n^{n})$. This vector is used to calculate a new value for A_2. Again linear interpolation is used if there is a change in the sign of the auxiliary function. At the end of this step a vector of the form $(x_1^{n+1}, x_2^{n+1} \ldots x_n^{n})$ is available and a value for A_2 using this vector. This procedure is followed until we reach the n^{th} step. On entry to the n^{th} step a vector $(x_1^{n+1}, x_2^{n+1} \ldots x_{n-1}^{n+1}, x_n^{n})$ will be available together with a value of A_{n-1} using this vector.

n^{th} step.

Again we require the auxiliary function A_n^* to be evaluated. This takes the form

$$A_n^*(x_1^{n+1} \ldots x_{n-1}^{n+1}, x_n^{n}) = A_{n-1}(x_1^{n+1}, \ldots x_{n-1}^{n+1}, x_n^{n}) + f_n(x_1^{n+1} \ldots x_{n-1}^{n+1}, x_n^{n})$$

It is now possible to expand $A_n^*(\underset{\sim}{x})$ using a Taylor series about the point $(x_1^{n+1}, \ldots x_{n-1}^{n+1}, x_n^{n})$ to obtain a uni-directional expansion in x_n. Retaining first order terms

$$A_n^*(\underset{\sim}{x}) \sim A_n^*(x_1^{n+1} \ldots x_{n-1}^{n+1}, x_n^{n}) + \frac{\partial A_n^*) x_1^{n+1} \ldots x_{n-1}^{n+1}, x_n^{n})}{\partial x_n} \; h_n^{n}$$

where $h_n^{n} = x_n - x_n^{n}$. This means the above can be rewritten as

$$A_n^*(\underset{\sim}{x}) \sim A_n^*(x_1^{n+1} \ldots x_{n-1}^{n+1}, x_n^{n}) + \frac{\partial A_n^*(x_1^{n+1} \ldots x_{n-1}^{n+1}, x_n^{n})}{\partial x_n} (x_n - x_n^{n}) \quad (4.4)$$

Equating the right hand side of (4.4) to zero and

4.2) contd.

solving for the variable x_n gives

$$x_n = x_n^n - \frac{A_n^*(x_1^{n+1} \ldots x_{n-1}^{n+1}, x_n^n)}{\partial A_n^*/\partial x_n}$$

This gives an improved value for x_n giving a vector $(x_1^{n+1} \ldots x_n^{n+1})$ or $\underset{\sim}{x}^{n+1}$. This vector is used to calculate a new value for A_n and linear interpolation is used if there has been a change in sign with A_n^*. A vector $\underset{\sim}{x}^{n+1}$ is now available.

The method now requires that the value of a norm, $\mathfrak{F}^2$, at the new point $\underset{\sim}{x}^{n+1}$ is less than the value of the same norm at the starting point $\underset{\sim}{x}^n$ i.e.

$$\| \underset{\sim}{f}(\underset{\sim}{x}^{n+1}) \| \leq \| \underset{\sim}{f}(\underset{\sim}{x}^n) \|$$

If this condition has been satisfied then it is possible to return to the first step immediately but if it is not the norm is reduced by a simple reduction procedure. If there has been an increase in the norm the assumption made is that one or more of the variables has overshot their solution points. This reduction process constructs a ratio of the two norms, less than one, and every variable is moved to a position inbetween its starting and finishing value in the ratio of the norm so that

$$\underset{\sim}{x} = \underset{\sim}{x}_{end} - \frac{\text{Norm}_{begin}}{\text{Norm}_{end}} \left(\underset{\sim}{x}_{end} - \underset{\sim}{x}_{begin} \right)$$

If the variables have overshot its solution then this interpolation procedure will have the desired effect of reducing the norm. In the majority of cases this is what has happened when the new norm is greater than the old norm. If no variables have overshot

4.2) contd.

its solution then the method appears to have found a point at which it can reduce the norm no further. This type of point is called a local solution. When such a position has been found the reduction procedure used above is again invoked returning the iteration process to the value at the start of that particular iteration $\underset{\sim}{x}_{begin}$. When this has been achieved a different procedure is used which is described in one of the following sections. When the norm has been successfully reduced it is possible to test if $\underset{\sim}{x}^{n+1}$ is the solution and if not return to start the process over again.

4.3) Developments and Modifications.

Tests on simple examples shows that general improvements can be incorporated into the method with very little extra calculation or modification. Several minor modifications are made and each one is discussed separately together with an example.

4.3.1) Construction of the increment.

From the Taylor series expansion

$$\text{increment} = - \frac{\text{Auxiliary Function}}{\text{Derivative of the Auxiliary Function}} \qquad (4.5)$$

It can be seen from this expression that if $\frac{\partial A_i}{\partial x_i}$ becomes small then the value of the increment will become very large. The increment could then be modified by a linear interpolation procedure if there is a change in the sign of the Auxiliary Function. If linear interpolation is used the value of the increment will become very small returning that particular variable to its value at the beginning of the iteration. If interpolation is not used the increment

4.3.1) contd.

is large and will overshadow the other increments directing the course of the iteration which is totally undesirable. Hence, at this stage, two types of increment have been tried. An increment of the form

$$\text{increment} = -\frac{\text{Auxiliary Function}}{\left\{\text{Sign of derivative}\right\}\left\{\text{Absolute value}(A_i)+\text{Absolute value}\left(\frac{\partial A_i}{\partial x_i}\right)\right\}}$$

is now used. The value of this increment is approximately one (if the derivative is zero then the increment is one) and this is an intermediate value for the increment. This form of increment is only used when there has been three successive increases in the value of the increment given by (4.5). An example of the use of this modification is shown in the following

$$f_1 = x_1^3 x_2 + x_2^2 - 6$$

$$f_2 = 2x_1^2 + x_2^3 + 25$$

The derivative of the auxiliary function A_1 with respect to x_1 is zero at the point (0,-3) so if the process uses the starting point (-2,-2) then with no modification the following results are obtained.

x_1	x_2	inc_1	inc_2	f_1	f_2	NORM
-2.00	-2.00	-	-	14.00	25.00	821.0
-1.42	-2.48	0.58	-0.48	7.17	13.85	243.3
-0.94	-3.14	0.48	-0.66	6.42	- 4.10	58.0
-0.16	-3.05	0.78	-0.08	3.33	- 3.39	22.6
-0.15	-3.05	0.01*	0.00	3.31	- 3.31	21.9
-0.15	-3.05	15.10	0.00	3.31	- 3.31	21.9

Fig: 3

*Here the auxiliary function A_1 changed sign so linear interpolation was used. The actual calculated value of the increment before interpolation was 15.00.

4.3.1) contd.

As the increment for x_1 is so large no further progress can be made. If the modification is now used the iteration process proceeds as follows:-

x_1	x_2	inc_1	inc_2	f_1	f_2	NORM
-2.00	-2.00	-	-	14.00	25.00	821.0
-1.42	-2.48	0.58	-0.48	7.17	13.85	243.3
-0.94	-3.14	0.48	-0.66	6.42	- 4.10	58.0
-0.16	-3.09	0.78	0.04	3.59	- 4.58	33.9
-0.16	-3.05	0.01*	0.04	3.32	- 3.33	22.1
0.78	-3.04	0.94	0.01	1.76	- 1.77	6.2
1.00	-3.00	0.22	0.04	0.01	- 0.02	5.0×10^{-4}

Fig: 4

which now converges to the solution (1,-3).

4.3.2) Norm reduction.

Norm reduction is used to overcome overshoot if the solution straddles the two calculated vectors. It becomes particularly noticeable towards the latter stages of the iteration process when the calculated vector is relatively close to the solution. In an effort to overcome this oscillation, usually illustrated by a change in the sign of the increments, the value of the increments changing sign is halved. If this oscillation is persistent in the process then the number of total iterations is reduced if this modification is employed. If there is no oscillation and the increment is halved, the iteration process will still tend towards the solution making good the loss in the next iterative step. The following example illustrates the success of halving the increment.

4.3.2) contd.

$$f_1 = 3x_1 + x_2 + 2x_3^2 - 3$$

$$f_2 = -3x_1 + 5x_2^2 + 2x_1x_3 - 1$$

$$f_3 = 25x_1x_2 + 20x_3 + 12$$

Starting Point	Number of iterations	
	Before Adjustment	After Adjustment
-1, -1, 0	31	14
0, -1, -1	27	8
0, 0, -1	9	11
0, 0, 0	9	10

Fig: 5

4.3.3) Local solution.

When a local solution has been found usually the value of the increment is not large enough to find a point where the norm at the new point is less than the norm at the local solution. It is possible to illustrate this situation by considering a diagram of the norm as drawn below

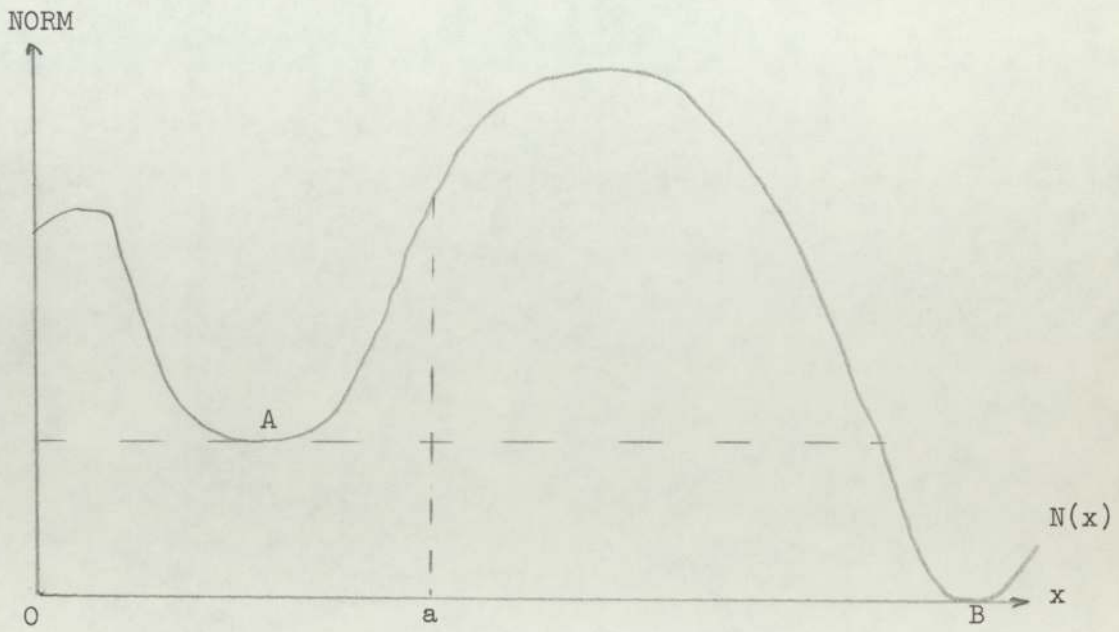

Fig: 6

If the iteration is in the region 0 → a then it is

4.3.3) contd.

very probable that the process will converge to the point A which has the smallest value of the norm in this range. However, the correct solution is at the point B. The value of the increment will be too small to move the vector $\underset{\sim}{x}$ corresponding to point A away from point A. In this case the method, as stated, breaks donw. The technique to overcome this problem is to make a search for a value where the norm is smaller than its present value. At present, the technique used to overcome this difficulty is to use Newton's method for one iteration to move the point away from the local solution. This has the effect of allowing the iteration process to continue in the normal manner. Incorporating these modifications into the basic method, the method present can be expressed by the following algorithm.

Step 1 Obtain an initial estimate $\underset{\sim}{x}^{\circ}$ to the solution.

Step 2 Calculate $\underset{\sim}{f}(\underset{\sim}{x}^{\circ})$ and the corresponding norm $\sum_{i=1}^{n} \underset{\sim}{f}(\underset{\sim}{x}^{\circ})^{2}$

Step 3 Calculate increments for the variables using the Auxiliary functions. At each stage use linear interpolation, if required, to combat overshoot.

Step 4 Compute $\underset{\sim}{f}^{i+1} = \underset{\sim}{f}(x^{i+1}$

Step 5 Compute the new norm $\sum^{n} \underset{\sim}{f}(\underset{\sim}{x}^{i+1})^{2}$.

Step 6 Test if $\underset{\sim}{x}^{i+1}$ is satisfactory solution. Yes, go to Step 9.

Step 7 Test if norm is less than previous norm. Yes, go to Step 3.

Step 8 Use norm reduction to satisfy Step 7 and then go to Step 3.

Step 9 Stop. Solution has been found.

4.4) Results of Modifications.

In the next chapter comparisons are made between the method presented and methods described earlier. In this section a comparison is made between the basic method and the method incorporating the various modifications with reference to the rate of convergence. Consider the example:-

$$f_1 = 3x_1 + x_2 + 2x_3^2 - 3$$

$$f_2 = -3x_1 + 5x_2^2 - 2x_1x_3 - 1$$

$$f_3 = 25x_1x_2 + 20x_3 + 12$$

Starting point (0,-1,0) Solution point (1.1, -0.8, 0.5)

Method presented with	No.of iterations	No.of function calls
No modification	FAIL	FAIL
Interpolation of x values (1)	31	279
Norm reduction (2)	34	238
Norm reduction halving increment (3)	13	91
Modifications (1) and (3) together	14	140

Fig: 7

It can be seen that the modifications to the method make a considerable improvement in reducing the number of function evaluations required.

4.5) A comparison of function evaluations.

It is considered that the evaluation of a derivative is equivalent to a function evaluation. In the worst possible case there will be (n+1) number of applications of the interpolation routines to overcome overshoot consisting of one interpolation for

4.5) contd.

for each variable and one for the norm. This means the maximum number of function evaluations for each iteration is (3n+1). Theoretical considerations of other methods, see Appendix (1), are shown below

NEWTON	BROWN	BROYDEN	WOLFE	NEW
$(2N^2+N)m$	$\left(\frac{N^2}{2} + \frac{3N}{2}\right)m$	$(2N^2+ N) + 3Nm$	$(2N+1)(N+1) +(3N+1)m$	$(3N+1)m$

N is the number of variables and m is the number of iterations.

4.6) Conclusions.

A new technique for finding a solution of a set of nonlinear algebraic equations has been presented which offers an attractive alternative to other available methods because of the simplicity of the method. The use of auxiliary functions reduces storage requirements. The construction of these auxiliary functions is the simplest possible according to the prerequisite of the method, that is, it should be easy to use. Interpolation and norm reduction are used to overcome the problem of overshoot and to improve convergence. This has the effect of reducing the computational time.

At present the procedure for overcoming a local solution is by means of a Newtonian step. Whilst this overcomes the problem it may not be the best solution. Perhaps further work can be developed to overcome the difficulty. Search techniques could be used provided they do not involve a large amount of computational time. In the final analysis irregardless of what technique is used to overcome this

4.6) contd.

problem care has to be taken to see that the computational time does not become prohibitive.

CHAPTER 5.

COMPARATIVE STUDY OF THE METHODS.

5.1) Introduction.

The selection of a method to solve a system of algebraic equations is one of the most difficult problems. The difficulty results from the diverse characteristics and storage requirements of each method. To the best of the author's knowledge there appears to be no adequate criteria available to select the most suitable method. Suggestions have been made with regard to computational efficiency based on the total number of iterations required to achieve the desired solution. This criterion is obviously inadequate as it takes no account of the varying complexities of each iterative step. An improved criterion was to consider the total number of function evaluations to achieve the desired solutions. This is certainly a more accurate estimate of the computational efficiency of the method but it still takes no account of the degree of computational complexity of each function. Yet another basis for comparisons is to use run-time timings. This, perhaps, is the most accurate of the three provided the computer system can offer an accurate logging of the execution time. The main objective of this chapter is to present a more suitable criteria to evaluate the relative overall qualities of the diverse methods used. A weighted value has been suggested to account for the storage requirements in addition to computational efficiency.

5.2) Standards adopted for Comparison.

A direct method of comparison between the methods is not possible since each method of solution requires a different number of function evaluations, a different degree of complexity for the construction of each incremental step and a different storage requirement.

Run-time timings will encompass all these facts but

5.2) contd.

unfortunately this is unavailable on the University Computer. The computer is an ICL 1905 machine and there are multi-programming facilities which are used by the University. This means that at any particular time up to four programs may be being processed by the machine making the time required for one particular program difficult to estimate accurately. Therefore, it is not possible to undertake any direct time comparisons.

The concept that the number of iterations required to reach a suitable solution in a method can form a basis for evaluating the efficiency of the method, should be examined more closely. In the simplest case of a one parameter system the number of iterations is probably an adequate basis for comparisons between various methods. In a many parameter system this will not be the case for the number of iterations will not provide a fair comparisons as there will be many additional calculations hidden in a step of the iteration process. A more accurate basis for comparison is the number of function evaluations performed. Even this may not provide an accurate basis of comparison because the function evaluations in each method will have varying degrees of complexity.

The theoretical number of function evaluations in each iterative step of the methods described has been calculated in the previous chapter. However, during the test runs it was noted that the theoretical maximum and actual values for the number of evaluations in the method presented did not entirely agree. This is because, generally speaking, if linear interpolation was performed on each variable, norm reduction was not performed and vice-versa, except in a few cases. Thus it was considered that a more accurate estimate to the maximum number of function

5.2) contd.

evaluations in each iteration should be 3N and not (3N+1), where N is the number of variables. Fig. 8 shows the function evaluations now required for a 2, 3 and n variable system, m being the total number of iterations performed. The derivation of these values is shown in Appendix 1.

TOTAL NUMBER OF FUNCTION EVALUATIONS

METHODS	FUNCTION EVALUATIONS IN SYSTEM		
	2 Variable	3 Variable	n Variable
Method Presented	6m	9m	3nm
Brown	5m	9m	$\left(\frac{n+3}{2}\right)$ nm
Broyden	10 + 6m	21 + 9m	(2n+1)n+ 3nm
Newton	10m	21m	(2n+1)nm
Wolfe	15 + 7m	28 + 10m	(2n+1)(n+1)+(3n+1)m

Fig: 8

Next a comparison of the storage requirements of each method should be considered. A direct comparisons is unrealistic since the method presented only requires a vector for storing the auxiliary functions whilst all the other methods require the storage of a Jacobian or a similar equivalent. A more accurate comparison is considered to be that if there are two methods whose storage requirements are m_1 and m_2 then the weighted storage ratio, S, will be calculated as

$$S = 1 + \log_{10}\left(\frac{\sqrt{m_1}}{\sqrt{m_2}}\right) \qquad \text{provided } m_1 > m_2$$

The weighted storage ratio, S, should satisfy the following basic criteria:-

5.2) contd.

(i) In a one parameter system S = 1

and (ii) as the number of variables increases S is kept within a reasonable bound.

The table in Fig. 9 gives a sample of the possible ratios for S together with the reason for discarding. m_1 is taken to be the storage requirement for a method which requires a Jacobian and m_2 is the storage requirements for the method presented.

SAMPLE OF STORAGE RATIOS.

RATIO	NUMBER OF VARIABLES IN SYSTEM			REASONS FOR DISCARDING
	1	2	50	
(m_1-m_2)	0	2	2450	i, ii
m_1/m_2	ø 1	2	50	ii
$\sqrt{m_1-m_2}$	0.0	1.14	$\simeq$ 50	i, ii
$\sqrt{m_1/m_2}$	1.0	1.14	$\simeq$ 7	ii
e^{m_1/m_2}	2.8	7.39	5.2×10^{21}	i, ii
$e^{\sqrt{m_1/m_2}}$	2.8	3.13	1.1×10^{3}	i, ii
$1 + \log_{10}(m_i-m_2)$	1.0	1.30	4.39	ii
$1 + \log_{10}(m_1-m_2)$	1.0	1.15	2.70	Possible Formulae
$1 + \log_{10}(m_1/m_2)$	1.0	1.30	2.70	
$1 + \log_{10}\sqrt{m_1/m_2}$	1.0	1.15	1.85	

Fig: 9

Most of the ratios in the table cannot be used because they do not satisfy the basic requirement. The final choice of S between the reamining ratios is from the viewpoint of a uniform rate of change as there is an increase in the number of variables

5.2) contd.

present. The ratio which satisfies this criteria best is the one $S = 1 + \log_{10}\sqrt{\frac{\bar{m}_1}{\bar{m}_2}}$. If the method presented is taken as having an S values of 1.0 then the following table, Fig.10 can be built up for S.

S-VALUES FOR OTHER METHODS

NUMBER OF VARIABLES	BROWN	BROYDEN, WOLFE NEWTON
2	1.10	1.15
3	1.15	1.24
5	1.24	1.35
50	1.72	1.85
100	1.85	2.00

Fig: 10

5.3) Test Examples.

Four specific examples are used to compare the difference between each method. These examples have been presented in other papers and all possess special characteristics which make them difficult to solve. Also, each example tests the different facilities incorporated in the method presented to see if they enhance the method.

5.3.1) Test Function 1.

$$f_1 = x_1^2 - x_2$$

$$f_2 = x_2(x_1 - 1)$$

This pair of equations form a parabolic valley

5.3.1) contd.

similar to Rosenbrook's banana function[11]. However, there are two solution points, one at (0,0) and the other at (1,1). At the solution point (0,0) the auxiliary function and the derivative of the auxiliary function for the variable x_1 are zero. It is possible that this will cause problems for the method presented.

5.3.2) Test Function 2 Brown[9]

$$f_1 = 3x_1 + x_2 + 2x_3^2 - 3$$
$$f_2 = -3x_1 + 5x_2^2 + 2x_1x_2 - 1$$
$$f_3 = 25x_1x_2 + 20x_3 + 12$$

This test function has been presented in a paper by Brown and Conte where it was used to indicate the rate of convergence of their method. There are two solution points one at (1.1, -0.8, 0.5) and the other at (0.29, 0.687, -0.849).

5.3.3) Test Function 3 Kuo[12]

$$f_1 = x_1^3x_2 + x_2^2 - 6$$
$$f_2 = 2x_1^2 + x_2^3 + 25$$

This test function has been presented in a paper by Kuo. There is one solution point at (1,-3). This example demonstrates the use of a modified increment if the derivative of the auxiliary function is zero.

5.3.4) Test Function 4 Wolfe[8]

$$f_1 = x_1^2 + x_1 - x_2^2 + 1$$
$$f_2 = x_2(1 + 2x)$$

The final example was presented in a paper by Wolfe.

5.3.4) contd.

There is only one solution point at (-0.5, 0.866). Again the derivative of the auxiliary function for x_1 is zero at the solution point as in the first test function.

5.4) Description of programs.

All the methods under consideration were programmed in Fortran IV with the exception of Brown's method. As previously explained in Chapter 3 it is not possible to program Brown in a high level language so every test function solution using Brown's method was achieved by using a desk calculator. Four different programs had to be written but each method had some common calculations to perform and so these similar sets of calculation were performed by a subroutine which was stored in a personal subroutine library ready for immediate use. This, for example, meant that there was a subroutine for the evaluation of the Jacobian in the library. In addition, the remaining details of each method were stored on magnetic tape rather than directly input into the computer for every solution run. The benefit of this is twofold; the user has only to specify his function and run the program and secondly, when the program contains a great number of instructions as in Broyden or the method presented there is no danger of the cards being lost or the order changed. All the methods posed no programming difficulties once one had appreciated the theory of the methods. Broyden's paper was particularly useful in providing a routine for norm reduction ~~meaning that the difficult programming of Broyden's method had been removed~~ and thus the programming difficulty of the method is removed. All programs were provided with the same terminating constraint namely all the f_i's $< 10^{-4}$ and the value of the norm being less than 10^{-6}. The detailed listings of Broyden and the method presented are to be found in the appendices.

$f_1 = x_1^2 - x_2$

$f_2 = x_2(x_1 - 1.0)$

METHODS

METHOD PRESENTED			BROWN			BROYDEN			NEWTON			WOLFE		
Variables		Norm	Variables		Norm	Variables		Norm	Variables		Norm	Variables		Norm
x_1	x_2	N	x_1	x_2	N	x_1	x_2	N	x_1	x_2	N	x_1	x_2	N
-1.000	1.000	4.000	-1.000	1.000	4.000	-1.000	1.000	4.000	-1.000	1.000	4.000	-1.000	1.000	4.000
-1.000	0.330	0.890	-0.600	0.200	0.128	-1.000	0.200	0.800	-0.600	0.200	0.128	-0.500	1.000	2.800
-0.660	0.167	0.154	-0.320	0.020	0.007	-0.320	-0.070	0.040	-0.330	0.030	0.008	-1.000	0.500	1.250
-0.458	0.085	0.031	-0.180	0.008	7.0 &-4	-0.160	0.010	4.0 &-4	-0.170	0.004	6.0 &-4	-0.500	0.250	0.140
-0.323	0.045	0.007	-0.150	0.020	5.0 &-4	-0.090	0.000	6.0 &-5	-0.090	0.003	5.0 &-5	-0.360	0.048	0.010
-0.230	0.024	0.002	-0.080	-0.040	0.004	-0.052	-0.002	3.0 &-5	-0.040	0.000	1.6 &-5	-0.300	0.029	0.005
-0.167	0.013	5.0 &-4	-0.040	0.000	4.0 &-5	0.031	0.000		-0.020	0.000	2.2 &-7	-0.170	0.003	6.0 &-4
-0.122	0.007	1.0 &-4	-0.034	0.000	1.0 &-6	0.013	0.000		-0.010	0.000	1.4 &-8	-2.460	0.153	35.300
-0.090	0.004	4.0 &-5	-0.024	0.002	5.0 &-6	-0.016	0.000					-0.153	-0.001	5.0 &-4
-0.070	0.002	1.0 &-5	-0.012	0.000	1.0 &-8	-0.010	0.000					0.030	0.005	5.0 &-5
-0.050	0.002					0.005	0.001					0.187	0.002	1.3 &-3
-0.040	0.001					0.007	0.000	1.0 &-8				-0.285	0.000	6.6 &-3
-0.030	0.000											-0.056	0.000	1.2 &-5
-0.020	0.000											-0.050	0.000	6.4 &-6
-0.015	0.000	2.0 &-8										-0.010	0.000	4.4 &-8

FIGURE 11

$$f_1 = 3x_1 + x_2 + 2x_3^2 - 3$$

$$f_2 = -3x_1 + 5x_2^2 + 2x_1x_3 - 1$$

$$f_3 = 25x_1x_2 + 20x_3 - 12$$

METHODS

METHOD PRESENTED				BROWN				BROYDEN				NEWTON				WOLFE			
Variables			Norm	Variables			Norm	Variables			Norm	Variables			Norm	Variables			Norm
x_1	x_2	x_3	N	x_1	x_2	x_3	N	x_1	x_2	x_3	N	x_1	x_2	x_3	N	x_1	x_2	x_3	N
0.000	0.000	0.000	154.000	0.000	0.000	0.000	154.000	0.000	0.000	0.000	154.000	0.000	0.000	0.000	154.000	0.000	0.000	0.000	154.000
1.000	0.200	-0.600	49.200	2.160	3.480	0.380	31351.000	-0.110	1.020	-0.210	51.000	-0.330	3.980	-0.600	7398.000	1.000	0.000	0.000	160.000
0.850	0.590	-1.050	31.000	1.180	-1.730	0.020	1345.000	-0.340	0.900	-0.210	26.800	-0.110	2.000	-0.860	496.000	0.000	1.000	0.000	164.000
0.070	0.560	-0.810	11.800	1.280	-1.000	1.000	4118.000	0.370	0.700	-0.750	13.000	0.080	1.070	-0.920	38.200	0.000	0.000	-1.000	66.000
0.220	0.640	-0.880	4.460	1.170	-0.810	0.580	0.160	0.340	0.730	-0.900	0.140	0.240	0.730	-0.880	2.080	0.440	0.470	-0.600	30.300
0.270	0.670	-0.850	0.220	1.100	-0.800	0.500	0.007	0.290	0.690	-0.850	0.010	0.290	0.690	-0.850	0.004	0.160	1.230	-1.230	116.000
0.300	0.690	-0.850	0.004	1.100	-0.800	0.500	1.0 &-8	0.290	0.688	-0.850	0.002	0.290	0.687	-0.849	2.0 &-9	0.310	0.570	-0.820	0.760
0.290	0.687	-0.850	0.001					0.290	0.688	-0.849	3.0 &-6					0.310	0.570	-0.820	0.700
0.290	0.688	-0.850	2.0 &-4					0.290	0.687	-0.849	1.0 &-9					0.290	0.720	-0.850	0.100
0.290	0.687	-0.849	7.0 &-5													0.290	0.680	-0.849	0.002
0.290	0.687	-0.849	1.0 &-5													0.290	0.690	-0.851	0.006
0.290	0.687	-0.849	5.0 &-7													0.290	0.687	-0.849	4.0 &-6

FIGURE 12

$$f_1 = x_1^3 x_2 + x_2^2 - 6$$
$$f_2 = 2x_1^2 + x_2^3 + 25$$

METHODS

METHOD PRESENTED			BROWN			BROYDEN			NEWTON			WOLFE		
Variables		Norm	Variables		Norm	Variables		Norm	Variables		Norm	Variables		Norm
x_1	x_2	N	x_1	x_2	N	x_1	x_2	N	x_1	x_2	N	x_1	x_2	N
-1.000	-2.000	361.000	-1.000	-2.000	361.000	-1.000	-2.000	361.000	-1.000	-2.000	361.000	-1.000	-2.000	361.000
-1.000	-2.680	74.700	0.033	-3.240	101.500	-1.000	-2.930	33.800	0.033	-3.240	101.500	-2.000	-2.000	821.000
-0.520	-3.080	28.400	-160.500	-1.600	1.0 &12	-0.530	-2.990	12.800	-162.900	-2.300	9.9 &14	-2.000	-1.000	1033.000
1.030	-3.040	0.840	8.5 &4	8.300	2.4 &31	-0.260	-2.890	6.670	-81.500	-3.500	3.5 &12	0.270	-3.620	549.800
1.020	-3.000	0.090				0.910	-2.950	1.450	-41.000	-5.200	1.3 &11	-0.240	-2.690	32.500
0.998	-2.998	0.002				0.990	-2.960	1.110	-22.200	-7.100	6.0 & 9	0.710	-3.290	104.700
0.999	-3.000	3.0 &-4				0.999	-2.990	4.0 &-5	-14.800	-7.100	5.3 & 8	-0.130	-3.380	209.300
1.000	-3.000	1.0 &-5				1.000	-3.000	4.8 &-6	-10.500	-6.100	5.2 & 8	2.080	-3.030	610.100
1.000	-3.000	1.0 &-7							-7.600	-5.200	5.1 & 6	0.310	-2.940	6.700
									-5.400	-4.400	5.0 & 5	0.920	-2.810	37.000
									-3.800	-3.800	5.0 & 4	1.050	-2.860	15.700
									-2.700	-3.400	4.8 & 3	1.290	-3.010	12.400
									-1.800	-3.100	478.000	0.720	-2.980	3.400
									-1.130	-2.990	52.700	0.930	-2.990	0.330
									-0.540	-2.920	9.350	1.060	-3.000	0.320
									0.490	-2.860	7.760	0.980	-3.000	0.020
									1.890	-3.060	316.600	1.000	-3.000	1.4 &-6
									1.370	-3.040	20.500			
									1.100	-3.010	0.820			
									1.010	-3.000	0.006			
									1.000	-3.000	6.8 &-7			

FIGURE 13

$$f_1 = x_1^2 + x_1 - x_2^2 + 1$$
$$f_2 = x_2(1 + 2x_1)$$

METHODS

METHOD PRESENTED			BROWN			BROYDEN			NEWTON			WOLFE		
Variables		Norm	Variables		Norm	Variables		Norm	Variables		Norm	Variables		Norm
x_1	x_2	N	x_1	x_2	N	x_1	x_2	N	x_1	x_2	N	x_1	x_2	N
-0.600	1.400	1.520	-0.600	1.400	1.520	-0.600	1.400	1.520	-0.600	1.400	1.520	-0.600	1.400	1.520
-0.790	0.830	0.260	-0.500	1.000	0.063	-0.477	0.870	0.002	-0.530	0.970	0.040	-0.300	1.100	0.370
-0.660	0.830	0.080	—	0.875	—	-0.500	0.866	4.0 &-7	-0.500	0.870	1.0 &-4	-0.600	1.100	0.250
-0.400	0.980	0.080				-0.500	0.866	9.0 &-9	-0.500	0.866	1.3 &-9	-0.520	0.920	0.010
-0.450	0.910	0.014										-0.503	0.870	1.0 &-4
-0.470	0.880	0.004										-0.501	0.867	4.0 &-6
-0.480	0.860	0.001										-0.500	0.866	3.0 &-9
-0.480	0.866	0.001												
-0.499	0.850	5.0 &-4												
-0.500	0.859	2.0 &-4												
-0.499	0.864	2.0 &-5												
-0.500	0.865	1.2 &-6												
-0.500	0.865	1.1 &-6												
-0.500	0.866	1.5 &-7												

FIGURE 14

5.5) Results.

The results are presented in a tabular form to facilitate ease of comparisons. Each table shows the values of the individual variables and the norms at every iterative step in the computational process. A summary of the number of function evaluations is given in Fig.15.

NUMBER OF FUNCTION EVALUATIONS

METHOD / FUNCTION	PRESENTED	BROWN	BROYDEN	NEWTON	WOLFE
TEST 1	84	45	76	70	99
TEST 2	99	54	93	126	108
TEST 3	48	FAIL	52	200	113
TEST 4	78	FAIL	28	30	43

Fig: 15

5.6) Discussion.

The following comments relate to the performance of the methods in solving the test functions. Comparisons are made between the individual methods with particular reference to their ability to obtain a solution, rate of convergence, and ease of programming.

5.6.1) Test Function 1.

This example was solved successfuly by all methods. there were no general problems for any method but all had a slow convergence to the solution notably in the variable x_1.

5.6.2) Test Function 2.

Again all methods found a solution. Brown's method located a different solution to the other methods due to the increment in x_2 following a different direction vector. It is possible for the other methods to locate the alternative root by using a different starting value for x_2. This example illustrates the differing area of convergence for Brown's method.

5.6.3) Test Function 3.

In this example a failure was experienced in Brown's method. His method was totally unstable and each variable overshot its solution value, the method being totally inadequate at attempting to control the overshoot. Newton's method also behaved erratically but did successfully find the solution. All the remaining methods were extremely successful.

5.6.4) Test Function 4.

Another failure by Brown's method was experienced in solving this example. This was because the construction of an intermediate function for one of the variables, x_1 in this case, it was necessary to divide by a zero value meaning that the function was indefinable. The method presented did not perform particularly well but did find the solution. The other three methods experienced no difficulties whatsoever in obtaining the solution.

5.7) Comparison Index.

It is now possible to form a basis as to the relative merits of each method. If there are two methods

5.7) contd.

which require n_1 and n_2 function evaluations to solve a set of equations and their storage requirements have resulted in a weighted storage ratio, S, then the comparison index, C, is defined as

$$C = \sigma \frac{n_2}{n_1} - 1 \qquad \text{where } \sigma = \frac{1}{S}$$

In this case the relative merits of the methods considered were referenced from the method presented. This is consistent with the calculation of the weighted storage ratio S, as shown in Fig.10. The interpretation for the value of C is if C is zero the method being compared with the method presented is considered to be equally good, if C is less than zero then the method in relation to the method presented is inferior, and if C is greater than zero then the method in relation to the method presented is superior. The table, Fig.16, shows the values of C for each of the methods in solving the test functions.

COMPARISON INDEX

METHOD / FUNCTION	PRESENTED	BROWN	BROYDEN	NEWTON	WOLFE
TEST 1	0.00	0.70	-0.03	0.03	-0.28
TEST 2	0.00	0.60	-0.14	-0.28	-0.24
TEST 3	0.00	FAIL	-0.20	-0.80	-0.60
TEST 4	0.00	FAIL	1.44	1.00	0.56

Fig: 16

5.8) Conclusions.

It is now possible to consider the methods which are more suitable and those which are not so adequate for solving

5.8) contd.

a set of algebraic equations. Before commenting on the various methods it is first necessary to consider the importance of a norm reduction characteristic. If a method demands that at every iteration the norm is reduced then convergence towards a solution point is guaranteed provided there is no local solution. If norm reduction is employed this may result in an increase in computational effort.

It is considered desirable to find a solution at the expense of a little extra computation rather than not find a solution. This viewpoint is reflected in the following discussion on the particular methods under consideration.

Brown's method is the most difficult to assess due to there being two failures in solving the test examples. In solving the other two examples Brown's method had the best comparison index in each case so it is not really possible to have any definite conclusions only to say that further tests should be undertaken. Whilst drawing no definite conclusions it is possible to make a few general observations. Brown does not require the storage of a Jacobian matrix. Intermediate functions are used to construct the incremental steps but their construction is of such a type that a high-level scientific programming language cannot be used for constructing them as there will be very limited character handling facilities. These intermediate functions whilst being difficult to construct do reduce the erratic tendancies found in Newton's method on which Brown's method is based. Finally, Brown's method has no norm reducing facility so there is no guarantee of convergence towards a solution.

Wolfe's method is the only method which requires additional starting values. Apart from this minor drawback the

5.8) contd.

method proved to be adequate. Although its rate of convergence is slow, as reflected in the comparison index of table, it did successfully solve all the test problems. However, a matrix based on the Jacobian matrix is needed by the method so storage requirements are large.

Newton's method, the traditional method of solution, has no norm reducing facility. Progress towards a solution can be erratic and the computational effort for every step of the iteration is ~~enormous~~ large / excessive. The Jacobian is evaluated at every step and then inverted, both of these calculations are very costly numerical processes. Newton's method, however, has the best theoretical convergence rate to the solution and because of this and the ease of programming it is still a popular method used for solution.

Broyden's method is mathematically more elegant than the other methods presented here. It was developed as an improvement to Newton's method by reducing the total computation involved in an iterative step. An approximation to the inverse Jacobian is stored and this is modified after every step so that it becomes a better approximation to the inverse Jacobian. Broyden also demanded that norm reduction was enforced so that convergence towards a solution was in a uniform manner. The price of these modifications is that the rate of convergence is slower than Newton's but as can be seen from the comparison ratio table his method is shown to be superior to Newton's method.

The method presented is an alternative approach to reducing the prohibitive calculation and storage requirements of Newton's method. The method does not require the use of a Jacobian and perhaps the main characteristic of the method is norm reduction to ensure convergence towards the desired

5.8) contd.

solution. It is not possible to derive a rate of convergence for the method since linear interpolation is used to overcome overshoot but it is only used when there has been no significant improvement made by the iterative step. However, it performs well in solving the test examples and is comparable with Broyden's method. A general observation is that as the solution approaches the order of convergence becomes slower and an improvement in this area would be desirable.

It would be unwise to say categorically that any one method is superior to another. However, it is reasonable to say that the method presented offers a suitable alternative method of solution. Furthermore, the method is easily programmable and the relative stability of obtaining a solution from a reasonable starting point does recommend it.

CHAPTER 6.

DERIVATION OF A SYSTEM OF ALGEBRAIC EQUATIONS.

6.1) Introduction.

In the preceding chapters the discussions have been concerned with the solution of a set of nonlinear algebraic equations. In practice the solution of these algebraic equations is only an integral part of a method of solution for a set of nonlinear differential equations.

In this chapter the problem of transforming a set of differential equations to a form where it is possible to use the new method and existing methods is discussed. Three possible methods of achieving the transformation are discussed, the Ritz-Averaging Method[13], Harmonic Balancing Method[14] and the Energy Balancing Method[15]. Having successfully completed the transformation phase it is possible to use the methods discussed to solve the system of algebraic equations which will have been generated.

6.2) Nonlinear differential equations.

$$M_s \ddot{x} + A_{xx}\, x + A_{xy}\, y + A_{x\dot{x}}\, \dot{x} + A_{x\dot{y}}\, \dot{y} + A_{xx^2}\, x^2 + A_{xy^2}\, y^2$$
$$+A_{xxy}\, xy + A_{xx\dot{x}}\, \dot{x}^2 + A_{xy\dot{x}}\, \dot{y}^2 + A_{xx\dot{y}}\, \dot{x}\,\dot{y} + A_{xxx}\, x\dot{x} + A_{xxy}\, x\dot{y}$$
$$+A_{xyy}\, y\dot{y} + A_{xx\dot{y}}\, x\dot{y} = m\omega^2 r \cos\omega t \qquad - (6.1.1)$$

$$M_s \ddot{y} + A_{yx}\, x + A_{yy}\, y + A_{y\dot{x}}\, \dot{x} + A_{y\dot{y}}\, \dot{y} + A_{yx^2}\, x^2 + A_{yy^2}\, y^2 + A_{yxy}\, xy$$
$$A_{y\dot{x}^2}\, \dot{x}^2 + A_{y\dot{y}^2}\, \dot{y}^2 + A_{yx\dot{y}}\, \dot{x}\,\dot{y} + A_{yx\dot{x}}\, x\dot{x} + A_{y\dot{x}y}\, \dot{x}\, y + A_{yxy}\, x\dot{y}$$
$$+ A_{yy\dot{y}}\, y\dot{y} = m\omega^2 r \sin\omega t \qquad - (6.1.2)$$

The pair of second order differential equations written above, equations (6.1.1) and (6.1.2), have been obtained from private communications with R.H.Bannister[16]. They describe the dynamics of a rigid rotor system running on journal bearings. The hydrodynamic coefficients were obtained experimentally and

6.2) contd.

new measurable terms were introduced into the system. The coefficients are read as follows: A_{xy}, for example, is the force induced in the direction x at the centre of the rotor due to unit displacement in the direction y. Thus, it can be seen that not only nonlinear but also cross-coupling terms exist in the system.

The system of equations which has been adopted to test the method presented offers two principal advantages. Firstly, the coefficients used were extremely large and they do represent an industrial type of problem. Next, since the system of equations were processed from experimental data, solutions could be compared easily.

6.3) Application of Ritz Method to nonlinear vibrations.

In nonlinear vibration problems the equation of motion in the general case has the form

$$\ddot{x} + 2ng(\dot{x}) + p^2 f(x) = F(t) \qquad -(6.2)$$

where $g(\dot{x})$, $f(x)$ and $F(t)$ are given functions of velocity, displacement and the disturbing force per unit mass, n and p^2 are constant defining the magnitude of the resisting force and the restoring force per unit mass of the system.

In equation (6.2) the various forces can be viewed as in a state of dynamic equilibrium in which the excitation force is balanced by the resisting force, the spring force and the disturbing force. The work done by this system of forces on any virtual displacement δx must vanish and hence

$$[\ddot{x} + 2ng(\dot{x}) + p^2 f(x) - F(t)]\, \delta x = 0 \qquad -(6.3)$$

Ritz method assumes an approximate solution for the steady-state vibration in the form of a series

6.3) contd.

$$x = a_1 \phi_1(t) + a_2 \phi_2(t) + a_3 \phi(t) + \ldots \qquad - (6.4)$$

in which $\phi_1(t)$, $\phi_2(t)$, ... and a_1, a_2, ... minimize the number of terms to x. Virtual displacements then have the form

$$\delta x = \delta a_n \phi_n(t) \qquad - (6.5)$$

Substituting equation (6.4) and (6.5) into equation (6.3) it will usually be found that some work on the assumed virtual displacement is produced since the series is an approximation for x not an exact solution. To obtain an accurate approximation the parameters a_1, a_2, ... are selected so as to make the average value of the virtual work per cycle vanish. This will result in equations of the following form

$$\int_0^\tau [\ddot{x} + 2ng(\dot{x}) + p^2 f(x) - F(t)] \phi_n(t) \, dt = 0$$

in which the series for x has to be substituted and then the integration performed over the period of one cycle τ. In this way as many algebraic equations as number of terms in the series are obtained and by solving we then find the values for all the parameters a_1, a_2, ...

6.3.1) Ritz method applied to rigid rotor problem.

To solve the set of equations (6.1), assume a solution of the form

$$x = a \sin\omega t + b \cos\omega t$$

$$y = c \sin\omega t + d \cos\omega t.$$

Then the nature of the differential equation is such that with this form of solution terms of even order disappear, since integrals of the type

6.3.1) contd.

$$\int_0^{2\pi} x^2 \cos\omega t \, dt = 0 \quad ; \quad \int_0^{2\pi} x^2 \sin\omega t \, dt = 0 \; ; \quad \text{etc}$$

will always be zero. Since terms of even symmetry can be neglected the equations of motion can now be represented by the following

$$M_s \ddot{x} + A_{xx}x + A_{xy}y + A_{x\dot{x}} \dot{x} + A_{x\dot{y}} \dot{y} = m\omega^2 r \cos\omega t$$

$$M_s \ddot{y} + A_{yx}x + A_{yy}y + A_{y\dot{x}} \dot{x} + A_{y\dot{y}} \dot{y} = m\omega^2 r \sin\omega t$$

$$x = a \sin\omega t + b \cos\omega t$$

$$\dot{x} = a\omega \cos\omega t - b\omega \sin\omega t$$

$$\ddot{x} = -a\omega^2 \sin\omega t - b\omega^2 \cos\omega t$$

$$y = c \sin\omega t + d \cos\omega t$$

$$\dot{y} = c\omega \cos\omega t - d\omega \sin\omega t$$

$$\ddot{y} = -c\omega^2 \sin\omega t - d\omega^2 \cos\omega t$$

Substitution yields

$$-M_s a\omega^2 \sin\omega t - M_s b\omega^2 \cos\omega t + A_{xx} a \sin\omega t + A_{xx} b \cos\omega t + A_{xy} c \sin\omega t$$
$$+ A_{xy} d \cos\omega t + A_{x\dot{x}} a\omega \cos\omega t - A_{x\dot{x}} b\omega \sin\omega t + A_{x\dot{y}} c\omega \cos\omega t$$
$$- A_{x\dot{y}} d\omega \sin\omega t - m\omega^2 r \cos\omega t = 0$$

$$-M_s c\omega^2 \sin\omega t - M_s d\omega^2 \cos\omega t + A_{yx} a \sin\omega t + A_{yx} b \cos\omega t + A_{yy} c \sin\omega t$$
$$+ A_{yy} d \cos\omega t + A_{y\dot{x}} a\omega \cos\omega t - A_{y\dot{x}} b\omega \sin\omega t + A_{y\dot{y}} c\omega \cos\omega t$$
$$- A_{y\dot{y}} d\omega \sin\omega t - m\omega^2 r \sin\omega t = 0$$

which can be rearranged as

6.3.1) contd.

$$[(A_{xx} - M_s\omega^2)a - A_{xx}\dot{}\, b\omega + A_{xy}c - A_{xy}\dot{}\, d\omega]\sin\omega t$$

$$+[A_{xx}\dot{}\,\omega a + (A_{xx} - M_s\omega^2)b + A_{xy}\dot{}\,\omega c + A_{xy}d - m\omega^2 r]\cos\omega t = 0$$

$$[A_{yx}a - A_{yx}\dot{} + (A_{yy} - M_s\omega^2)c - A_{yy}\dot{}\,\omega d - m\omega^2 r]\sin\omega t$$

$$+[A_{yx}\dot{}\,\omega a + A_{yx}b + A_{yy}\dot{}\,\omega c + (A_{yy} - M_s\omega^2)d]\cos\omega t = 0$$

$$F_1 \sin\omega t + F_2 \cos\omega t = 0$$

$$F_3 \sin\omega t + F_4 \cos\omega t = 0$$

If we now apply Ritz method we require

$$\int_0^{2\pi} (F_1 \sin\omega t + F_2 \cos\omega t) \sin\omega t \, d(\omega t) = 0$$

$$\int_0^{2\pi} (F_1 \sin\omega t + F_2 \cos\omega t) \cos\omega t \, d(\omega t) = 0$$

$$\int_0^{2\pi} (F_3 \sin\omega t + F_4 \cos\omega t) \sin\omega t \, d(\omega t) = 0$$

$$\int_0^{2\pi} (F_3 \sin\omega t + F_4 \cos\omega t) \cos\omega t \, d(\omega t) = 0$$

which is the same as $F_1, F_2, F_3, F_4 = 0$.

Hence it is required to solve

$$\begin{bmatrix} (A_{xx}-M_s\omega^2) & -A_{xx}\dot{}\,\omega & A_{xy} & -A_{xy}\dot{}\,\omega \\ A_{xx}\dot{}\,\omega & (A_{xx}-M_s\omega^2) & A_{xy}\dot{}\,\omega & A_{xy} \\ A_{yx} & -A_{yx}\dot{}\,\omega & (A_{yy}-M_s\omega^2) & -A_{yy}\dot{}\,\omega \\ A_{yx}\dot{}\,\omega & A_{yx} & A_{yy}\dot{}\,\omega & (A_{yy}-M_s\omega^2) \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} = \begin{bmatrix} 0 \\ m\omega^2 r \\ m\omega^2 r \\ 0 \end{bmatrix}$$

which will be referred to as Equations (A) throughout the remainder of the thesis. This set of equations can be solved by

6.3.1) contd.

direct matrix inversion of the (n × n) matrix. Ritz method is a general approach but if a harmonic solution is assumed then the Ritz method gives the same solution as the method of Harmonic Balance, which is developed in the next section.

6.4) The principle of harmonic balance.

Oscillations in a nonlinear system are rarely simple harmonic functions of time but are often periodic. A periodic oscillation can be expressed as a Fourier series of sine and cosine components. In many cases the fundamental frequency together with perhaps one or two higher harmonics provides a significantly accurate approximation to the true oscillatory motion. Accordingly the principle of harmonic balance asserts that the criterion for an approximate oscillatory solution is that the fundamental component is adjusted to satisfy all terms of fundamental frequency in the equation. Better approximations may be obtained, for if in addition to the fundamental component higher order components are also accounted and adjusted to satisfy all terms at their respective frequencies. This principle follows directly from the Ritz method when applied to an oscillatory system. If an assumed solution of the form

$$x = A_1 \cos\omega t + A_m \cos(n\omega t + \theta_m)$$

can be applied to the equations of motion

$$\ddot{x} + \omega_o^2 x + \phi(x, \dot{x}, t) = 0$$

then the Ritz method will result in the following equations

$$\int_0^{2\pi} [\ddot{x} + \omega_o^2 x + \phi(x, \dot{x}, t)]\, A_1 \cos\omega t \;\; d(\omega t) = 0$$

and

6.4) contd.

$$\int_0^{2\pi} [\ddot{x} + \omega_o^2 x + \phi(x, \dot{x}, t)]A_m \cos n\omega t \, d(\omega t) = 0$$

Because of the orthogonality of the circular trigonometric functions the sum of coefficients of the terms $\cos\omega t$ must each be equal to zero in the above integral equations. This is precisely the requirement for the principle harmonic balance. If we now consider the two applications of harmonic balancing to the pair of simultaneous differential equations (6.1), this will provide two different cases to consider dependent upon the initial solution assumed. The first case will use only the linear terms of the equation whilst the second will contain all the terms linear and nonlinear.

6.4.1) Linear Case.

If a solution of the form

$$x = a \sin\omega t + b \cos\omega t$$

$$y = c \sin\omega t + d \cos\omega t$$

is assumed then all the square and cross-product terms of equations (6.1) will disappear when harmonic balancing is applied. Consider, for example, the term $x\dot{x}$ then

$$x\dot{x} = a^2\omega \sin\omega t \cos\omega t + ab\omega \cos^2\omega t - ab\omega \sin^2\omega t - b^2\omega \sin\omega t \cos\omega t$$

and using the simple trigonometric identities this becomes

$$x\dot{x} = \tfrac{1}{2}(a^2-b^2)\omega\sin 2\omega t + \tfrac{1}{2}ab\omega(1+\cos 2\omega t) - \tfrac{1}{2}ab\omega(1-\cos 2\omega t)$$

It can be seen that there are no terms in $\sin\omega t$ and $\cos\omega t$, and thus no contribution will be made by this term in the balanced equations. A similar situation arises in all the square and cross-product terms so it is possible to reduce equations (6.1) to the following set of equations

6.4.1) contd.

$$M_s \ddot{x} + A_{xx}x + A_{xy}y + A_{xx}\dot{x} + A_{xy}\dot{y} = m\omega^2 r \cos\omega t$$

$$M_s \ddot{y} + A_{yx}x + A_{yy}y + A_{yx}\dot{x} + A_{yy}\dot{y} = m\omega^2 r \sin\omega t$$

Assuming the following solution

$$x = a \sin\omega t + b \cos\omega t$$

$$\dot{x} = a\omega \cos\omega t - b\omega \sin\omega t$$

$$\ddot{x} = -a\omega^2 \sin\omega t - b\omega^2 \cos\omega t$$

$$y = c \sin\omega t + d \cos\omega t$$

$$\dot{y} = c\omega \cos\omega t - d\omega \sin\omega t$$

$$\ddot{y} = -c\omega^2 \sin\omega t - d\omega^2 \cos\omega t$$

Substitution into the previous set of equations yields

$$-M_s(a\omega^2\sin\omega t+b\omega^2\cos\omega t)+A_{xx}(a\sin\omega t+b\cos\omega t)+A_{xy}(c\sin\omega t+d\cos\omega t)$$
$$+A_{xx}(a\omega\cos\omega t-b\omega\sin\omega t)+A_{xy}(c\omega\cos\omega t-d\omega\sin\omega t) = m\omega^2 r \cos\omega t$$

$$-M_s(c\omega^2\sin\omega t+d\omega^2\cos\omega t)+A_{yx}(a\sin\omega t+b\cos\omega t)+A_{yy}(c\sin\omega t+d\cos\omega t)$$
$$+A_{yx}(a\omega\cos\omega t-b\omega\sin\omega t)+A_{yy}(c\omega\cos\omega t-d\omega\sin\omega t) = m\omega^2 r \sin\omega t$$

Collecting like terms in $\sin\omega t$ and $\cos\omega t$, the following equations are obtained

$$-M_s a\omega^2 + A_{xx}a + A_{xy}c - A_{xx} b\omega - A_{xy} d\omega = 0$$

$$-M_s b\omega^2 + A_{xx}b + A_{xy}d + A_{xx}a\omega + A_{xy}c\omega = m\omega^2 r$$

$$-M_s c\omega^2 + A_{yx}a + A_{yy}c - A_{gx}b\omega - A_{yy}d\omega = m\omega^2 r$$

$$-M_s d\omega^2 + A_{yx}b + A_{yy}d + A_{yx}a\omega + A_{yy}c\omega = 0$$

Rearranging and expressing these equations in matrix form we obtain the equations from the application of the Ritz principle, equations (A), as predicted. Again we can solve for the vector [a,b,c,d] by direct matrix inversion.

6.4.2) Nonlinear case.

If a solution of the form

6.4.2) contd.

$$x = a_0 + a_1 \sin\omega t + a_2 \cos\omega t$$

$$y = b_0 + b_1 \sin\omega t + b_2 \cos\omega t$$

is now assumed all the terms of equations (6.1) will have to be considered. All the square and cross-product terms are expressed as simply as possible before substituting them into equations (6.1).

Using $x = a_0 + a_1 \sin\omega t + a_2 \cos\omega t$

$$\dot{x} = a_1\omega \cos\omega t - a_2\omega \sin\omega t$$

$$\ddot{x} = -a_1\omega^2 \sin\omega t - a_2\omega^2 \cos\omega t$$

and $y = b_0 + b_1 \sin\omega t + b_2 \cos\omega t$

$$\dot{y} = b_1\omega \cos\omega t - b_2\omega \sin\omega t$$

$$\ddot{y} = -b_1\omega^2 \sin\omega t - b_2\omega^2 \cos\omega t$$

the square and cross-product terms can be evaluated as

$$x^2 = a_0^2 + 2a_0a_1\sin\omega t + 2a_1a_2\sin\omega t\cos\omega t + 2a_0a_2\cos\omega t + a_1^2\sin^2\omega t + a_2^2\cos^2\omega t$$

$$y^2 = b_0^2 + 2b_0b_1\sin\omega t + 2b_1b_2\sin\omega t\cos\omega t + 2b_0b_2\cos\omega t + b_1^2\sin^2\omega t + b_2^2\cos^2\omega t$$

$$xy = a_0b_0 + a_1b_0\sin\omega t + a_2b_0\cos\omega t + a_0b_1\sin\omega t + a_1b_1\sin^2\omega t + a_2b_1\sin\omega t\cos\omega t + a_0b_2\cos\omega t + a_1b_2\sin\omega t\cos\omega t + a_2b_2\cos^2\omega t$$

$$\dot{x}^2 = a_1^2\omega^2 \cos^2\omega t - 2a_1a_2\omega^2\sin\omega t \cos\omega t + a_2^2\omega^2 \sin^2\omega t$$

$$\dot{y}^2 = b_1^2\omega^2 \cos^2\omega t - 2b_1b_2 \sin\omega t \cos\omega t + b_2^2\omega^2\sin^2\omega t$$

$$\dot{x}\dot{y} = a_1b_1\omega^2\cos^2\omega t - (a_1b_2 + a_2b_1)\omega^2\sin\omega t\cos\omega t + a_2b_2\omega^2\sin^2\omega t$$

$$x\dot{x} = a_0a_1\omega\cos\omega t + a_1^2\omega\sin\omega t\cos\omega t + a_1a_2\omega\cos^2\omega t - a_0a_2\omega\sin\omega t - a_1a_2\omega\sin^2\omega t - a_2^2\omega\sin\omega t\cos\omega t$$

$$x\dot{y} = a_0b_1\omega\cos\omega t + a_1b_1\omega\sin\omega t\cos\omega t + a_2b_1\omega\cos^2\omega t - a_0b_2\omega\sin\omega t - a_1b_2\omega\sin^2\omega t - a_2b_2\omega\sin\omega t\cos\omega t$$

$$\dot{x}y = a_1b_0\omega\cos\omega t + a_1b_1\omega\sin\omega t\cos\omega t + a_1b_2\omega\cos^2\omega t - a_2b_0\omega\sin\omega t - a_2b_1\omega\sin^2\omega t - a_2b_2\omega\sin\omega t\cos\omega t$$

$$y\dot{y} = b_0b_1\omega\cos\omega t + b_1^2\omega\sin\omega t\cos\omega t + b_1b_2\omega\cos^2\omega t - b_0b_2\omega\sin\omega t - b_1b_2\omega\sin^2\omega t - b_2^2\omega\sin\omega t\cos\omega t$$

Using the simple trigonometric identities

6.4.2) contd.

$$\sin\omega t \cos\omega t = \tfrac{1}{2}\sin 2\omega t$$

$$\cos^2\omega t = \tfrac{1}{2}(1+\cos 2\omega t)$$

$$\sin^2\omega t = \tfrac{1}{2}(1-\cos 2\omega t)$$

and neglecting terms in $2\omega t$ then

$$x^2 = a_o^2 + 2a_oa_1 \sin\omega t + 2a_oa_2 \cos\omega t + \tfrac{1}{2}a_1^2 + \tfrac{1}{2}a_2^2$$

$$y^2 = b_o^2 + 2b_ob_1 \sin\omega t + 2b_ob_2 \cos\omega t + \tfrac{1}{2}b_1^2 + \tfrac{1}{2}b_2^2$$

$$xy = a_ob_o + a_1b_o \sin\omega t + a_2b_o\cos\omega t + a_ob_1\sin\omega t + a_ob_2\cos\omega t + \tfrac{1}{2}a_1b_1 + \tfrac{1}{2}a_2b_2$$

$$\dot{x}^2 = \tfrac{1}{2}a_1^2\omega^2 + \tfrac{1}{2}a_2^2\omega^2$$

$$\dot{y}^2 = \tfrac{1}{2}b_1^2\omega^2 + \tfrac{1}{2}b_2^2\omega^2$$

$$\dot{x}\dot{y} = \tfrac{1}{2}a_1b_2\omega^2 + \tfrac{1}{2}a_2b_2\omega^2$$

$$x\dot{x} = a_oa_1\omega \cos\omega t - a_oa_2\omega \sin\omega t$$

$$y\dot{x} = a_1b_o\omega\cos\omega t + \tfrac{1}{2}a_1b_2\omega - a_2b_o\omega \sin\omega t - \tfrac{1}{2}a_2b_1\omega$$

$$x\dot{y} = a_ob_1\omega \cos\omega t + \tfrac{1}{2}a_2b_1\omega - a_ob_2\omega \sin\omega t - \tfrac{1}{2}a_1b_2\omega$$

$$y\dot{y} = b_ob_1\omega \cos\omega t - b_ob_1\omega \sin\omega t$$

Substituting these equations into equation (6.1.1) will yield after collecting and balancing of terms three equations one for the constant terms, one for the terms in $\sin\omega t$ and one for the terms in $\cos\omega t$.

Constant terms.

$$A_{xx}a_o+A_{xy}b_o+A_{xx^2}(a_o^2+\tfrac{1}{2}a_1^2+\tfrac{1}{2}b_1^2)+A_{xy^2}(b_o^2+\tfrac{1}{2}b_1^2+\tfrac{1}{2}b_2^2)$$
$$+A_{xxy}(a_ob_o+\tfrac{1}{2}b_1a_1+\tfrac{1}{2}a_2b_2)+A_{xx\dot{}^2}(\tfrac{1}{2}a_1^2\omega^2+\tfrac{1}{2}a_2^2\omega^2)A_{xy\dot{}^2}(\tfrac{1}{2}b_1^2\omega^2+\tfrac{1}{2}b_2^2\omega^2)$$
$$+A_{xx\dot{}\dot{}}(\tfrac{1}{2}a_1b_1\omega^2+\tfrac{1}{2}a_2b_2\omega^2)+A_{xxy\dot{}}(\tfrac{1}{2}a_2b_1\omega-\tfrac{1}{2}a_1b_2\omega)$$
$$+A_{xxy\dot{}}(\tfrac{1}{2}a_1b_2\omega - \tfrac{1}{2}a_2b_1\omega) = 0 \qquad - (B.1)$$

Sinωt terms

$$(A_{xx}-M_s\omega^2)a_1+A_{xy}b_1-A_{xx\dot{}}\omega a_2- A_{xy\dot{}}\omega b_2 + 2A_{xx^2}a_oa_1+2A_{xy^2}b_ob_1$$
$$+A_{xxy}(a_1b_o+a_ob_1)-A_{xxx\dot{}}\omega a_oa_2-A_{xxy\dot{}}\omega a_2b_o-A_{xyy\dot{}}\omega b_ob_2-A_{xxy\dot{}}\omega a_ob_2 = 0 \quad (B.2)$$

Cosωt terms.

$$(A_{xx}-M_s\omega^2)a_2+A_{xy}b_2+A_{xx\dot{}}a_1\omega+A_{xy\dot{}}b_1\omega+2A_{xx^2}a_oa_2+2A_{xy^2}b_ob_2$$

6.4.2) contd.

$$+A_{xxy}(a_2b_0+a_0b_2)+A_{xxx}\cdot\omega a_0a_1+A_{xxy}\cdot\omega a_0b_1+A_{xxy}\cdot\omega a_0b_1+A_{xyy}\cdot\omega b_0b_1$$
$$= m\omega^2 r \qquad -(B.3)$$

Similarly substitution into equation (6.1.2) will again yield a further three equations.

Constant terms

$$A_{yx}a_0+A_{yy}b_0+A_{yx^2}(a_0^2+\tfrac{1}{2}a_1^2+\tfrac{1}{2}a_2^2)+A_{yy^2}(b_0^2+\tfrac{1}{2}b_1^2+\tfrac{1}{2}b_2^2)$$
$$+A_{yxy}(a_0b_0+\tfrac{1}{2}a_1b_1+\tfrac{1}{2}a_2b_2)+A_{yx\cdot^2}(\tfrac{1}{2}a_1^2\omega^2+\tfrac{1}{2}a_2^2\omega^2)+A_{yy\cdot^2}(\tfrac{1}{2}b_1^2\omega^2+\tfrac{1}{2}b_2^2\omega^2)$$
$$+A_{yxy\cdot\cdot}(\tfrac{1}{2}a_1b_1\omega^2+\tfrac{1}{2}a_2b_2\omega^2)+A_{yxy\cdot}(\tfrac{1}{2}a_2b_1\omega-\tfrac{1}{2}a_1b_2\omega)\ +A_{yyx\cdot}(\tfrac{1}{2}a_1b_2\omega-\tfrac{1}{2}a_2b_1\omega)$$
$$= 0 \qquad -(B.4)$$

Sinωt terms.

$$(A_{yy}-M_s\omega^2)b_1+A_{yx}a_1-A_{yx\cdot}a_1-A_{yx\cdot}\omega a_2-A_{yy\cdot}\omega b_2+2A_{yx^2}a_0a_1+2A_{yy^2}b_0b_1$$
$$+A_{yxy}(a_1b_0+a_0b_1)-A_{yxx\cdot}\omega a_0a_2-A_{yyy\cdot}\omega b_0b_2-A_{yyx\cdot}\omega a_2b_0$$
$$-A_{yxy\cdot}\omega a_0b_2 = m\omega^2 r \qquad -(B.5)$$

Cosωt terms

$$(A_{yy}-M_s\omega^2)b_2+A_{yx}a_2+A_{yx\cdot}\omega a_1+A_{yy\cdot}\omega b_1+2A_{yx^2}a_0a_2+2Ayy^2b_0b_2$$
$$+A_{yxy}(a_2b_0+a_0b_2)+A_{yxx\cdot}\omega a_0a_1+A_{yyy\cdot}\omega b_0b_1+A_{yxy\cdot}\omega a_0b_1$$
$$+A_{yyx\cdot}\omega a_1b_0 = 0 \qquad -(B.6)$$

Hence we have obtained a set of nonlinear algebraic equations, (B.1) to (B.6), to solve for a_0, a_1, a_2, b_0, b_1 and b_2. This set of equations will be called equations (B), and they have been solved by using the method presented and Broyden's method.

6.5) Method of Energy Balance.

As well as developing a set of algebraic equations by Harmonic Balancing it is also possible to develop a set of equations by using Energy Balancing. The energy in the system is constant so it can be represented by a simple energy component which will be constant in the system. If we again substitute

6.5) contd.

into the equations of motion, equations 1, we will obtain a set of nonlinear equations which can be solved for the various components. Consider

$$X = X_R \cos\omega t + X_i \sin\omega t$$
$$\dot{X} = X_R\omega \sin\omega t + X_i\omega \sin\omega t$$
$$\ddot{X} = -X_R\omega^2 \cos\omega t - X_i\omega^2 \sin\omega t$$

$$Y = Y_i \cos\omega t - Y_R \sin\omega t$$
$$\dot{Y} = -Y_i\omega \sin\omega t - Y_R\omega \cos\omega t$$
$$\ddot{Y} = -Y_i\omega^2 \cos\omega t + Y_R\omega^2 \sin\omega t$$

Now, if $\omega t = 0$ then

$$X = X_R \qquad Y = Y_i$$
$$\dot{X} = X_i\omega \qquad \dot{Y} = -Y_R\omega$$
$$\ddot{X} = -X_R\omega^2 \qquad \ddot{Y} = -Y_i\omega^2$$

and if $\omega t = -\frac{\pi}{2}$

$$X = -X_i \qquad Y = Y_R$$
$$\dot{X} = X_R\omega \qquad \dot{Y} = Y_i\omega$$
$$\ddot{X} = X_i\omega^2 \qquad \ddot{Y} = -Y_R\omega^2$$

Substituion into the equation of motion (1a) will yield two equations in the horizontal direction which take the form

$$-M_s X_R\omega^2 + A_{xx}X_R + A_{xy}Y_i + A_{xx\cdot}X_i\omega - A_{xy\cdot}Y_R\omega + A_{xx^2}X_R^2 + A_{xy^2}Y_i^2$$
$$+A_{xxy}X_RY_i + A_{xx\cdot^2}X_i^2\omega^2 + A_{xy\cdot^2}Y_R^2\omega^2 + A_{xxy\cdot\cdot}X_i\omega(-Y_R\omega) + A_{xxx\cdot}X_RX_i\omega$$
$$-A_{xxy\cdot}X_RY_R\omega + A_{xyy\cdot}Y_i(-Y_R\omega) + A_{xyx\cdot}X_iY_i\omega = m\omega^2 r \qquad - (C.1)$$

and

$$M_s X_i\omega^2 - A_{xx}X_i + A_{xy}Y_R + A_{xx\cdot}X_R\omega + A_{xy\cdot}Y_i\omega + A_{xx^2}X_i^2 + A_{xy^2}Y_R^2$$
$$-A_{xxy}X_iY_R + A_{xx\cdot^2}X_R^2\omega^2 + A_{xy\cdot^2}Y_i^2\omega^2 + A_{xxy\cdot\cdot}\omega^2 - A_{xxx\cdot}X_iX_r\omega$$
$$+A_{xxy\cdot}(-X_i)(Y_i\omega) + A_{xyy\cdot}Y_RY_i\omega + A_{xxy\cdot}X_RY_R\omega = 0 \qquad - (C.2)$$

6.5) contd.

Similarly substitution into the equation of motion, equation 1b, will yield two equations in the vertical direction which take the form

$$-M_s Y_i \omega^2 + A_{yx} X_R + A_{yy} Y_i + A_{yx\cdot} X_i \omega - A_{yy\cdot} Y_R \omega + A_{yx^2} X_R^2$$
$$+ A_{yy^2} Y_i^2 + A_{yxy} X_R Y_i + A_{yx\cdot^2} X_i^2 \omega^2 + A_{yy\cdot^2} Y_R^2 \omega^2 - A_{yxy\cdot\cdot} X_i Y_R \omega^2 +$$
$$A_{yxx\cdot} X_R X_i \omega - A_{yyy\cdot} Y_i Y_R \omega - A_{yxy\cdot} X_R Y_R \omega + A_{yyx\cdot} Y_i X_i \omega = 0 \qquad - (C.3)$$

and

$$-M_s \omega^2 Y_R - A_{yx} X_i + A_{yy} Y_R + A_{yx\cdot} X_R \omega + A_{yy} X_i \omega + A_{yx^2} X_i^2 + A_{yy^2} Y_R^2$$
$$- A_{yxy} X_i Y_R + A_{yx\cdot^2} X_R^2 \omega^2 + A_{yy\cdot^2} Y_i^2 \omega^2 + A_{yxy\cdot\cdot} X_R Y_i \omega^2 - A_{yxx\cdot} Y_i X_r \omega$$
$$+ A_{yyy\cdot} Y_R Y_i \omega - A_{yxy\cdot} X_i Y_i \omega + A_{yyx\cdot} Y_R X_R \omega = 0 \qquad - (C.4)$$

Hence, a set of nonlinear algebraic equations, (C.1) to (C.4), has been obtained for X_R, Y_R, X_i, Y_i. This set of equations will be referred to as equations (C). Again they have been solved by using the method presented and Broyden's method.

6.6) Conclusions.

In this chapter three different sets of nonlinear algebraic equations have been obtained. Each set was derived from a different method of transformation.

The first set of equations (A) were obtained by two different methods of transformation, Ritz-Averaging and Harmonic Balancing. The reason is in the approximation to the solution which takes the form ($a \sin\omega t + b \cos\omega t$) and the differentiation and integration properties of the circular functions, sine and cosine. Therefore, in this case application of two different methods of transformation result in the same equation.

The second set of equations (B) are generated from an approximation to the solution which takes the form ($a+b\sin\omega t+\cos\omega t$).

6.6) contd.

This approxation contains more terms than the previous ones and therefore should be more accurate. In this case all terms, linear and nonlinear, are used by the approximation.

Finally, the last equations (C) are obtained from Energy Balancing. A solution of the form ($a\cos\omega t + b\sin\omega t$) is assumed but unlike case (A) all terms, linear and nonlinear, are included in the equations by this particular method. As only four terms not six, as in the case of equations (B), are used the solution will not be as accurate as case (B).

The following table relates the identification of the sets of algebraic equations with each of the methods used to generate them.

DERIVATION OF SETS OF ALGEBRAIC EQUATIONS.

METHOD OF DERIVATION	RITZ	HARMONIC BALANCING		ENERGY BALANCING
		Linear	Nonlinear	
EQUATIONS GENERATED	A	A	B	C

Fig: 17

CHAPTER 7.

SOLUTION OF RIGID ROTOR SYSTEM.

7.1) Introduction.

Here the three different sets of algebraic equations developed in the Harmonic chapter are to be solved by the methods described in Chapters 2 and 4. The first set of equations (A) is a linear set of equations and can be solved by a direct matrix inversion method. The second and third sets, equations (B) and (C), are nonlinear in structure and must be solved by non-analytic or iterative methods of solution.

Two different sets of values for the coefficients were obtained experimentally. This initial data is substituted into equations (A), (B) and (C). There will be three different forms of solution available for each of the two cases.

Finally, there is a discussion on the physical interpretation of these results and on the agreement between the different forms of solution.

7.2) Methods of Solution.

There are three different sets of equations to be solved, one linear and the other two nonlinear. The first set of equations (A) is the linear case obtained from the Ritz-Averaging Method. To solve these equations direct matrix inversion is used.

The second set, equations (B), is the nonlinear case obtained from the Harmonic Balancing Method. In this particular case the solution to this set of equations was not known. Therefore, two different methods of solution, Broyden's and the method presented, were used.

Finally the last case equations (C) is another nonlinear case obtained from the Energy Balancing Method. Here Broyden's method is used to confirm results obtained by the new method.

7.2) contd.

In the case of the second set of tests data, experimental results were available for equation (C). As the solution point was previously known, a spread of values around the solution point were used to examine the behaviour of the new method.

7.3) Case I.

The values of the coefficients are

$M_s = 1.5$ $\omega = 125.0$ $m\omega^2 r = 125.0$

$A_{xx} = 352000.0$	$A_{yx} = 1174000.0$
$A_{xy} = -129000.0$	$A_{yy} = 783000.0$
$A_{x\dot{x}} = 1300.0$	$A_{y\dot{x}} = 1520.0$
$A_{x\dot{y}} = 1520.0$	$A_{y\dot{y}} = 8300.0$
$A_{xx^2} = 202500000.0$	$A_{yx^2} = 170000000.0$
$A_{xy^2} = -95300000.0$	$A_{yy^2} = 157000000.0$
$A_{xxy} = 235000000.0$	$A_{yxy} = 849000000.0$
$A_{x\dot{x}^2} = 244.7$	$A_{y\dot{x}^2} = 24.5$
$A_{x\dot{y}^2} = 962.0$	$A_{y\dot{y}^2} = 2120.0$
$A_{xx\dot{y}} = 783.0$	$A_{yx\dot{y}} = 604.0$
$A_{xx\dot{x}} = 115500.0$	$A_{yx\dot{x}} = -1386000.0$
$A_{xy\dot{y}} = 1477000.0$	$A_{yy\dot{y}} = 2310000.0$
$A_{xy\dot{x}} = 1200000.0$	$A_{yy\dot{x}} = 2310000.0$
$A_{xy\dot{y}} = 923000.0$	$A_{yy\dot{y}} = 6460000.0$

7.3.1) Solution of Equation (A)

This is the linear case so only the coefficients which refer to linear terms are required. The calculated solution point is

$$x = (-0.08141 \sin\omega t + 0.34600 \cos\omega t) \times 10^{-3}$$

$$y = (-0.11380 \sin\omega t - 0.35922 \cos\omega t) \times 10^{-3}$$

7.3.1) contd.

which results in a maximum amplitude of

$$x_{max} = 0.35545 \times 10^{-3}$$

$$y_{max} = 0.37682 \times 10^{-3}$$

7.3.2) Solution of Equations (B).

These equations were solved by Broyden's method and the method presented. Both used the same starting point and reached the same solution point

Starting Point

$\underset{\sim}{f}$	$\underset{\sim}{x}$
-12.1	0.4×10^{-4}
30.4	-1.7×10^{-5}
38.0	0.5×10^{-3}
10.7	0.3×10^{-4}
- 48.7	-0.3×10^{-3}
- 39.7	-0.4×10^{-3}

Fig: 18

which leads to the solution point

$\underset{\sim}{f}$	$\underset{\sim}{x}$
0.0	0.32878×10^{-4}
0.0	-0.76053×10^{-4}
0.0	0.34361×10^{-3}
0.0	-0.58543×10^{-5}
0.0	-0.11129×10^{-3}
0.0	-0.35070×10^{-3}

Fig: 19

The solution can be expressed as

7.3.2) contd.

$$x = (0.03288 - 0.07605 \sin\omega t + 0.34361 \cos\omega t) \times 10^{-3}$$

$$y = (-0.00585 - 0.11129 \sin\omega t - 0.35070 \cos\omega t) \times 10^{-3}$$

and expressing this result in terms of double amplitudes

$$x_{max} = 0.35192 \times 10^{-3}$$

$$y_{max} = 0.36792 \times 10^{-3}$$

7.3.3) Solution of Equations (C).

These equations were obtained from Energy Balancing. The solution was unknown and as a consequence Broyden's method was used to check the results obtained by the method presented. Again the same starting point was used by both methods and they both reached the same solution point.

Starting Point

$\underset{\sim}{f}$	$\underset{\sim}{x}$
0.0	0.0
-125.0	0.0
-125.0	0.0
0.0	0.0

Fig: 20

and this leads to a solution

$\underset{\sim}{f}$	$\underset{\sim}{x}$
0.0	-0.06078×10^{-3}
0.0	0.21021×10^{-3}
0.0	0.31234×10^{-3}
0.0	-0.16722×10^{-3}

Fig: 21

Hence we can express the solution as

$$x = (-0.06078 \sin\omega t + 0.31234 \cos\omega t) \times 10^{-3}$$

$$y = (+0.21021 \sin\omega t - 0.16722 \cos\omega t) \times 10^{-3}$$

7.3.3) contd.

leading to a maximum amplitude of

$$x_{max} = 0.31293 \times 10^{-3}$$

$$y_{max} = 0.26861 \times 10^{-3}$$

7.3.4) Summary

The figure (22) shows the values of the maximum amplitude calculated for each of the three different solutions.

VALUES OF SCALED MAXIMUM AMPLITUDE.

VARIABLE / EQUATION	$x \times 10^{-3}$	$y \times 10^{-3}$
A Linear	0.35545	0.37681
B Nonlinear	0.35192	0.36792
C Nonlinear	0.31293	0.26861

Fig: 22

7.4) Case II.

The values of the coefficients in this case are

7.4) contd

$M_s = 1.52$ $\omega = 282.00$ $m\omega^2 r = 300.00$

A_{xx}	=	446000	A_{yx}	=	1481600
A_{xy}	=	58900	A_{yy}	=	2447000
$A_{\dot{x}x}$	=	1078	$A_{\dot{y}x}$	=	2070
$A_{\dot{x}y}$	=	2085	$A_{\dot{y}y}$	=	1125
A_{xx^2}	=	294000000	A_{yx^2}	=	427000000
A_{xy^2}	=	45400000	A_{yy^2}	=	1503000000
A_{xxy}	=	575000000	A_{yxy}	=	1840000000
$A_{x\dot{x}^2}$	=	1332	$A_{y\dot{x}^2}$	=	2653
$A_{x\dot{y}^2}$	=	4655	$A_{y\dot{y}^2}$	=	9190
$A_{\dot{x}\dot{y}}$	=	4985	$A_{\dot{y}\dot{y}}$	=	9920
$A_{xx\dot{x}}$	=	1468000	$A_{yx\dot{x}}$	=	1970000
$A_{xy\dot{y}}$	=	4080000	$A_{yy\dot{y}}$	=	7920000
$A_{xy\dot{x}}$	=	1570000	$A_{yy\dot{x}}$	=	3855000
$A_{xy\dot{y}}$	=	3320000	$A_{yy\dot{y}}$	=	14440000

7.4.1) Solution of Equations (A).

Again only the terms associated with linear terms are required. The calculated solution point is

$$x = (0.46274 \sin\omega t + 1.13000 \cos\omega t) \times 10^{-3}$$

$$y = (-0.35740 \sin\omega t - 0.34843 \cos\omega t) \times 10^{-3}$$

with a maximum amplitude of

$$x_{max} = 1.22108 \times 10^{-3}$$

$$y_{max} = 0.49913 \times 10^{-3}$$

7.4.2) Solution of Equations (B).

The two methods of solution were again used to solve this set of equations. Each method used the same starting point and reached the same solution point

7.4.2) contd.

Starting Point.

$\underset{\sim}{f}$	$\underset{\sim}{x}$
-125.0	-0.25×10^{-3}
48.8	0.10×10^{-3}
-122.7	0.55×10^{-3}
346.4	0.30×10^{-3}
127.7	-0.10×10^{-3}
108.4	-0.20×10^{-3}

Fig: 23

and this leads to the solution point

$\underset{\sim}{f}$	$\underset{\sim}{x}$
0.0	0.09009×10^{-3}
0.0	0.37325×10^{-3}
0.0	1.03640×10^{-3}
0.0	0.02418×10^{-3}
0.0	-0.30434×10^{-3}
0.0	-0.32806×10^{-3}

Fig: 24

The solution can be expressed as

$$x = (0.09009 + 0.37325 \sin\omega t + 1.03640 \cos\omega t) \times 10^{-3}$$

$$y = (0.02418 - 0.30434 \sin\omega t - 0.32806 \cos\omega t) \times 10^{-3}$$

and expressing the result in terms of double amplitudes

$$x_{max} = 1.10160 \times 10^{-3}$$

$$y_{max} = 0.44748 \times 10^{-3}$$

7.4.3) Solution of Equations (C)

This particular set of equations had already been solved by experiment and so the solution values were known. Therefore a spread of values about the solution point were used as starting values for the iteration. Again,Broyden

7.4.3) contd.

and the method presented were used as methods of solution.

Two tables of results are presented one for the method presented.

7.4.4) Summary.

The figure (27) shows the values of the maximum amplitude calculated for each of the three different solutions

VALUES OF SCALED MAXIMUM AMPLITUDE.

EQUATION \ VARIABLE		$x \times 10^{-3}$	$y \times 10^{-3}$
A	Linear	1.22108	0.49913
B	Nonlinear	1.10160	0.44748
C	Nonlinear	1.15493	0.43303

Fig: 27

7.5) Results.

The two figures (22), (27) showed that the amplitudes calculated for each test case shared a measure of agreement. In the cases where the solutions included nonlinear terms the amplitudes was smaller than the linear case. This is to be expected as nonlinear terms are usually components of restoring and/or damping forces.

The results quite clearly indicate that nonlinearities exist and do contribute to a significant degree in the results. However the perturbations were not undly large as evident by the solution of the linear and nonlinear cases. The experimental results have already shown that this in fact was true.

7.6) Conclusions.

A feasible method for solving a set of differential

NEW METHOD

Starting Point				Norm	Solution Point				Number of Iterations	Number of Functions Evaluations
10^{-3}	10^{-3}	10^{-3}	10^{-4}		10^{-3}	10^{-3}	10^{-3}	10^{-4}		
0.0	0.0	0.0	0.0	180000	✓	✓	✓	✓	45	450
0.0	0.5	0.0	0.0	3760000	✓	✓	✓	✓	47	564
0.0	0.0	1.0	0.0	4170000	✓	✓	✓	✓	32	384
0.0	0.0	0.0	-1.0	516000	✓	✓	✓	✓	44	528
0.0	0.0	0.0	-10.0	8340000	-1.300	-0.993	-1.290	-19.493	58	696
1.0	0.0	0.0	0.0	2480000	-0.669	-1.371	0.169	-8.786	26	312
1.0	0.5	1.0	-1.0	133000	✓	✓	✓	✓	14	168
1.0	1.0	1.0	-1.0	7660000	✓	✓	✓	✓	41	492
1.0	0.5	1.0	1.0	713000	✓	✓	✓	✓	41	492
1.0	1.0	1.0	1.0	11100000	✓	✓	✓	✓	34	408
0.0	1.0	0.0	0.0	18700000	FAILURE				—	—
-1.0	0.0	0.0	-10.0	9950000	FAILURE				—	—
0.0	-0.5	0.0	-10.0	3630000	FAILURE				—	—
0.0	-1.0	0.0	-10.0	1820000	FAILURE				—	—
-1.0	-1.0	0.0	-10.0	4230000	FAILURE				—	—

FIGURE 25

BROYDEN

Starting Point				Norm	Solution Point				Number of Iterations	Number of Functions Evaluations
10^{-3}	10^{-3}	10^{-3}	10^{-4}		10^{-3}	10^{-3}	10^{-3}	10^{-4}		
0.0	0.0	0.0	0.0	180000	✓	✓	✓	✓	13	188
0.0	0.5	0.0	0.0	3760000	✓	✓	✓	✓	17	236
0.0	0.0	1.0	0.0	4170000	✓	✓	✓	✓	12	176
0.0	0.0	0.0	-1.0	516000	✓	✓	✓	✓	15	212
1.0	0.0	0.0	0.0	2480000	✓	✓	✓	✓	26	344
0.0	0.0	0.0	-10.0	8340000	-0.669	-1.371	0.169	-8.786	24	320
1.0	0.5	1.0	-1.0	133000	✓	✓	✓	✓	8	128
1.0	1.0	1.0	-1.0	7660000	✓	✓	✓	✓	15	212
1.0	0.5	1.0	1.0	713000	✓	✓	✓	✓	10	152
1.0	1.0	1.0	1.0	11100000	✓	✓	✓	✓	16	224
0.0	1.0	0.0	0.0	18700000	✓	✓	✓	✓	17	236
-1.0	0.0	0.0	-10.0	9950000	-1.460	0.440	0.280	-7.330	29	380
0.0	-0.5	0.0	-10.0	3630000	-0.669	-1.371	0.169	-8.786	19	260
0.0	-1.0	0.0	-10.0	1820000	-0.699	-1.371	0.169	-8.786	17	236
-1.0	-1.0	0.0	-10.0	4230000	-1.460	0.440	0.280	-7.330	27	356

FIGURE 26

7.6) contd.

equations is to transform them to a set of algebraic equations and then solve these equations.

The method presented prove satisfactory in solving these algebraic equations. Broyden's method was also shown to be satisfactory and was, in fact, slightly superior. For example, in Case 2 Equations (C) Broyden's method finds solution points from every starting point whereas the method presented suffers several failures. In fact, this illustrates a general point made throughout the thesis that no one method of solution can be said to be the best method to use in all cases. In this particular case Broyden's method proves to be the more successful.

There is agreement between the theoretical and experimental results for the rigid rotor problem. Equations (B) should give the more accurate solutions as they include all terms linear and nonlinear and more terms are included in the initial approximation to the solution. Equations (C) offer reasonable solution values but still not be as accurate as (B) since the initial approximation to the solution contained fewer terms. Equations (A) are remarkably close to the solution considering that only linear terms were present in the approximate solution. Hence this would seem to indicate that the contributions of the nonlinear terms are small and also the assumptions made concerning the nonlinearities are valid.

C H A P T E R 8.

STABILITY OF NONLINEAR SYSTEMS.

8.1) Introduction.

In general the question of stability is concerned with the determination of conditions of equilibrium, be it dynamic or otherwise, and with what happens if the system is disturbed slightly near an equilibrium condition. Any disturbance near an unstable equilibrium condition leads to a larger and larger departure from the original position. Near a stable equilibrium condition any small disturbance leads to a return to the original position. It is usually not difficult to define exactly what is meant by stability in a linear system, but because new types of phenomena arise in a nonlinear system, it is not possible to use a single definition for stability which is meaningful to every case.

8.2) Structural Stability.

A concept known as structural stability is sometimes introduced in discussion of physical systems and is used in a sense somewhat different from the more general concept that might be called dynamic stability. The observation is made that the coefficients in equations describing physical systems are generally not known to a high degree of accuracy. These coefficients are the results of experimental measurements which are always subject to error. Particularly in the case of nonlinear systems, where the coefficients are functions of the operating conditions, it is difficult to determine values of the coefficients to a high degree of accuracy. Furthermore, the physical parameters are often subject to change with such ambient conditions as time, temperature and humidity which is difficult to control accurately. Changes of this sort are probably not included in equations describing the system. As a result, coefficients in the equations are invariably subject to considerable uncertainty.

8.2) contd.

The mathematical solutions for nonlinear equations can usually be found only approximately. Depending upon the nature of the nonlinearity and the method employed, the solution may or may not be of a high degree of accuracy and hence some uncertainty about the solution would arise. Because of this uncertainty as to whether or not the numerical values of coefficients are actually valid for the physical system being studied, there is reason to question whether or not the solution finally obtained actually applies to the physical system under study.

Structural stability, then, so that property of a physical system such that the qualitative nature of its operation remains unchanged if the parameters of the system are subject to small variation. The properties of the mathematical solution for equations describing the system are unchanged if small variations occur in coefficients of the equations. Because of the inherent uncertainty in relating mathematical solutions to physical systems, it is desirable to require that the system be arranged in such a way that it possesses the property of structural stability.

8.3) Dynamic Stability.

A physical system may be described by a set of simultaneous differential equations of the form

$$\frac{dx_1}{dt} = \dot{x}_1 = f_1(x_1, \ldots x_n)$$

$$\frac{dx_2}{dt} = \dot{x}_2 = f_2(x_1, \ldots x_n)$$

$$\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot$$

$$\frac{dx_n}{dt} = \dot{x}_n = f_n(x_1, \ldots x_n)$$

where t is the independent variable, $x_1, x_2, \ldots x_n$ are the n dependent variables, and the functions $f_1, f_2 \ldots f_n$ are nonlinear functions

8.3) contd.

of the dependent variables.

The simplest equilibrium, or singular, points are those points where all the derivatives $\dot{x}_1, \dot{x}_2, \ldots \dot{x}_n$ are simultaneously zero. The system is accordingly at rest, since all the dependent variables are constant and not varying with time. These functions form a set of nonlinear algebraic equations corresponding to

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \ldots\ldots a_{1n}x_n &= 0 \\
a_{21}x_1 + a_{22}x_2 + \ldots\ldots a_{2n}x_n &= 0 \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \\
a_{n1}x_1 + a_{n2}x_2 + \ldots\ldots a_{nn}x_n &= 0
\end{aligned}$$

when the system is at rest. These nonlinear equations may be satisfied by values for the variables $x_1, x_2 \ldots x_n$ which are non-zero, the equilibrium point in the linear case, and more than a single set of values may exist. Nonlinear systems, therefore, may have many equilibrium points.

In investigating the stability of a system near a chosen equilibrium point, essentially what is done is to perturb the system slightly by changing all the values of the vector $\underset{\sim}{x}$ from their equilibrium values. If, as t increases indefinitely and the values of the vector $\underset{\sim}{x}$ return to their original equilibrium values then the system is said to be asymptotically stable. On the other hand, if the values of the vector $\underset{\sim}{x}$ depart further from their equilibrium values with increasing t, then the system is said to be dynamically unstable. In a few special cases, the values of the vector $\underset{\sim}{x}$ may neither return to their original values nor depart from them. A system with this property is said to be neutrally, or temporarily stable. For a nonlinear system, it is necessary to require that the initial disturbances of the values of the vector

8.3) contd.

$\underset{\sim}{x}$ be small enough to keep them in the region controlled by the equilibrium point in question. If the initial disturbances are too large, the vector $\underset{\sim}{x}$ may then be located in a region, controlled by some other singular point.

In addition to the possible appearance of many equilibrium points, the operation of a nonlinear system may be complicated by new phenomena associated with the appearance of limit cycles. Limit cycles are steady state periodic oscillations with their properties determined entirely by parameters of the system. Because the variables in the system are undergoing continuous periodic changes in these cases, a different definition may be required.

The most rigid definition for the stability of an oscillating system is similar to the definition of asymptotic stability. According to this test for stability, the values of the vector $\underset{\sim}{x}$ of the system are perturbed slightly from their steady state motion. If the differences between the vector $\underset{\sim}{x}$ and the ensuing motion and the original undisturbed motion ultimately return to zero, the system is said to be asymptotically stable. If the differences neither vanish or increase, the system is neutrally stable.

In many applications, a change of period, accompanying only a small and nonvarying change in amplitude, does not seem to fit the usual connotations of instability. For this reason, yet another definition, that of orbital stability, is made. The solution for a system having a steady-state oscillatory motion may be represented as a closed curve in the phase-plane. If a small disturbance applied to the system results in a curve which ultimately returns to the first curve, the system is said to have orbital stability. If the small disturbances results in a curve

8.3) contd.

which leaves the first curve, the system is orbitally unstable. The definition here is related only to the amplitude and not the period of oscillation.

8.4) Rigid Rotor System.

In the preceding sections we have been discussing the stability of nonlinear systems. Two entirely different concepts of stability, structural and dynamic, have been presented. In the following sections we are going to show these two concepts in relation to the rigid rotor problem.

8.4.1) Structural Stability.

The nonlinear equations which have been presented were derived from a study of the characteristic of a rigid rotor running in a journal bearing. The coefficients, A_{xx} et cetera, of these equations have been measured experimentally. Therefore, although precautions were taken to ensure the coefficients were measured as accurately as possible they will nevertheless still contain a degree of experimental error. When the rotor is operating heat, vibration and the like can cause changes in the operating characteristics of the rotor. Again, coefficients of the equations will be subject to change and their values will be uncertain .

If a solution to the equations is asymptotically stable but does not appear to be correct practical solution then the inherent variation in the coefficient might be one of the causes. Also, if no finite solution can be found it may be that the deviations in the coefficients has rendered the system of equations unsolvable and unstable.

8.4.2) Dynamic Stability.

Dynamic stability is concerned with actual solutions obtained from a set of equations. There are several different methods for classifying the stability of a solution, Method of D-partitioning[17], Routh-Hurwitz method[18], Liapunov Methods[19] and others.

If a solution point is stable then the equations of the process under consideration are in a region of stability where the process has been defined accurately and is behaving as expected. However, if a solution point is unstable then the equations of the process under consideration are in a region of instability and the system of equations has not represented the process correctly.

The problem of instability in a nonlinear system is more significant than the corresponding linear situation for, in the nonlinear case, instability will usually mean that the equations become difficult to solve. Accordingly a degree of consideration to the question of both the stability of the coefficients and of the system must be included in a general solution of a nonlinear set of equations. However, this is not a simple matter for stability can be regarded as a separate discipline rather than a subset of any particular subject.

CHAPTER 9.

GENERAL CONCLUSIONS.

9.1) Comments on the method of approach.

The discussion in the thesis has been concerned with the available root finding techniques for solving algebraic equations, their merits and their limitations. A comprehensive study of the existing techniques was undertaken to ascertain the requirements of a root finding method. Four different methods, those of Brown, Broyden, Newton and Wolfe, were considered and a definite gap was found to exist in general root finding techniques for the solution of a set of algebraic equations. It was felt that the above mentioned techniques owe their original conceptual form to numerical rather than the engineering and/or computational viewpoint. That is to say that the methods available at present require a considerable effort to be put into their understanding and application.

The major aim of the thesis has been to develop a root finding technique which is simple to understand, is easily programmable, has a rate of convergence equal to that of existing methods and has small storage requirements.

The technique is of the simplest possible form with its origins coming from Newton's method. The desirable characteristics of existing methods have been incorporated into the structure of the method. Undesirable characteristics have been eliminated as much as possible. For example, a Jacobian matrix would cause considerable storage problems if there are a large number of variables and therefore the method presented is constructed in such a way which does not require the evaluation of the Jacobian matrix. Linear interpoltation is used at every stage of the iterative process inducing a steady convergence towards the root. Finally, norm reduction is used to ensure convergence to the required solution.

Theoretical comparisons between the various methods

9.1) contd.

of solution to solve several test examples are undertaken. Unfortunately, there is no simple technique available for a direct comparison to show which techniques offer the better methods of solution. It is possible to use the number of iterations, the number of function evaluations or run-time timings but each one has its own failings. Therefore, a secondary but nevertheless important objective has been to develop a simple ratio which can be used for comparison purposes. The results show that the method presented can offer a suitable alternative as a method of solution. Broyden's method is shown to be the best of the methods available at the present time. Newton and Wolfe's methods are shown to be techniques which will find a solution point but their order of convergence and prohibitive computational requirements reduce their effectiveness as methods of solution. No definite conclusions are made regarding Brown's method as the comparison ratio showed the method to be in the extreme, in other words extremely good or bad.

Finally, comparisons between the method presented and Broyden's method to solve an industrial problem are undertaken. Results show that Broyden's method offers a better method of solution for this particular problem.

From these comparisons the method presented can be said to offer an attractive alternative to the methods available at present. Problems of a particularly difficult nature may lead to failure of the method presented. In such a case Broyden's method, because of its mathematical foundation and derivation, is the method which should be employed to find a solution point.

In conclusion, a direct comparison between the two methods would, perhaps, find Broyden's method as superior but in defence of the method presented the prohibitive numerical search

9.1) contd.

and storage requirements weigh heavily against Broyden's method as a numerical method of solution.

9.2) Suggestions for future developments of the method presented.

In several areas the method of solution which has been presented has proved to be inadequate or in the worst possible case has broken down.

The most obvious area for development is when the method finds a local solution. At the present time, Newton's method is used to calculate an increment which will move the vector $\underset{\sim}{x}$ away from the local solution. Whilst this has proved to be an adequate solution as an interim measure the prohibitive calculation required by Newton's method is undesirable. Further work should be undertaken to overcome this difficulty. Search techniques can be used to overcome this particular problem provided they do not involve too large a computational effort. At present the search techniques used are proving to be a costly process from the computational viewpoint.

When comparisons between the various methods were undertaken it was seen that as the root was approached by the method present the rate of convergence became less than quadratic. If the functions are concave near the solution point then convergence became even slower. We know that if the functions we wish to solve are convex then convergence to the solution point is guaranteed. However, if the functions are concave this is not the case. Here it is possible to draw a comparison between Newton and Broyden's method. Broyden modified the Newtonian step by a further t, $0 < t \leq 1$, so that

$$x^{i+1} = x^{i} + t^{i} p^{i}$$

Broyden restricted his value of t such that he obtained

9.2) contd.

a Newtonian step if the functions were convex, i.e. t = 1, but he improved his convergence ratio compared with Newton if the functions were concave by allowing t to be less than 1. It is this step of Broyden's method which causes the prohibitive calculation of his norm reducing search.

If a simple test could be developed which showed whether functions are concave the incremental step of the method presented could be modified as was Broyden's to improve convergence.

9.3) Recent developments on root finding techniques.

There have been two new methods of solution proposed recently which should be mentioned.

In 1970 Broyden[20] proposed a new method of solving nonlinear simultaneous equations. He used a particular form of his method of 1965[5] coupled with a method of solution proposed by Davidenko[21]. In this recent method Broyden abandoned the norm reduction approach but otherwise the basic method of solution is the same as previously. This modified Broyden method is used to solve the subproblem but Davidenko's method requires where previously Newton's method had been used. The early results show that this type of approach is superior to the previous Broyden method.

All the methods presented so far will converge to a particular solution but since 1971 interest has been shown in a method presented by Branin[22], for finding multiple solutions to a set of algebraic equations. The early indications are that methods of this type will become increasingly popular as they can produce all the solutions from an arbitrary starting point.

9.4) Future developments on nonlinear differential equations.

A method of solution for algebraic equations is only an integral part of a more general problem of solving nonlinear simultaneous differential equations. The entire process of solution is in three separate phases. First there is a transformation from differential to algebraic equation, then the solution of these algebraic equations and finally consideration about the stability of the system. These types of method may be adequate at the present time but in the future they will be used to a lesser degree for they are long and involved processes and can induce errors into the solution. A much more ideal situation is to solve the equations directly. Unfortunately, the numerical techniques available at present offer worse solutions than transformation methods. Also there are theoretical limits to the accuracy of the numerical process.

Simulation languages are in their infancy but will be used increasingly in the future. The integration process is difficult to simulate and languages such as Slang usually experience difficulties at that particular stage in the calculation. An improvement in the expression of the integration process would enhance the simulation but the process is still cumbersome and expensive computationally. A simulation can be viewed as a digital computer being made to function like an analogue machine. Direct analogue computer methods of solution are a time consuming process. Analogue machines are difficult to program and are subject to errors during their usage.

Hybrids (computers), combinations of analogue and digital machines, can be used to offer a more feasible and controllable method of solution. The analogue section can be used to integrate whilst the digital machine can perform the simple mathematical calculation and adjust values of potentiometers in the analogue computer to

9.4) contd.

reduce the variation in values and reduce drift errors during the running time. Whilst this process is feasible with today's machinery it should be remembered that large amounts of storage and crude usage of the connecting links of the hybrid system make this method of solution uncommercial. In the future as hybrid systems are developed a process for solving nonlinear differential equations should become a much more attractive alternative. It is here that future developments should prove to be most fruitful.

APPENDIX I.

A1.1) Introduction.

The number of function evaluations for each method of solution is different. Although most methods use a form of Jacobian each method uses a different technique to update the Jacobian, resulting in a different number of function evaluations each iteration. In all cases the evaluation of a derivative is considered to be equivalent to one function evaluation. Each method is considered separately and in all cases an N variable system is investigated.

A1.2) Newton's Method.

In Newton's method there are three different calcaulations which contribute to the total number of function evaluations. The functions are evaluated once, the Jacobian has to be calculated and then an inversion is performed on the Jacobian. Hence, the total contribution to the number of function evaluations is as follows

N for the function

N^2 for the Jacobian

and N^3 for the inversion of the Jacobian leading to a total of $(2N+1)N$ function evaluations per iteration.

A1.3) Brown's method.

Examination of Brown's method shows that at the first step $(N+1)$ function evaluations are required. At the following step, the second step, N function evaluations are required until at the last or N^{th} step 2 function evaluations are required. Hence, in all

$[(N+1) + N + \ldots + 3 + 2]$ function evaluations are required. The sum of the first natural N number is $\frac{n(n+1)}{2}$ so

A1.3) contd.

$$\frac{(N+1)(N+1+1)}{2} - 1$$ function evaluations are required

or $$\frac{N^2}{2} + \frac{3N}{2}$$ function evaluations per iteration.

A1.4) Broyden's method.

In Broyden's method there are two distinct phases in the calculation of the number of function evaluations. The first iteration is a direct application of Newton's method and hence will require $(2N^2+N)$ function evaluations. Subsequent iterations will require the following number of function evaluations

N for the evaluation of the increment $\underset{\sim}{p}^i$

N for the functions $\underset{\sim}{f}^i$,

and N for the estimate of the inverse Jacobian $\underset{\sim}{H}^i$

which leads to a total number of 3N function evaluations per iteration. Hence the total number of function evaluations is found by adding

$(2N^2+N)$ for the first iteration

and $3N$ for each subsequent iteration.

A1.5) Wolfe's Method.

Wolfe's method is similar to Broyden's in that there are two distinct phases. In the first iteration it is necessary to construct the estimate to the Jacobian and invert this estimate. Hence, an additional $N(N+1) + (N+1)^2$ function evaluations are required. Every iteration requires the following function evaluations

1 for the new function

2N to calculate the vector $\underset{\sim}{p}$ and $\underset{\sim}{g}$

A1.5) contd.

and N to calculate a new value for $\underline{A}$, the Jacobian estimate.

Hence to calculate the total number of function evaluations the following quantities are added

(2N+1(N+1) for the first iteration

and (3N+1) for subsequent iterations.

A1.6) New method.

In the New method three different calculations contribute to the total number of functions evaluations. The Auxiliary functions and their derivatives have to be evaluated every iteration. Linear interpolation and Norm reduction may or may not contribute to the function evaluations so the worst possible case is considered where both have to be included. Therefore, the total contribution to the number of function evaluations is as follows

N for the Auxiliary Functions $\underline{A}$

N for the Derivatives of the Auxiliary Functions

N+1 for Norm reduction and Linear interpolation.

Hence the total number of function evaluations per iteration is found to be (3N+1).

APPENDIX II.

A2.1) Introduction.

The program developed for the new method used to solve the examples presented in the thesis is listed in this appendix. It is written in Fortran IV and is of a form which reduces the user's participation to a minimum.

The University's computer has the facility of storing programs on magnetic tape which can be called to the computer when a run is required. Therefore, a user is only required to supply his function, a suitable starting point and call the program to produce results.The listing given is a copy of the program which is stored on magnetic tape.

The program has a small master section and seventeen subroutines,each one responsible for a particular section of the calculation.

A2.2) Master Section.

In this routine all arrays used in the program are given dimensions, a requirement of Fortran IV, and the starting values are read. The arrays used are as follows

AF in which the Auxiliary Functions $\underset{\sim}{A}$ are stored

DER used by the DERivatives of the Auxiliary Functions $\frac{\partial \underset{\sim}{A}}{\partial \underset{\sim}{x}}$

DER1 used by the Jacobian in Newton's method

F the values of the functions $\underset{\sim}{f}$

INC the values of the INCrement in $\underset{\sim}{x}$

ISIGN,LTEST,L5,M1 used in various subroutines for calculation purposes

OLDINC the values of the previous or OLD INCrement

P used to store increment values for Newton's method

X in which the vector $\underset{\sim}{x}$ is stored

Y in which the previous values of the vector $\underset{\sim}{x}$ is stored

A2.2) contd.

The values read by this routine are

N2	the number of starting points
N	the number of variables $x_1 \ldots x_n$
MODE	the output requirements (0 for values at every iteration 1 for the vector $\underset{\sim}{x}$ only)
EPS	the tolerance of the norm
EPS1	the tolerance upon the function values
X	the starting values of the vector $\underset{\sim}{x}$

After reading these initial values the routine calls the subroutine JOINTMETHOD.

A2.3) Subroutine JOINTMETHOD.

This subroutine organises the different sections of the calculation. No specific calculation is performed in this routine. All arrays and values read in the master routine are transferred to this subroutine.

A2.4) Subroutine NORM.

In this subroutine norm reduction is performed if required. If a local solution is found then the routine to overcome this situation is called from this subroutine. If the iteration is unsuccessful at overcoming the problem of moving from the local solution an error message routine is called from this subroutine and the iteration terminated. The values passed to this routine are AF, DER, EPS, F, INC, OLDINC, X, Y, N and ANORM the value of the norm.

A2.5) Subroutine LOCALSOLUTION.

This routine is used to move away from a local solution point. In this routine a Newton method of solution can be called for one iteration. The input values required by this subroutine are

A2.5) contd.

AF, EPS, INC, OLDINC, N and OLDNORM the previous value of the norm.

A2.6) Newton's method and related subroutines.

If Newton's method is required for one iteration the following suite of subroutines are used, NEWTONSMETHOD, JACOBIAN, MINV and NOUTPUT.

NEWTONSMETHOD directs the iteration and calculates the increment values stored in the array P. The input to this subroutine is the values of the vector $\underset{\sim}{x}$.

JACOBIAN calculates the values of the Jacobian matrix. Again the values of the vector $\underset{\sim}{x}$ are required. The Jacobian is stored in the vector DER1 on return to the subroutine NEWTONSMETHOD.

MINV is an inversion routine to calculate the inverse of the Jacobian. The input required is DER1, the Jacobian. On exit the inverse is stored in DER1, the value of the determinant of the inverse matrix is returned and stored in D.

Finally, NOUTPUT output the values of P, F and X together with ANORM at the end of the Newton iteration.

A2.7) Subroutine CLEAR.

This routine sets the values of an array to zero. The call requires an array and the number of elements of the array. Example, CLEAR (INC,N) will set the values of the array INC to zero.

A2.8) General output routines.

There are four routines to output the values which have been calculated.

A2.8) contd.

INPUT	outputs the starting values of F,X and ANORM
OUTPUT	outputs the values of AF, DER, INC, F,X and ANORM after every iteration
ERRORMESSAGE	output the values at the local solution when the iteration can move no further. The values output are AF, DER, OLDINC, INC, F, X and ANORM.
FINALOUTPUT	outputs the values of F,X and ANORM at the solution point.

A2.9) Subroutine STARTNORM.

In this routine the value of the norm, Σf_i^2, is calculated. The input required is F and the value of the norm, in ANORM, is available on return.

A2.10) Subroutine GDAF.

The nmenonic GDAF means Generate the Derivative of the Auxiliary Function. The routine requires the values of AF, F and X and the derivative DER is available on return.

A2.11) Subroutine GIX.

The nmenonic GIX means Generate Increment in X. The routine requires the values of AF, DER and X and on return the value of the increment is available in INC and the value of the previous increment in OLDINC.

A2.12) Subroutine INTERPOLATION.

This routine is used to perform linear interpolation of x values thereby preventing overshoot. Input values required are AF, F, INC and X and on exit INC and X contain the calculated interpolated values.

A2.13) Subroutine GAF.

The nmenonic GAF means Generate Auxiliary Function.

A2.13) contd.

The routine requires F and X and on exit AF contains the value of the auxiliary function.

A2.14) Calculating Arrays.

There are four arrays ISIGN, LTEST, L5 and M1 used by various subroutines to hold values calculated during the iteration. ISIGN is used to test for a change in sign of the increment. LTEST is used to store a count value, between 1 and 3, of the number of successive increases in the value of the Derivative of the Auxiliary Function. If the value decreases LTEST is set to 0. L5 and M1 are used by the inversion routine MINV for storing counts of the row and column numbers of the elements of the array being inverted.

```
      MASTER CONTROL JOINT
      REAL INC
      DIMENSION AF(50),DER(50),DERI(2500),F(50),INC(50),ISIGN(50),LIEST(
     150),L5(50),MI(50),OLDINC(50),P(50),X(50),Y(50)
      READ (1,100) N2,N,MODE,EPS,EPS1
      NI=N*N
      DO 1 K=1,N2
      READ (1,101) (X(I),I=1,N)
      CALL JOINTMETHOD(AF,DER,DERI,EPS,EPS1,F,INC,ISIGN,LIEST,L5,MODE,
     1MI,OLDINC,P,X,Y,N,NI)
    1 CONTINUE
      STOP
  100 FORMAT(3I0,2F0.0)
  101 FORMAT(50F0.0)
      END



      SUBROUTINE JOINTMETHOD(AF,DER,DERI,EPS,EPS1,F,INC,ISIGN,LIEST,L5,
     1MODE,MI,OLDINC,P,X,Y,N,NI)
      REAL INC
      DIMENSION AF(N),DER(N),DERI(NI),F(N),INC(N),ISIGN(N),LIEST(N),L5(N
     1),MI(N),OLDINC(N),P(N),X(N),Y(N)
      M=0
      CALL FUNCTION(F,X,N)
      CALL STARTNORM(ANORM,F,N)
      CALL INPUT(ANORM,F,X,MODE,N)
    1 I=1
      DO 2 K1=1,N
      CALL GDAF(AF,DER,F,X,Y,I,N)
      CALL GIX(AF,DER,EPS,F,INC,OLDINC,X,I,LIEST,M,N)
      CALL INTERPOLATION(AF,EPS,F,INC,X,I,N)
      I=I+1
    2 CONTINUE
      CALL NORM(AF,ANORM,DER,DERI,EPS,F,INC,ISIGN,LIEST,L5,MI,OLDINC,P,
     1X,Y,M,MODE,N,NI)
      IF (M) 4,0,0
      M=M+1
      IF (MODE) 3,0,3
      CALL OUTPUT(AF,ANORM,DER,F,INC,X,M,N)
    3 IF (ANORM.GT.EPS1) GO TO 1
      CALL CLEAR(INC,N)
      CALL FINALOUTPUT(ANORM,F,X,M,N)
    4 RETURN
      END
```

```
      SUBROUTINE NORM(AF,ANORM,DER,DER1,EPS,F,INC,ISIGN,LTEST,L5,M1,OLDI
     1NC,P,X,Y,M,MODE,N,N1)
      REAL INC,MODI
      DIMENSION AF(N),DER(N),DER1(N1),F(N),INC(N),ISIGN(N),LTEST(N),
     1L5(N),M1(N),OLDINC(N),P(N),X(N),Y(N)
      ISECOND=0
      ITERNUMBER=0
      OLDNORM=ANORM
    1 INORM=0
      LITERNUMBER=0
      ITOTAL=0
      LOCAL=0
      TOTAL=0.0
      CALL CLEAR(ISIGN,N)
      DO 2 L=1,N
      IF (ABS(INC(L)).LT.EPS) GO TO 2
      IF ((OLDINC(L)/INC(L)).GE.0.0) GO TO 2
      INC(L)=INC(L)/2.0
      X(L)=X(L)-INC(L)
      INORM=INORM+1
    2 CONTINUE
      L1=1
      L4=1
      DO 3 L=1,N
      IF (ABS(INC(L)).GT.EPS) GO TO 4
      L4=L4+1
    3 CONTINUE
      L4=L4-1
      TEST=X(L4)
      GO TO 9
    4 TEST=X(L4)-INC(L4)
      TEST1=X(L4)
      IF ((TEST1-TEST).LT.0.0) L1=-1
    5 CALL FUNCTION(F,X,N)
      CALL STARTNORM(ANORM,F,N)
      IF (OLDNORM.GE.ANORM) GO TO 9
      MODI=OLDNORM/ANORM
      L2=1
      IF ((ABS(X(L4)-TEST)).LT.EPS) GO TO 7
      IF ((X(L4)-TEST).LT.0.0) L2=-1
      IF (L2+L1) 7,0,7
      RATIO=(X(L4)-TEST)/INC(L4)
      DO 6 L=1,N
      X(L)=X(L)-RATIO*INC(L)
    6 CONTINUE
      LITERNUMBER=LITERNUMBER+1
      IF (LITERNUMBER-20) 5,11,11
```

```
 7 DO 8 L=1,N
   IF (ABS(INC(L)).LT.EPS) GO TO 8
   X(L)=X(L)-(1.0-MODI)*INC(L)
 8 CONTINUE
   LITERNUMBER=LITERNUMBER+1
   IF (LITERNUMBER-20) 5,11,11
 9 IF (OLDNORM.GT.ANORM) GO TO 12
10 IF ((INORM.NE.0).AND.(OLDNORM.GE.ANORM)) RETURN
   IF ((X(L4)-TEST).LT.EPS) GO TO 11
   RATIO=(X(L4)-TEST)/INC(L4)
   DO 11 L=1,N
   X(L)=X(L)-RATIO*INC(L)
11 CONTINUE
   CALL LOCALSOLUTION(AF,DER,DER1,EPS,F,INC,L5,M1,OLDINC,OLDNORM,P,TO
  1TAL,X,Y,ISIGN,ITOTAL,LOCAL,MODE,N,N1)
   CALL FUNCTION(F,X,N)
   INORM=0
   ITERNUMBER=ITERNUMBER+1
   IF (ITERNUMBER-20) 5,14,5
12 IF (((ABS(X(L4)-TEST)).GT.EPS).OR.((ABS(OLDNORM-ANORM)).GT.EPS))
  1RETURN
   I=1
   DO 13 K1=1,N
   CALL GDAF(AF,DER,F,X,Y,I,N)
   CALL GIX(AF,DER,EPS,F,INC,OLDINC,X,I,LTEST,M,N)
   CALL INTERPOLATION(AF,EPS,F,INC,X,I,N)
   CALL GAF(AF,F,X,I,N)
   I=I+1
13 CONTINUE
   ITERNUMBER=ITERNUMBER+1
   IF (ITERNUMBER-20) 1,0,0
   TOTAL=0.0
   DO 14 L=1,N
   TOTAL=TOTAL+ABS(INC(L))
14 CONTINUE
   IF (ISECOND) 16,0,16
   IF (TOTAL.EQ.0.0) TOTAL=N
   VALUE=TOTAL/N
   DO 15 L=1,N
   IF (INC(L).EQ.0.0) INC(L)=1.0
   INC(L)=SIGN(VALUE,INC(L))
15 CONTINUE
   ISECOND=ISECOND+1
   GO TO 1
16 CALL ERRORMESSAGE(AF,ANORM,DER,F,INC,OLDINC,X,N)
   M=-1
   RETURN
   END
```

```
      SUBROUTINE LOCALSOLUTION(AF,DER,DER1,EPS,F,INC,L5,M1,OLDINC,OLDNOR
     1M,P,TOTAL,X,Y,ISIGN,ITOTAL,EXIT,LOCAL,M,MODE,N,N1)
      REAL INC
      DIMENSION AF(N),DER(N),DER1(N1),F(N),INC(N),ISIGN(N),L5(N),M1(N),
     1OLDINC(N),P(N),X(N),Y(N)
      ICOUNT=1
    1 L1=1
      DO 2 L=1,N
      IF (ABS(AF(L)).LT.EPS) GO TO 6
      L1=L1+1
    2 CONTINUE
      CALL NEWTONSMETHOD(OLDNORM,DER1,EPS,EXIT,F,P,X,Y,L5,M,MODE,M1,N,N1
     1)
      IF (LOCAL-1) 4,4,0
      DO 3 L=1,N
      P(L)=-P(L)
    3 CONTINUE
    4 CONTINUE
      DO 5 L=1,N
      INC(L)=P(L)
    5 CONTINUE
      LOCAL=LOCAL+1
      ICOUNT=5
      GO TO 19
    6 IF (ABS(INC(L1)).LT.EPS) GO TO 16
      ISIGN(L1)=ISIGN(L1)+1
      IF (ISIGN(L1)-1) 9,9,0
    7 IF (L1-N) 0,11,11
      L2=L1+1
      L1=L2
      DO 8 L=L2,N
      IF (ABS(AF(L)).LT.EPS) GO TO 10
      L1=L1+1
    8 CONTINUE
      GO TO 11
    9 INC(L1)=-INC(L1)
      ITOTAL=ITOTAL+1
      TOTAL=TOTAL+ABS(INC(L1))
      GO TO 20
   10 ISIGN(L1)=ISIGN(L1)+1
      IF (ISIGN(L1)-1) 9,9,7
```

```
11 CALL NEWTONSMETHOD(OLDNORM,DER1,EPS,EXIT,F,P,X,Y,L5,M,MODE,M1,N,NI
  1)
   IF (LOCAL-1) 13,13,0
   DO 12 L=1,N
   P(L)=-P(L)
12 CONTINUE
13 CONTINUE
   DO 14 L=1,N
   INC(L)=P(L)
14 CONTINUE
   LOCAL=LOCAL+1
   ICOUNT=5
   TOTAL=0.0
   DO 15 L=1,N
   IF (ABS(AF(L)).LT.EPS) TOTAL=TOTAL+ABS(INC(L))
15 CONTINUE
   GO TO 20
16 IF ((ABS(INC(L1)).LT.EPS).AND.(ABS(OLDINC(L1)).LT.EPS)) GO TO 17
   INC(L1)=OLDINC(L1)
   GO TO 20
17 STOTAL=0.0
   DO 18 L=1,N
   STOTAL=STOTAL+ABS(INC(L))
18 CONTINUE
19 CONTINUE
   IF (STOTAL.EQ.0.0) STOTAL=N
   SVALUE=STOTAL/N
   IF (INC(L1).EQ.0.0) INC(L1)=1.0
   INC(L1)=SIGN(SVALUE,INC(L1))
20 DO 21 L=1,N
   X(L)=X(L)+INC(L)
21 CONTINUE
   IF (EXIT.EQ.1.0) RETURN
   CALL FUNCTION(F,X,N)
   CALL STARTNORM(TEST,F,N)
   IF (ABS(OLDNORM-TEST).GT.EPS) RETURN
   IF ((ICOUNT-5).GE.0) RETURN
   ICOUNT=ICOUNT+1
   DO 22 L=1,N
   X(L)=X(L)-INC(L)
22 CONTINUE
   GO TO 1
   END
```

```
      SUBROUTINE NEWTONSMETHOD(OLDNORM,DER1,EPS,EXIT,F,P,X,Y,L5,M,MODE,
     1M1,N,N1)
      DIMENSION DER1(N1),F(N),P(N),X(N),Y(N),L5(N),M1(N)
    1 CALL JACOBIAN(DER1,F,X,Y,N,N1)
      CALL MINV(DER1,N,D,L5,M1,N1)
      IF (D) 0,6,0
      DO 3 I=1,N
      K=I
      P(I)=0.0
      DO 2 J=1,N
      P(I)=-DER1(K)*F(J)+P(I)
      K=K+N
    2 CONTINUE
    3 CONTINUE
      IF (OLDNORM.GT.0.003) RETURN
      DO 4 I=1,N
      X(I)=X(I)+P(I)
    4 CONTINUE
      CALL FUNCTION(F,X,N)
      CALL STARTNORM(OLDNORM,F,N)
      IF (OLDNORM-EPS) 5,5,0
      M=M+1
      IF (MODE) 1,0,1
      CALL NOUTPUT(OLDNORM,F,P,X,M,N)
      GO TO 1
    5 CALL FINALOUTPUT(OLDNORM,F,X,M,N)
      EXIT=1.0
      RETURN
    6 WRITE (2,600)
      RETURN
  600 FORMAT(/////50X,25H THE JACOBIAN IS SINGULAR)
      END
```

```
      SUBROUTINE JACOBIAN(DER1,F,X,Y,N,N1)
      DIMENSION DER1(N1),F(N),X(N),Y(N)
      K=0
      K1=0
      DO 1 L=1,N
      Y(L)=X(L)
    1 CONTINUE
      DO 9 L=1,N
      K=K1+1
      K1=L*N
      IF (X(L).NE.0.0) GO TO 2
      Y(L)=0.001
      GO TO 3
    2 Y(L)=X(L)*1.001
    3 CALL FUNCTION(F,Y,N)
      DO 4 K2=K,K1
      K3=K2+N-L*N
      DER1(K2)=F(K3)
    4 CONTINUE
      CALL FUNCTION(F,X,N)
      IF (X(L).NE.0.0) GO TO 5
      Y(L)=0.0
      GO TO 7
    5 Y(L)=Y(L)/1.001
      DO 6 K2=K,K1
      K3=K2+N-L*N
      DER1(K2)=(DER1(K2)-F(K3))*1000.0/X(L)
    6 CONTINUE
      GO TO 9
    7 DO 8 K2=K,K1
      K3=K2+N-L*N
      DER1(K2)=(DER1(K2)-F(K3))*1000.0
    8 CONTINUE
    9 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MINV(A,N,D,L5,MI,NI)
      DIMENSION A(NI),L5(N),MI(N)
      D=1.0
      NK=-N
      DO 10 K=1,N
      NK=NK+N
      L5(K)=K
      MI(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 1 J=K,N
      IZ=N*(J-1)
      DO 1 I=K,N
      IJ=IZ+I
      IF (ABS(BIGA)-ABS(A(IJ))) 0,1,1
      BIGA=A(IJ)
      L5(K)=I
      MI(K)=J
    1 CONTINUE
      J=L5(K)
      IF (J-K) 3,3,0
      KI=K-N
      DO 2 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
    2 A(JI)=HOLD
    3 I=MI(K)
      IF (I-K) 5,5,0
      JP=N*(I-1)
      DO 4 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
    4 A(JI)=HOLD
    5 IF (BIGA) 6,0,6
      D=0.0
      RETURN
    6 DO 7 I=1,N
      IF (I-K) 0,7,0
      IK=NK+I
      A(IK)=A(IK)/(-BIGA)
    7 CONTINUE
```

```
      DO 8 I=1,N
      IK=NK+I
      IJ=I-N
      DO 8 J=1,N
      IJ=IJ+N
      IF (I-K) 0,8,0
      IF (J-K) 0,8,0
      KJ=IJ-I+K
      A(IJ)=A(IK)*A(KJ)+A(IJ)
    8 CONTINUE
      KJ=K-N
      DO 9 J=1,N
      KJ=KJ+N
      IF (J-K) 0,9,0
      A(KJ)=A(KJ)/BIGA
    9 CONTINUE
      D=D*BIGA
      A(KK)=1.0/BIGA
   10 CONTINUE
      K=N
   11 K=(K-1)
      IF (K) 15,15,0
      I=L5(K)
      IF (I-K) 13,13,0
      JQ=N*(K-1)
      JR=N*(I-1)
      DO 12 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
   12 A(JI)=HOLD
   13 J=M1(K)
      IF (J-K) 11,11,0
      KI=K-N
      DO 14 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
   14 A(JI)=HOLD
      GO TO 11
   15 RETURN
      END
```

```
      SUBROUTINE CLEAR(INC,N)
      REAL INC
      DIMENSION INC(N)
      DO 1 L=1,N
      INC(L)=0.0
    1 CONTINUE
      RETURN
      END



      SUBROUTINE INPUT(ANORM,F,X,MODE,N)
      DIMENSION F(N),X(N)
      WRITE(2,200)
      WRITE(2,201)
      DO 1 I=1,N
      WRITE(2,202) F(I),X(I)
    1 CONTINUE
      WRITE(2,203)
      WRITE(2,204) ANORM
      IF (MODE) 2,0,2
      WRITE(2,205)
    2 RETURN
  200 FORMAT(1H1,51X,25H NEW SET OF STARTING DATA)
  201 FORMAT(/50X,16H FUNCTION VALUES,5X,9H X VALUES)
  202 FORMAT(51X,F10.5,8X,F10.5)
  203 FORMAT(/60X,11H NORM VALUE)
  204 FORMAT(61X,F10.5)
  205 FORMAT(/15X,20H AUXILIARY FUNCTIONS,5X,12H DERIVATIVES,7X,10H INCR
     1EMENT,5X,16H FUNCTION VALUES,9X,9H X VALUES)
      END



      SUBROUTINE OUTPUT(AF,ANORM,DER,F,INC,X,M,N)
      REAL INC
      DIMENSION AF(N),DER(N),F(N),INC(N),X(N)
      WRITE (2,300) M
      DO 1 I=1,N
      WRITE (2,301) AF(I),DER(I),INC(I),F(I),X(I)
    1 CONTINUE
      WRITE (2,302) ANORM
      RETURN
  300 FORMAT(///57X,11H ITERATION ,I3)
  301 FORMAT(15X,F16.5,3X,F16.5,7X,F11.5,6X,F11.5,11X,F11.5)
  302 FORMAT(/57X,12H NORM VALUE=,F10.5)
      END
```

```
      SUBROUTINE ERRORMESSAGE(AF,ANORM,DER,F,INC,OLDINC,X,N)
      REAL INC
      DIMENSION AF(N),DER(N),F(N),INC(N),OLDINC(N),X(N)
      WRITE (2,400)
      WRITE (2,401)
      WRITE (2,402)
      WRITE (2,403)
      WRITE (2,404)
      WRITE (2,401)
      WRITE (2,400)
      WRITE (2,405)
      DO 1 L=1,N
      WRITE (2,406) AF(L),DER(L),OLDINC(L),INC(L),F(L),X(L)
    1 CONTINUE
      WRITE (2,407) ANORM
      RETURN
  400 FORMAT(5(1H ))
  401 FORMAT(40X,42(1H*))
  402 FORMAT(40X,42H*       THE METHOD HAS FOUND A LOCAL       *)
  403 FORMAT(40X,42H*     SOLUTION AND IS UNABLE TO MOVE       *)
  404 FORMAT(40X,42H*       FROM THE POSITION GIVEN BELOW      *)
  405 FORMAT(/7X,20H AUXILIARY FUNCTIONS,5X,12H DERIVATIVES,7X,13H OLDIN
     1CREMENT,7X,10H INCREMENT,5X,16H FUNCTION VALUES,9X,9H X VALUES)
  406 FORMAT(7X,F16.5,3X,F16.5,9X,F11.5,7X,F11.5,6X,F11.5,11X,F11.5)
  407 FORMAT(/51X,12H NORM VALUE=,F10.5)
      END


      SUBROUTINE FINALOUTPUT(ANORM,F,X,M,N)
      DIMENSION F(N),X(N)
      KITER=3*M*N
      WRITE(2,500)
      WRITE(2,501)
      DO 1 I=1,N
      WRITE(2,502) F(I),X(I)
    1 CONTINUE
      WRITE(2,503)
      WRITE(2,504) ANORM
      WRITE(2,505) M
      WRITE(2,506) KITER
      RETURN
  500 FORMAT(/////52X,26H FINAL VALUES OF VARIABLES)
  501 FORMAT(/50X,16H FUNCTION VALUES,5X,9H X VALUES)
  502 FORMAT(51X,F10.5,8X,F10.5)
  503 FORMAT(/60X,11H NORM VALUE)
  504 FORMAT(61X,F10.5)
  505 FORMAT(/53X,21HNUMBER OF ITERATIONS=,I2)
  506 FORMAT(/48X,31HNUMBER OF FUNCTION EVALUATIONS=,I3)
      END
```

```
      SUBROUTINE NOUTPUT(ANORM,F,P,X,M,N)
      DIMENSION F(N),P(N),X(N)
      DO 1 I=1,N
      WRITE (2,700) P(I),F(I),X(I)
    1 CONTINUE
      WRITE (2,701) ANORM
      RETURN
  700 FORMAT(57X,F11.5,6X,F11.5,11X,F11.5)
  701 FORMAT(/57X,12H NORM VALUE=,F10.5)
      END



      SUBROUTINE STARTNORM(ANORM,F,N)
      DIMENSION F(N)
      ANORM=0.0
      DO 1 L=1,N
      ANORM=F(L)*F(L)+ANORM
    1 CONTINUE
      RETURN
      END



      SUBROUTINE GDAF(AF,DER,F,X,Y,I,N)
      DIMENSION AF(N),DER(N),F(N),X(N),Y(N)
      DO 1 L=1,N
      Y(L)=X(L)
    1 CONTINUE
      IF (X(I).NE.0.0) GO TO 2
      Y(I)=0.001
      GO TO 3
    2 Y(I)=X(I)*1.001
    3 CALL GAF(AF,F,Y,I,N)
      DER(I)=AF(I)
      CALL GAF(AF,F,X,I,N)
      IF (X(I).NE.0.0) GO TO 4
      Y(I)=0.0
      GO TO 5
    4 Y(I)=Y(I)/1.001
      DER(I)=(DER(I)-AF(I))*1000.0/X(I)
      RETURN
    5 DER(I)=(DER(I)-AF(I))*1000.0
      RETURN
      END
```

```
      SUBROUTINE GIX(AF,DER,EPS,F,INC,OLDINC,X,I,LTEST,M,N)
      REAL INC
      DIMENSION AF(N),DER(N),F(N),INC(N),OLDINC(N),X(N),LTEST(N)
      TOTAL=AF(I)
      OLDINC(I)=INC(I)
      IF (DER(I).EQ.0.0) GO TO 4
      INC(I)=-TOTAL/DER(I)
      IF (INC(I).EQ.0.0) GO TO 1
      IF (M) 0,1,0
      IF (ABS(OLDINC(I)).LT.ABS(INC(I))) GO TO 3
    1 LTEST(I)=-3
    2 X(I)=X(I)+INC(I)
      RETURN
    3 LTEST(I)=LTEST(I)+1
      IF (LTEST(I)) 2,0,2
    4 L2=1
      IF (DER(I).LT.0.0) L2=-1
      INC(I)=-TOTAL/((ABS(TOTAL)+ABS(DER(I)))*L2)
      GO TO 2
      END
```

```
      SUBROUTINE INTERPOLATION(AF,EPS,F,INC,X,I,N)
      REAL INC,NEW,NINC
      DIMENSION AF(N),F(N),INC(N),X(N)
      OLD=AF(I)
      CALL GAF(AF,F,X,I,N)
      NEW=AF(I)
      IF ((ABS(NEW).LT.EPS).OR.(ABS(OLD).LT.EPS)) RETURN
      IF ((OLD/NEW).GT.0.1) RETURN
      NINC=(ABS(NEW)*INC(I))/(ABS(NEW)+ABS(OLD))
      X(I)=X(I)-NINC
      INC(I)=INC(I)-NINC
      RETURN
      END
```

```
      SUBROUTINE GAF(AF,F,X,I,N)
      DIMENSION AF(N),F(N),X(N)
      CALL FUNCTION(F,X,N)
      SUM=0.0
      DO 1 L=1,I
      AF(I)=F(L)+SUM
      SUM=AF(I)
    1 CONTINUE
      RETURN
      END
```

APPENDIX III.

A3.1) Introduction.

The program presented in this section is Broyden's method of solution, 1965[5]. It is written in Fortran IV and is structured the same as the previous program of Appendix II. Again there is a small master routine but this time there are only nine subroutines.

A3.2) Master routine.

The master routine dimensions the arrays used, reads in the starting values and organises the subroutines used in the calculation. The arrays used are

FI	to store the function values $\underset{\sim}{f}$
P	to store the incremental values for the vector $\underset{\sim}{x}$
X	to store the values of the vector $\underset{\sim}{x}$
Y	to store the differences in the function values for successive iterations i.e. $\underset{\sim}{y} = \underset{\sim}{f}(\underset{\sim}{x}_{i+1}) - \underset{\sim}{f}(\underset{\sim}{x}_i)$
H	to store the Jacobian $\underset{\sim}{J}$

with the starting values

H	the number of variables
EPS	the tolerance upon the norm
X	the starting values of the vector $\underset{\sim}{x}$

The arrays COL, MK, ROW, SUM1 and SUM2 are for storing intermediate values in the calculation.

A3.3) Subroutine NEW.

This subroutine is used to calculate the increment, P. The routine requires the values of FI, H and X. On output a new value for X is available together with a value for P. This subroutine calls the routine QUADMIN, used to reduce the value of the norm.

A3.4) Subroutine INITIALH.

At the beginning of the iteration a Newton method iteration, which will require an estimate of the inverse Jacobian matrix, is performed. This subroutine calculates the initial estimate to the inverse Jacobian. The input required is the value of FI and X. During the evaluation of the estimate, an inversion routine, INVERT, is called. On exit the matrix, H, contains the values of the inverse Jacobian.

A3.5) Subroutine HNEW.

In this subroutine the values of the estimate to the inverse Jacobian are updated using the latest information. The required input values are H, P and T and on return the matrix H contains the latest estimate to the inverse Jacobian.

A3.6) Subroutine NORM.

This routine calculates the value of the norm $\sum_{i=1}^{n} f_i^2$. The input required is FI and the value of the norm, in ANORM, is available on return.

A3.7) Subroutine YNEW.

This subroutine calculates the difference between the function values, that is to say $y = f(x_{i+1}) - f(x_i)$. The input values required are FI, P, T and X and on return Y will contain the difference between the function values.

A3.8) Subroutine OUTPUT.

In this routine the values P, Y, FI, ANORM and X are output. This routine is called at the end of every iteration.

A3.9) Other routines.

Finally, there are three routines remaining INVERT, QUADMIN and SET. The routines QUADMIN and SET were developed by Broyden in his paper[5] and are adequately documented. The remaining subroutine, INVERT, is a general routine for inverting a matrix.

```
      MASTER BROYDEN ORIGINAL
      REAL MK
      INTEGER COL,ROW
      DIMENSION COL(2),FI(2),MK(2),P(2),ROW(2),SUM1(2),SUM2(2),X(2),Y(2)
      COMMON H(2,2)
      N2=1
      READ(1,100) N,EPS,X
      CALL INITIALH(COL,FI,MK,ROW,X,Y,N)
    1 CALL NEW(ANORM,FI,P,X,T,N)
      IF (ANORM-EPS) 2,2,0
      CALL YNEW(FI,P,T,X,Y,N)
      CALL HNEW(P,SUM1,SUM2,T,Y,N)
      CALL FUNCTION(FI,X,N)
      CALL OUTPUT(ANORM,FI,P,X,Y,N,N2)
      N2=N2+1
      GO TO 1
    2 CALL OUTPUT(ANORM,FI,P,X,Y,N,N2)
  100 FORMAT(I0,3F0.0)
      STOP
      END



      SUBROUTINE NEW(ANORM,FI,P,X,T,N)
      DIMENSION FI(N),P(N),X(N)
      COMMON H(2,2)
      CALL NORM(ANORM,FI,N)
      DO 1 I=1,N
      P(I)=0.0
      DO 2 J=1,N
      P(I)=-H(I,J)*FI(J)+P(I)
    2 CONTINUE
      X(I)=X(I)+P(I)
    1 CONTINUE
      CALL QUADMIN(ANORM,FI,P,T,X,N)
      RETURN
      END
```

```
      SUBROUTINE INITIALH(COL,FI,MK,ROW,X,Y,N)
      REAL MK
      INTEGER COL,ROW
      DIMENSION COL(N),FI(N),MK(N),ROW(N),X(N),Y(N)
      COMMON H(2,2)
      DO 1 I=1,N
      Y(I)=X(I)
    1 CONTINUE
      DO 8 J=1,N
      IF (X(J).NE.0.0) GO TO 2
      Y(J)=0.001
      GO TO 3
    2 Y(J)=X(J)*1.001
    3 CALL FUNCTION(FI,Y,N)
      DO 4 I=1,N
      H(I,J)=FI(I)
    4 CONTINUE
      CALL FUNCTION(FI,X,N)
      IF (X(J).NE.0.0) GO TO 6
      Y(J)=0.0
      DO 5 I=1,N
      H(I,J)=(H(I,J)-FI(I))*1000.0
    5 CONTINUE
      GO TO 8
    6 Y(J)=Y(J)/1.001
      DO 7 I=1,N
      H(I,J)=(H(I,J)-FI(I))*1000.0/X(J)
    7 CONTINUE
    8 CONTINUE
      CALL INVERT(COL,MK,ROW,N)
      RETURN
      END
```

```
      SUBROUTINE HNEW(P,SUM1,SUM2,T,Y,N)
      DIMENSION P(N),SUM1(N),SUM2(N),Y(N)
      COMMON H(2,2)
      SUM3=0.0
      DO 1 L=1,N
      DO 1 M=1,N
      SUM3=P(L)*H(L,M)*Y(M)+SUM3
    1 CONTINUE
      DO 3 I=1,N
      DO 3 J=1,N
      SUM1(I)=0.0
      SUM2(J)=0.0
      DO 2 K=1,N
      SUM1(I)=H(I,K)*Y(K)+SUM1(I)
      SUM2(J)=P(K)*H(K,J)+SUM2(J)
    2 CONTINUE
      SUM1(I)=SUM1(I)-P(I)*T
      SUM2(J)=SUM2(J)/SUM3
    3 CONTINUE
      DO 4 I=1,N
      DO 4 J=1,N
      H(I,J)=H(I,J)-SUM1(I)*SUM2(J)
    4 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE NORM(ANORM,FI,N)
      DIMENSION FI(N)
      ANORM=0.0
      DO 1 I=1,N
      ANORM=FI(I)*FI(I)+ANORM
    1 CONTINUE
      RETURN
      END



      SUBROUTINE YNEW(FI,P,T,X,Y,N)
      DIMENSION FI(N),P(N),X(N),Y(N)
      DO 1 I=1,N
      X(I)=X(I)-P(I)*T
      Y(I)=FI(I)
    1 CONTINUE
      CALL FUNCTION(FI,X,N)
      DO 2 I=1,N
      Y(I)=Y(I)-FI(I)
      X(I)=X(I)+P(I)*T
    2 CONTINUE
      RETURN
      END



      SUBROUTINE OUTPUT(ANORM,FI,P,X,Y,N,N2)
      DIMENSION FI(N),P(N),X(N),Y(N)
      IF (N2-1) 0,0,1
      WRITE (2,200)
    1 WRITE (2,201) N2
      DO 2 I=1,N
      WRITE (2,202) P(I),Y(I),FI(I),ANORM,X(I)
    2 CONTINUE
      RETURN
  200 FORMAT(1H1,/////37X,4HP(I),6X,4HY(I),5X,7HFI(I+1),3X,5H NORM,6X,6H
     1X(I+1))
  201 FORMAT(/55X,11H ITERATION=,I2)
  202 FORMAT(33X,5F10.5)
      END
```

```
      SUBROUTINE INVERT(COL,MK,ROW,N)
      REAL MK
      INTEGER COL,ROW
      DIMENSION COL(N),MK(N),ROW(N)
      COMMON H(2,2)
      X3=1.0
      DO 100 I=1,N
  100 ROW(I),COL(I)=I
      DO 200 I8=1,N
      I2=ROW(I8)
      I3=COL(I8)
      I6,I7=I8
      X1=H(I2,I3)
      DO 300 I1=I8,N
      I4=ROW(I1)
      DO 400 I9=I8,N
      I5=COL(I9)
      X2=H(I4,I5)
      IF (ABS(X2).LE.ABS(X1)) GO TO 400
      X1=X2
      I2=I4
      I3=I5
      I6=I1
      I7=I9
  400 CONTINUE
  300 CONTINUE
      IF (I8.EQ.I6) GO TO 10
      ROW(I6)=ROW(I8)
      ROW(I8)=I2
   10 IF (I8.EQ.I7) GO TO 20
      COL(I7)=COL(I8)
      COL(I8)=I3
   20 IF (ABS(X1/X3).GE.0.5E-8) GO TO 30
      WRITE (2,1) X1,I8
    1 FORMAT(///,22H  MXINV:PIVOT RATIO = ,F6.4,/,8H  STAGE ,I3)
   30 X3=X1
      DO 500 I9=1,N
  500 H(I2,I9)=H(I2,I9)/X1
```

```
      DO 600  I1=1,N
      IF (I1.EQ.I2) GO TO 600
      X4=H(I1,I3)
      DO 700  I9=1,N
  700 H(I1,I9)=H(I1,I9)-H(I2,I9)*X4
      H(I2,I3)=-X4/X1
  600 CONTINUE
      H(I2,I3)=1.0/X1
  200 CONTINUE
      DO 800  I1=1,N
      DO 900  I2=1,N
  900 MK(I2)=H(I1,I2)
      DO 110  I2=1,N
      K1=COL(I2)
      K2=ROW(I2)
  110 H(I1,K1)=MK(K2)
  800 CONTINUE
      DO 210  I1=1,N
      DO 310  I2=1,N
  310 MK(I2)=H(I2,I1)
      DO 410  I2=1,N
      K1=COL(I2)
      K2=ROW(I2)
  410 H(K1,I1)=MK(K2)
  210 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE QUADMIN(ANORM,FI,P,T,X,N)
      DIMENSION FI(N),P(N),X(N),PHI(3),VT(3)
      ANORM1=ANORM
      TLAST=1.0
      T=0.0
      IT=0
    1 IT=IT+1
      CALL FUNCTION(FI,X,N)
      CALL NORM(ANORM,FI,N)  ← IF (ANORM.LT.ANORM1) RETURN
      IF (((ABS(T-TLAST)).GT.(0.01*ABS(T)).AND.(IT.LE.10)).OR.(IT.EQ.2))
     1GO TO 2
      RETURN
    2 IF (IT.NE.1) GO TO 3
      VT(1)=0.0
      VT(3)=1.0
      PHI(1)=ANORM1
      PHI(3)=ANORM
      XX=ANORM/ANORM1
      T=(SQRT(1.0+6.0*XX)-1.0)/(3.0*XX)
      I=2
      K=1
      L=2
      M=3
      CALL SET(X,VT,P,N,T,TLAST,I1,I)
      GO TO 1
    3 PHI(I1)=ANORM
      XW=VT(2)-VT(3)
      XX=VT(3)-VT(1)
      XY=VT(1)-VT(2)
      XW=-(PHI(1)*XW+PHI(2)*XX+PHI(3)*XY)/(XW*XX*XY)
      XX=(PHI(1)-PHI(2))/XY-XW*(VT(1)+VT(2))
      TLAST=T
      IF (XW.GT.0.0) GO TO 5
      IF (PHI(M).GT.PHI(K)) GO TO 4
      T=3.0*VT(M)-2.0*VT(L)
      GO TO 6
    4 T=3.0*VT(K)-2.0*VT(L)
      GO TO 6
```

```
5 T=-XX/(2.0*XW)
6 IF (T.LE.VT(M)) GO TO 7
  I=K
  K=L
  L=M
  M=I
  CALL SET(X,VT,P,N,T,TLAST,I1,M)
  GO TO 1
7 IF (T.GE.VT(K)) GO TO 8
  I=M
  M=L
  L=K
  K=I
  CALL SET(X,VT,P,N,T,TLAST,I1,K)
  GO TO 1
8 IF (T.LE.VT(L)) GO TO 9
  I=K
  K=L
  L=I
  CALL SET(X,VT,P,N,T,TLAST,I1,L)
  GO TO 1
9 I=M
  M=L
  L=I
  CALL SET(X,VT,P,N,T,TLAST,I1,L)
  GO TO 1
  END



  SUBROUTINE SET(X,VT,P,N,T,TLAST,I1,I)
  DIMENSION X(N),P(N),VT(3)
  I1=I
  VT(I)=T
  DO 1 J=1,N
  X(J)=X(J)+(T-TLAST)*P(J)
1 CONTINUE
  RETURN
  END
```

APPENDIX IV.

A4.1) Introduction.

The subroutine in this appendix is an example of a user definition of a function. In this case the function defined in Case 1 Equation B of the industrial problem solved in Chapter 7. The routine is written in FORTRAN IV to be inserted in the main program.

```
      SUBROUTINE FUNCTION(FI,X,N)
      DIMENSION FI(N),X(N)
      REAL MS,MW2
      W=125.0
      WSQ=W*W
      MW2=125.0
      MS=1.5
      AXX=352000.0
      AXY=-129000.0
      AXXD=1300.0
      AXYD=1520.0
      AXXSQ=202500000.0
      AXYSQ=-95300000.0
      AXXY=235000000.0
      AXXDSQ=244.7
      AXYDSQ=962.0
      AXXDYD=783.0
      AXXXD=115500.0
      AXXYD=1477000.0
      AXYYD=923000.0
      AXYXD=1200000.0
      AYX=1174000.0
      AYY=783000.0
      AYXD=1520.0
      AYYD=8300.0
      AYXSQ=170000000.0
      AYYSQ=157000000.0
      AYXY=849000000.0
      AYXDSQ=24.5
      AYYDSQ=2120.0
      AYXDYD=604.0
      AYXXD=-1386000.0
      AYXYD=2310000.0
      AYYYD=6460000.0
      AYYXD=2310000.0
      FI(1)=AXX*X(1)+AXY*X(4)+AXXSQ*(X(1)*X(1)+((X(2)*X(2)+X(3)*X(3))/2.
     1))+AXYSQ*(X(4)*X(4)+((X(5)*X(5)+X(6)*X(6))/2.0))+AXXY*(X(1)*X(4)+(
     2(X(2)*X(5)+X(3)*X(6))/2.0))+AXXDSQ*WSQ*((X(2)*X(2)+X(3)*X(3))/2.0)
     3+AXYDSQ*WSQ*((X(5)*X(5)+X(6)*X(6))/2.0)+AXXDYD*WSQ*((X(2)*X(5)+X(3
     4)*X(6))/2.0)+AXXYD*W*((X(3)*X(5)-X(2)*X(6))/2.0)+AXXDY*W*((X(2)*X(
     56)-X(3)*X(5))/2.0)
      FI(2)=(AXX-MS*WSQ)*X(2)+AXY*X(5)-AXXD*W*X(3)-AXYD*W*X(6)+AXXSQ*(2.
     1*X(1)*X(2))+AXYSQ*2.0*X(4)*X(5)+AXXY*(X(2)*X(4)+X(1)*X(5))-AXXXD*W
     2*X(1)*X(3)-AXXDY*W*X(3)*X(4)-AXYYD*W*X(4)*X(6)-AXXYD*W*X(1)*X(6)
      FI(3)=(AXX-MS*WSQ)*X(3)+AXY*X(6)+AXXD*W*X(2)+AXYD*W*X(5)+AXXSQ*2.*
     1X(1)*X(3)+AXYSQ*2.0*X(4)*X(6)+AXXY*(X(3)*X(4)+X(1)*X(6))+AXXXD*W*X
     2(1)*X(2)+AXXYD*W*X(1)*X(5)+AXXDY*W*X(2)*X(4)+AXYYD*W*X(4)*X(5)-MW2
      FI(4)=AYX*X(1)+AYY*X(4)+AYXSQ*(X(1)*X(1)+((X(2)*X(2)+X(3)*X(3))/2.
     1))+AYYSQ*(X(4)*X(4)+((X(5)*X(5)+X(6)*X(6))/2.0))+AYXY*(X(1)*X(4)+(
     2(X(2)*X(5)+X(3)*X(6))/2.0))+AYXDSQ*WSQ*((X(2)*X(2)+X(3)*X(3))/2.0)
     3+AYYDSQ*WSQ*((X(5)*X(5)+X(6)*X(6))/2.0)+AYXDYD*WSQ*((X(2)*X(5)+X(3
     4)*X(6))/2.0)+AYXYD*W*((X(3)*X(5)-X(2)*X(6))/2.0)+AYXDY*W*((X(2)*X(
     56)-X(3)*X(5))/2.0)
      FI(5)=(AYY-MS*WSQ)*X(5)+AYX*X(2)-AYXD*W*X(3)-AYYD*W*X(6)+AYXSQ*(2.
     1*X(1)*X(2))+AYYSQ*2.0*X(4)*X(5)+AYXY*(X(2)*X(4)+X(1)*X(5))-AYXXD*W
     2*X(1)*X(3)-AYXDY*W*X(3)*X(4)-AYYYD*W*X(4)*X(6)-AYXYD*W*X(1)*X(6)-M
     3W2
      FI(6)=AYX*X(3)+(AYY-MS*WSQ)*X(6)+AYXD*W*X(2)+AYYD*W*X(5)+AYXSQ*2.*
     1X(1)*X(3)+AYYSQ*2.0*X(4)*X(6)+AYXY*(X(3)*X(4)+X(1)*X(6))+AYXXD*W*X
     2(1)*X(2)+AYXYD*W*X(1)*X(5)+AYXDY*W*X(2)*X(4)+AYYYD*W*X(4)*X(5)
      RETURN
      END
```

APPENDIX V.

A5.1) Introduction.

It is required to solve the system of equations

$$\underset{\sim}{f}(\underset{\sim}{x}) = \underset{\sim}{0}$$

using a method which is defined by

$$x_i^{n+1} = x_i^{n} - \frac{A}{\partial A_i / \partial x_i} \qquad (V.1)$$

where A_i is an auxiliary function.

A5.1.1) New Method.

In this method the auxiliary function is defined by

$$A_1 = f_1$$

$$A_2 = A_1^* + f_2$$

$$\cdot \quad \cdot \quad \cdot$$

$$A_n = A_{n-1}^* + f_n$$

Each auxiliary function is dependent upon the previous auxiliary function. This meant that cross coupling is introduced and therefore the elements of the vector $\underset{\sim}{x}$ become linearly dependent. Therefore, convergence to a solution is forced.

A5.1.2) Simplest Auxiliary Function.

The auxiliary function defined in 5.1.1. is not the simplest combination possible. The simplest combination is

$$A_1 = f_1$$

$$A_2 = f_2$$

$$\cdot \quad \cdot \quad \cdot$$

$$A_n = f_n$$

Each auxiliary function is independent so on iteration A_1 is a function of f_1 alone, and so on. Therefore

A5.1.2) contd.

using equation V.1 to define an iteration method means that each variable can converge separately to its own limit and therefore not converge to a solution point.

REFERENCES.

1) I.C.L. 1900 — "Programming Manual for Slang" Hawker Siddeley Dynamics Ltd., 1968.

2) I.C.L. 1900 — "Analogue Computer Simulation Program Mk.1." ICL 1900 Publication, 1969.

3) ORTEGA, J., and RHEINBOLT, W. — "Iterative Solution of Nonlinear Equations in Several Variables". pp. 181-189, Academic Press, New York, 1970.

4) LOHR, L., and RALL, L. — "Efficient use of Newton's method". ICC Bull.6, pp.99-103, 1967.

5) BROYDEN,C.G. — "A class of methods for solving non-linear simultaneous equations." Math.Comp.19, pp.577-593, 1965.

6) SOKOLNIKOFF,I.S., and REDHEFFER, R.M. — "Mathematics of Physics and Modern Engineering". pp.218, McGraw-Hill, New York.

7) HOUSEHOLDER,A.S. — "Principles of Numerical Analysis". pp.135-138, McGraw-Hill, New York, 1953.

8) WOLFE, P. — "The secant method for simultaneous nonlinear equations". Comm.A.C.M.2, pp.12-13, 1959.

9) BROWN, K., and CONTE, S. — "The solution of simultaneous nonlinear equations." Proc.22nd Nat.Conf.Assoc.Comp. Mach.,pp.111-114, Thompson Book Co., Washington D.C., 1967.

10) WILKINSON,J.H. — "The Algebraic Eigenvalue Problem". pp.200-205, Oxford, 1965.

11) ROSENBROOK,H.H. — "An automatic method for finding the greatest or least value of a function." Comput.J. 3, pp.175-184, 1960.

12) KUO, M. "Solution of nonlinear equations." IEEE Trans.Computers. 17, pp.897-898, 1968.

13) CUNNINGHAM,W.J. "Introduction to Nonlinear Analysis". pp.157-164, McGraw-Hill, 1958.

14) CUNNINGHAM,W.J. "Introduction to Nonlinear Analysis". pp.164-168, McGraw Hill, 1958.

15) CUNNINGHAM,W.J. "Introduction to Nonlinear Analysis". McGraw-Hill, 1958.

16) BANNISTER, R.H. "Dynamic Characteristics under Synchronous Out-of-Balance of a 5.0" Dia.x5.0" Lg. Bearing operating in the Laminar region". University of Aston, M.Sc. in Mechanics of Solids. October 1968.

17) PORTER,B. "Stability Criteria for linear dynamical systems". pp.116-137, Oliver and Boyd.

18) PORTER, B. "Stability Criteria for linear dynamical systems". Oliver and Boyd.

19) SAATY,T.L., and BRAM, J. "Nonlinear Mathematics". pp.235-250, McGraw Hill, New York, 1964.

20) BROYDEN,C.G. "A new method for solving nonlinear simultaneous equations." Comput.J.12, pp.94-99, 1969.

21) DAVIDENKO,D.R. "On a new method of Numerical Solution of Systems of Nonlinear Equations." Doklady Akad,Nauk SSSR(NS) pp.601-602, 88, 1953.

22) BRANIN,F.H., and HOO, S.K. "A method for finding multiple extrema of a function of a variable". Numerical Methods for Nonlinear Optimization, Applied Press, 1972.