1972

Digital Solution of Power-flow Problems by Newton's Method of Using a Hybrid Matrix

Prem K. Bahl

DIGITAL SOLUTION OF POWER-FLOW PROBLEMS BY NEWTON'S METHOD USING A HYBRID MATRIX

BY

PREM K. BAHL

A thesis submitted
in partial fulfilment of the requirement for the
degree Master of Science, Department of
Electrical Engineering, South Dakota
State University

March, 1972

DIGITAL SOLUTION OF POWER-FLOW PROBLEMS BY NEWTON'S METHOD USING A HYBRID MATRIX

This thesis is approved as a creditable and independent investigation by a candidate for the degree, Master of Science, and is acceptable as meeting the thesis requirements for this degree. Acceptance of this thesis does not imply that the conclusions reached by the candidate are necessarily the conclusions of the major department.

Thesis Adviser — Date

Head, Electrical Engineering Department — Date

ACKNOWLEDGEMENTS

The author wishes to express his deep gratitude to Dr. Junis Storry, Dean, College of Engineering, for his gracious help, excellent advice, and inspiration which made this investigation possible. The author's sincere thanks are due Dr. Frank Fitchen and staff of the Electrical Engineering Department and Mathematics Department whose guidance and encouragement throughout the graduate program made his education at the South Dakota State University a most valuable and rewarding experience.

Sulaksh, the author's wife, deserves a special word of thanks for typing his thesis and for having borne with him with patience and understanding during his graduate studies.

PKB

TABLE OF CONTENTS

APPENDICES

LIST OF TABLES

LIST OF ILLUSTRATIONS

CHAPTER I

INTRODUCTION

The last decade and a half has witnessed dramatic developments in the application of digital computers for solving power-flow problems. Previously these problems were analyzed on the direct analog computers called a-c calculating boards. With the enormous growth of the interconnected power systems during this period of time, digital computers established a distinct advantage over the analog computers for such reasons as:

(a) Their ability to analyze large-size systems (with such features as automatic tap setting, automatic area interchange control, and control of reactive constraints of generators).

(b) Elimination of human error in reading data and recording information on the system diagram.

(c) Accessibility and economy in making only a few changes from the base case.

(d) Availability of additional information such as the total transmission loss by easy extension of the power-flow program.

The power-flow problem can be solved by both direct and iterative methods. In fact, all the methods are iterative in the sense that the load flow problem involves the solution of a system of nonlinear equations. However, the so-called direct methods employ the direct solution of a related linear system in the iterative algorithm, whereas the iterative methods use a scheme of successive displacements such as Gauss-Seidel. Newton's method has an advantage over an iterative method because of its much faster (quadratic) convergence to a solution, thus saving computer time. The usual approach has been to use the bus admittance matrix for the network-defining equations.

The purpose of this investigation has been to apply Newton's method for the solution of power-flow problems employing a hybrid matrix for the network-defining equations in order to confirm the possibility of affecting further saving in computer time.

A sample 6-bus problem was solved on an IBM 360 Model 40 computer with 128 K core memory with single precision programming for the precision indices of 1×10^{-3} and 1×10^{-5} for real and reactive power mismatches at the busses. A double precision program was written for the precision index of 5×10^{-7}.

The hybrid matrix was formed by considering generator busses (1 and 2) as voltage-corrected and load busses (3 to 6) as current-corrected. Bus 1 is considered the swing bus.

CHAPTER II

LITERATURE REVIEW

The application of digital computers to the solution of power-flow problems has advanced the possibility of making system planning a continuous process rather than a periodic review as was previously done on a-c calculating boards.

Ward and Hale (1) presented an iterative method for the digital solution of power-flow problems, adopting the node basis of formulating network equations. Load and generator nodes are treated differently, because at the load terminals the specified quantities are real and reactive powers, whereas at the generator nodes the known quantities are voltage magnitude and real power. However, one generator bus is designated as slack bus or swing bus where only the voltage magnitude and phase angle are specified, but neither the real nor the reactive power is known. This provides the additional real and reactive power to supply the transmission losses which remain unknown until the final solution is obtained. An initial estimate of voltages is made at each bus. The iterative method proceeds from a trial set of terminal voltages to converge upon a corrected set of voltages which satisfy prescribed terminal inputs at each bus except the swing bus. The convergence rate is inherently slow and in certain cases the problem will 'explode' or diverge even if a solution exists. Ward and Hale made a significant contribution in forming an effective basis for the present computer analysis of load flow problems.

Brown and Tinney (2) improved the iterative process developed by Ward and Hale by applying acceleration factors to increase the speed of convergence.

Van Ness (3) formulated a coefficient matrix of partial derivatives of power with respect to voltage magnitude and phase angle from the equations for real and reactive power mismatches. This is called the Jacobian matrix. The corrections for voltage magnitude and angle are calculated for an iterative process by multiplying the inverted Jacobian matrix with the column vector of power mismatches.

Hale and Goodrich (4) originated the idea of using a hybrid matrix instead of an admittance matrix for the network-defining equations. This approach considers generator busses as voltage-corrected and load busses as current-corrected. The hybrid matrix method resulted in an 80% reduction in the number of iterations for convergence to the solution compared with the admittance matrix method.

Shipley and Coleman (5) developed a special purpose matrix inversion technique which is particularly applicable to the inversion of matrices that describe power system impedences and admittances. The inversion process proceeds to exchange one of the dependent variables (say voltage or current) with one of the independent variables (say current or voltage). The whole procedure may be summarized in four rules.

Van Ness and Griffin (6) adapted the elimination method to the power system equations. This method resulted in a much faster convergence to the solution. It also solved some problems that cannot be solved by the iteration method.

Storry (7) developed a method for getting a qualitative idea of the speed of convergence of the two iterative schemes employing the admittance matrix and hybrid matrix for the system equations. Methods for forming and altering a hybrid matrix were also developed.

Tinney and Hart (8) greatly improved the efficiency of the Newton's method for solution of power-flow problems. Newton's method offers quadratic convergence. Only 5 iterations, each equivalent to about 7 of the Gauss-Seidel method, are required for a solution.

CHAPTER III

FORMULATION OF THE HYBRID MATRIX

Forming the Hybrid Matrix of the Network-Defining Equations from the Bus Admittance Matrix

The elements of the bus admittance matrix, [Y], are the short-circuit driving-point and transfer admittances of the network. The driving-point admittance, Y_{kk}, is the sum of the admittances of all the components connected to the bus k. The transfer admittance, Y_{km}, is the negative of the sum of the admittances of the components connected between the busses k and m.

The elements of the hybrid matrix, [H], are the driving-point and transfer impedences and admittances of the network and dimensionless transfer ratios whose values are different than those of the original bus impedence and bus admittance matrices.

In matrix notation, the performance equation in the admittance form is:

$$[I] = [Y][E] \qquad (3.1)$$

where

[E] = matrix of bus voltages measured with respect to the reference bus

[I] = matrix of impressed bus currents

[Y] = bus admittance matrix

Considering that [Y] is an n x n matrix, the network equation (1) can be written more generally as:

$$\begin{bmatrix} I_i \\ --- \\ I_n \end{bmatrix} = \left[\begin{array}{c|c} Y_{ii} & Y_{in} \\ \hline Y_{ni} & Y_{nn} \end{array}\right] \begin{bmatrix} E_i \\ --- \\ E_n \end{bmatrix} \qquad (3.2)$$

Here, the terminal pairs are separated into two groups, one group consisting of 1 to i and the other of (i+1) to n. The position of I_n and E_n in (2) can be interchanged with the result that

$$\begin{bmatrix} I_i \\ --- \\ E_n \end{bmatrix} = \left[\begin{array}{c|c} Y_{ii} - Y_{in}Y_{nn}^{-1}Y_{ni} & Y_{in}Y_{nn}^{-1} \\ \hline -Y_{nn}^{-1}Y_{ni} & Y_{nn}^{-1} \end{array}\right] \begin{bmatrix} E_i \\ --- \\ I_n \end{bmatrix} \qquad (3.3)$$

or

$$\begin{bmatrix} I_i \\ --- \\ E_n \end{bmatrix} = \left[\begin{array}{c|c} Y_{ii} & D_{in} \\ \hline D_{ni} & Z_{nn} \end{array}\right] \begin{bmatrix} E_i \\ --- \\ I_n \end{bmatrix} \qquad (3.4)$$

where the Y's, D's, and Z's are admittances, dimensionless transfer ratios, and impedences, respectively. It should be remembered that Y_{ii} and Z_{nn} of (3.4) are different than those appearing in (3.2). Y_{ii} contains the driving-point and transfer admittances of terminal pairs 1-i with terminal pairs (i+1)-n open-circuited and Z_{nn} contains the driving-point and transfer impedences of terminal pairs (i+1)-n with terminal pairs 1-i short-circuited. D_{in} is a set of current transfer ratios and D_{ni} is a set of voltage transfer ratios. Thus, this general approach to the problem is designated as the transfer-ratio method.

The coefficient matrix, [H], in (3.4) is called the hybrid matrix because it is a mixed matrix of admittance, impedence, and dimensionless elements.

The terminal pairs are divided into two groups such that the generator busses are voltage-corrected, and the load busses are current-corrected. This means that the voltage magnitude is fixed at the generator busses but the phase angle is adjusted. At the load busses, the current magnitude as well as the phase angle is adjusted.

When the variable I_n is interchanged with E_n in (3.2), Y_{nn} is said to be the pivot element. Provided the pivot element is non-zero, the formation of the hybrid matrix may be summarized in the following four rules:

1. Any element other than that in the pivot row and pivot column (say Y_{lm}) is replaced by $(Y_{lm} - Y_{ln}Y_{nn}^{-1}Y_{nm})$ where Y_{nn} is the pivot element.
2. The pivot column elements except the pivot element are divided by the pivot element.
3. The pivot row elements except the pivot element are divided by the pivot element with a change of sign.
4. The pivot element is replaced by its reciprocal.

The exchange of variables is carried out with the matrix elements in place. The successive steps of calculation must use the original value of the elements in the preceding matrix.

If all the dependent variables of (3.2) are interchanged with all the independent variables, an inversion of the coefficient matrix results. The above four rules of the inversion process as applied to the coefficient matrix of the power network equations are adopted from the method introduced by Shipley and Coleman (5). This inversion process is applicable to matrices in general. However, all matrices do not have an inverse.

CHAPTER IV

THE THEORY OF NEWTON"S METHOD

It is well known that most functions of one variable have Taylor series representation written as

$$f(x) = f(a) + f'(a)(x - a) + \frac{f''(a)}{2!}(x - a)^2 + \dots + \frac{f^n(a)}{n!}(x - a)^n + \dots \quad (4.1)$$

This series is known to converge rapidly for values of x near a. As an approximation, assume convergence after the first two terms. Now if a is replaced by x and x by $(x + \Delta x)$, the series becomes

$$y = f(x + \Delta x) = f(x) + f'(x)\Delta x \quad (4.2)$$

(4.2) suffices for one unknown (x).

Let $x^{(0)}$ be the first estimate and $x^{(1)}$ the second to the root of the equation $f(x) = 0$, where

$$x^{(1)} = x^{(0)} + \Delta x$$

Substituting into (4.2),

$$f(x^{(1)}) = f(x^{(0)}) + f'(x^{(0)})(x^{(1)} - x^{(0)}) \quad (4.3)$$

Now if $x^{(1)}$ is to be a closer approximation than $x^{(0)}$, it could be assumed that the curve of (4.3) crosses the x-axis where $y^{(1)} = f(x^{(1)}) = 0$

Then

$$0 = f(x^{(0)}) + f'(x^{(0)})(x^{(1)} - x^{(0)}) \tag{4.4}$$

Solving (4.4) for the second approximation in terms of the first,

$$x^{(1)} = x^{(0)} - \frac{f(x^{(0)})}{f'(x^{(0)})} \tag{4.5}$$

Successive approximations follow and, in general, for the (k+1) approximation

$$x^{(k+1)} = x^{(k)} - \frac{f(x^{(k)})}{f'(x^{(k)})} \tag{4.6}$$

which is a special case of one equation with one unknown. The iteration process is merely continued until the original equation is satisfied within a preassigned tolerance.

If (4.2) is applied to a whole set of n nonlinear equations in n unknowns $(x_1, x_2, \cdots, x_n)$, the result is

$$y_1 = f_1(x_1, x_2, \cdots, x_n) + \Delta x_1 \frac{\partial f_1}{\partial x_1} + \Delta x_2 \frac{\partial f_1}{\partial x_2} + \cdots + \Delta x_n \frac{\partial f_1}{\partial x_n}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots$$

$$y_n = f_n(x_1, x_2, \cdots, x_n) + \Delta x_1 \frac{\partial f_n}{\partial x_1} + \Delta x_2 \frac{\partial f_n}{\partial x_2} + \cdots + \Delta x_n \frac{\partial f_n}{\partial x_n}$$

---(4.7)

As a first approximation, let the unknowns be represented by $x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}$. Then the first iteration for obtaining a second approximation is

$$y_1 = f_1(x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}) + \Delta x_1^{(0)} \frac{\partial f_1}{\partial x_1}\bigg|_0 + \ldots + \Delta x_n^{(0)} \frac{\partial f_1}{\partial x_n}\bigg|_0$$

....

$$y_n = f_n(x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}) + \Delta x_1^{(0)} \frac{\partial f_n}{\partial x_1}\bigg|_0 + \ldots + \Delta x_n^{(0)} \frac{\partial f_n}{\partial x_n}\bigg|_0$$

----(4.8)

The term $f_k(x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)})$ may henceforth be referred to as $f_k^{(0)}$. The values of $y_1, y_2, \ldots, y_n$ are normally known. The partial derivatives can be calculated as numerical quantities as can be $f_1^{(0)}, f_2^{(0)}, \ldots, f_n^{(0)}$. The unknowns are the $\Delta x^{(0)}$ quantities. (4.8) may be re-written as

$$y_1 - f_1^{(0)} = \frac{\partial f_1}{\partial x_1}\bigg|_0 \Delta x_1^{(0)} + \frac{\partial f_1}{\partial x_2}\bigg|_0 \Delta x_2^{(0)} + \ldots + \frac{\partial f_1}{\partial x_n}\bigg|_0 \Delta x_n^{(0)}$$

$$y_2 - f_2^{(0)} = \frac{\partial f_2}{\partial x_1}\bigg|_0 \Delta x_1^{(0)} + \frac{\partial f_2}{\partial x_2}\bigg|_0 \Delta x_2^{(0)} + \ldots + \frac{\partial f_2}{\partial x_n}\bigg|_0 \Delta x_n^{(0)}$$

....

$$y_n - f_n^{(0)} = \frac{\partial f_n}{\partial x_1}\bigg|_0 \Delta x_1^{(0)} + \frac{\partial f_n}{\partial x_2}\bigg|_0 \Delta x_2^{(0)} + \ldots + \frac{\partial f_n}{\partial x_n}\bigg|_0 \Delta x_n^{(0)}$$

----(4.9)

The original nonlinear equations have been changed into linear equations in the unknown variables $\Delta x_1^{(0)}$, $\Delta x_2^{(0)}$,

...., $\Delta x_n^{(0)}$ for the first iteration. After having solved for these incremental unknowns, the new values of x for the next iteration will be

$$x_1^{(1)} = x_1^{(0)} + \Delta x_1^{(0)}$$

$$x_2^{(1)} = x_2^{(0)} + \Delta x_2^{(0)}$$

....

$$x_n^{(1)} = x_n^{(0)} + \Delta x_n^{(0)} \tag{4.10}$$

(4.9) can also be expressed in matrix form as

$$\begin{bmatrix} y_1 - f_1^{(0)} \\ y_2 - f_2^{(0)} \\ \cdots \\ y_n - f_n^{(0)} \end{bmatrix} = \begin{bmatrix} \left.\frac{\partial f_1}{\partial x_1}\right|_0 & \left.\frac{\partial f_1}{\partial x_2}\right|_0 \cdots & \left.\frac{\partial f_1}{\partial x_n}\right|_0 \\ \left.\frac{\partial f_2}{\partial x_1}\right|_0 & \left.\frac{\partial f_2}{\partial x_2}\right|_0 \cdots & \left.\frac{\partial f_2}{\partial x_n}\right|_0 \\ \cdots & \cdots & \cdots \\ \left.\frac{\partial f_n}{\partial x_1}\right|_0 & \left.\frac{\partial f_n}{\partial x_2}\right|_0 \cdots & \left.\frac{\partial f_n}{\partial x_n}\right|_0 \end{bmatrix} \begin{bmatrix} \Delta x_1^{(0)} \\ \Delta x_2^{(0)} \\ \\ \Delta x_n^{(0)} \end{bmatrix} \tag{4.10a}$$

The matrix of partial derivative coefficients is referred to as the Jacobian matrix.

The iteration process of (4.9) is repeated until the Δx values are less than a preassigned precision index. This process is often accomplished by inverting the Jacobian in (4.10a), and multiplying the discrepancies in the y's by this inverse to give the Δx's.

CHAPTER V

APPLICATION OF NEWTON'S METHOD FOR POWER-FLOW PROBLEMS USING HYBRID MATRIX

Power-flow studies which normally pertain to balanced three-phase systems, and with calculations on a per phase, per unit basis, result in the determination of voltage magnitudes and phase angles for each node, phasor line currents, real and reactive power flow in the individual lines and line losses. The three kinds of nodes identified in a power network are shown in figure 5.1.

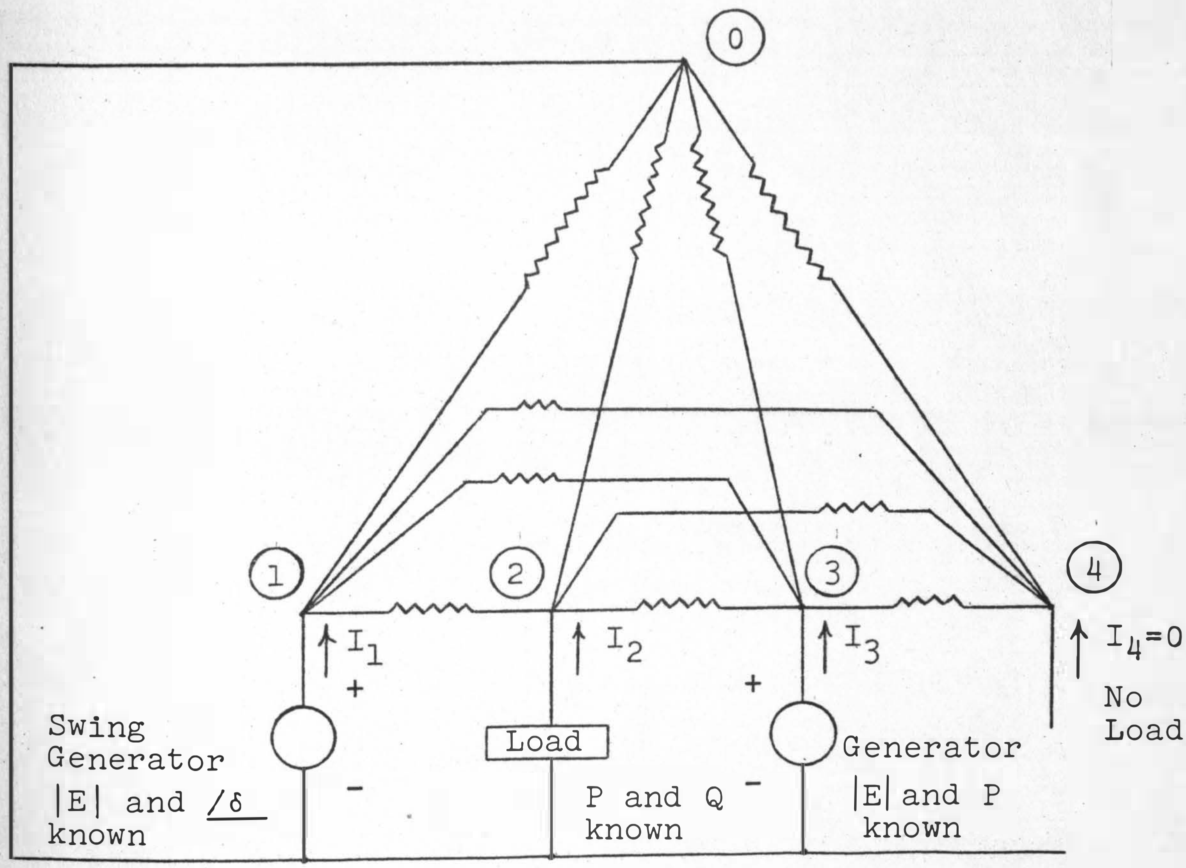

Fig. 5.1. Power network showing each type of node possible.

Symbols

n - number of busses

G - conductance

B - susceptance

Y = (G+jB) - admittance

H = (α+jB) - a complex hybrid matrix element - admittance, impedence or dimensionless transfer ratio

E = $|E| \underline{/\delta}$ - a complex voltage vector

I = $|I| \underline{/\phi}$ - a complex current vector

S = P-jQ = E*I - complex power

A. Form of Nonlinear Equations of Power Using Polar Coordinates

The network-defining equations in the hybrid matrix form can be represented as

$$\begin{bmatrix} I_1 \\ : \\ I_i \\ \\ E_k \\ : \\ E_n \end{bmatrix} = \begin{bmatrix} H_{11} & \cdot\cdot & H_{1i} & H_{1k} & \cdot\cdot & H_{1n} \\ : & & : & : & & : \\ H_{i1} & \cdot\cdot & H_{ii} & H_{ik} & \cdot\cdot & H_{in} \\ \\ H_{k1} & \cdot\cdot & H_{ki} & H_{kk} & \cdot\cdot & H_{kn} \\ : & & : & : & & : \\ H_{n1} & \cdot\cdot & H_{ni} & H_{nk} & \cdot\cdot & H_{nn} \end{bmatrix} \begin{bmatrix} E_1 \\ : \\ E_i \\ \\ I_k \\ : \\ I_n \end{bmatrix} \tag{5.1}$$

In (5.1) the generator busses 1-i are voltage-corrected and the load busses k-n are current-corrected, where k = i+1.

Generator Bus

From (5.1) we obtain

$$I_i = \sum_{m=1}^{i} H_{im}E_m + \sum_{l=k}^{n} H_{il}I_l \tag{5.2}$$

$$\begin{aligned} P_i - jQ_i &= E_i^* I_i \\ &= E_i^* \left[\sum_{m=1}^{i} H_{im}E_m + \sum_{l=k}^{n} H_{il}I_l \right] \end{aligned} \tag{5.3}$$

Expanding the above equation; we obtain

$$\begin{aligned} P_i - jQ_i = |E_i|(\cos\delta_i - j\sin\delta_i) \Bigg[\sum_{m=1}^{i} (\alpha_{im} + j\beta_{im})|E_m| \; x \\ (\cos\delta_m + j\sin\delta_m) + \sum_{l=k}^{n} (\alpha_{il} + j\beta_{il})|I_l| \; x \\ (\cos\phi_l + j\sin\phi_l) \Bigg] \end{aligned} \tag{5.4}$$

After expanding and separating the real and imaginary parts, the real and reactive powers are

$$P_i = \alpha_{ii}|E_i|^2 + |E_i| \sum_{m=1}^{i-1} |E_m|\Big[\alpha_{im} \cos (\delta_i-\delta_m)$$

$$+ \beta_{im} \sin (\delta_i-\delta_m)\Big] + |E_i| \sum_{l=k}^{n} |I_l| \ x$$

$$\Big[\alpha_{il} \cos (\delta_i-\phi_l) + \beta_{il} \sin (\delta_i-\phi_l)\Big] \qquad (5.5)$$

$$Q_i = |E_i| \sum_{m=1}^{i-1} |E_m|\Big[\alpha_{im} \sin (\delta_i-\delta_m) - \beta_{im} \cos(\delta_i-\delta_m)\Big]$$

$$- \beta_{ii}|E_i|^2 + |E_i| \sum_{l=k}^{n} |I_l|\Big[\alpha_{il} \sin (\delta_i-\phi_l)$$

$$- \beta_{il} \cos (\delta_i-\phi_l)\Big] \qquad (5.6)$$

Load Bus

From (5.1)

$$E_k = \sum_{m=1}^{i} H_{km}E_m + \sum_{l=k}^{n} H_{kl}I_l \qquad (5.7)$$

$$P_k - jQ_k = E_k{}^*I_k \qquad (5.8)$$

$$P_k - jQ_k = \left[\sum_{m=1}^{i} H_{km}{}^*E_m{}^* + \sum_{l=k}^{n} H_{kl}{}^*I_l{}^*\right] I_k$$

$$= \left[\sum_{m=1}^{i} (\alpha_{km} - j\beta_{km})|E_m|(\cos \delta_m - j \sin \delta_m) + \right.$$

$$\left.\sum_{l=k}^{n} (\alpha_{kl} - j\beta_{kl})|I_l|(\cos \phi_l - j \sin \phi_l)\right] x$$

$$|I_k|(\cos \phi_k + j \sin \phi_k) \text{--}(5.9)$$

whence we get

$$P_k = |I_k| \left\{ \sum_{m=1}^{i} |E_m| \left[\alpha_{km} \cos(\delta_m - \phi_k) - \beta_{km} \sin(\delta_m - \phi_k) \right] \right.$$

$$\left. + \sum_{l=k}^{n} |I_l| \left[\alpha_{kl} \cos(\phi_l - \phi_k) - \beta_{kl} \sin(\phi_l - \phi_k) \right] \right\} \quad \text{---(5.10)}$$

$$Q_k = |I_k| \left\{ \sum_{m=1}^{i} |E_m| \left[\alpha_{km} \sin(\delta_m - \phi_k) + \beta_{km} \cos(\delta_m - \phi_k) \right] \right.$$

$$\left. + \sum_{l=k}^{n} |I_l| \left[\alpha_{kl} \sin(\phi_l - \phi_k) + \beta_{kl} \cos(\phi_l - \phi_k) \right] \right\} \quad \text{---(5.11)}$$

B. Calculation of Real and Reactive Power Mismatches

Bus 1 may be considered as the swing bus where the phasor voltage is specified. Voltage magnitude and real power are fixed at the other generator busses 2-i. Real and reactive powers are known at the load busses k-n. It is customary to set the voltage magnitudes, where given, to their given values (as at the generator nodes). At all the load busses the voltage magnitude is assumed as unity as a first approximation. The use of per unit system is assumed. All voltage angles are set equal to that of the swing bus voltage. This is known as a "flat" voltage start.

The values of currents at the load busses k-n are calculated using the initially assumed voltages from the following equation.

$$I_k = S_k \text{ (scheduled)}/E_k^* \quad (5.12)$$

Currents at the generator nodes 1-i are computed from (5.12) as

$$I_i = \sum_{m=1}^{i} H_{im}E_m + \sum_{l=k}^{n} H_{il}I_l \qquad (5.13)$$

Since the phasor voltage at the swing bus remains the same throughout the process, no error equation is required for this bus. Therefore complex power at all except the swing bus is calculated as

$$P_p - jQ_p = E_p{*}I_p \qquad p = 2, 3, \cdots, n \qquad (5.14)$$

Real and reactive power mismatches are the differences between the scheduled and the calculated values.

$$\Delta P_p = P_p \text{ (scheduled)} - P_p \text{ (calculated)} \quad p = 2, 3, \cdots, n$$
$$\Delta Q_p = Q_p \text{ (scheduled)} - Q_p \text{ (calculated)} \quad p = k, \cdots, n$$

---(5.15)

C. Evaluation of the Jacobian Matrix Elements

Newton's method involves repeated direct solutions of a system of linear equations derived from (5.4) and (5.9). The Jacobian matrix of (5.4) and (5.9) establishes the linearized relationship between small errors in voltage and current angles and magnitudes, $\Delta\delta$, $\Delta\phi$, $\Delta|E|$, and $\Delta|I|$, and real and reactive power mismatches, ΔP and ΔQ. The Jacobian matrix elements are calculated by taking the partial derivatives of real and reactive powers given by (5.4) and (5.9) with respect to the voltage and current angles and magnitudes. At a generator bus, since the voltage magnitude is specified and the reactive power is not, the partial derivative of real power with respect to the voltage magnitude and that of the reactive power with respect to the voltage and current angles and magnitudes are ignored.

The matrix equation is as shown in (5.16).

$$
\begin{bmatrix} \Delta P_2 \\ \vdots \\ \Delta P_i \\ \Delta P_k \\ \Delta Q_k \\ \vdots \\ \Delta P_n \\ \Delta Q_n \end{bmatrix}
=
\begin{bmatrix}
\frac{\partial P_2}{\partial \delta_2} & \cdots & \frac{\partial P_2}{\partial \delta_i} & \frac{\partial P_2}{\partial \phi_k} & \frac{\partial P_2}{\partial I_k} & \cdots & \frac{\partial P_2}{\partial \phi_n} & \frac{\partial P_2}{\partial I_n} \\
\vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots \\
\frac{\partial P_i}{\partial \delta_2} & \cdots & \frac{\partial P_i}{\partial \delta_i} & \frac{\partial P_i}{\partial \phi_k} & \frac{\partial P_i}{\partial I_k} & \cdots & \frac{\partial P_i}{\partial \phi_n} & \frac{\partial P_i}{\partial I_n} \\
\frac{\partial P_k}{\partial \delta_2} & \cdots & \frac{\partial P_k}{\partial \delta_i} & \frac{\partial P_k}{\partial \phi_k} & \frac{\partial P_k}{\partial I_k} & \cdots & \frac{\partial P_k}{\partial \phi_n} & \frac{\partial P_k}{\partial I_n} \\
\frac{\partial Q_k}{\partial \delta_2} & \cdots & \frac{\partial Q_k}{\partial \delta_i} & \frac{\partial Q_k}{\partial \phi_k} & \frac{\partial Q_k}{\partial I_k} & \cdots & \frac{\partial Q_k}{\partial \phi_n} & \frac{\partial Q_k}{\partial I_n} \\
\vdots & & \vdots & \vdots & \vdots & & \vdots & \vdots \\
\frac{\partial P_n}{\partial \delta_2} & \cdots & \frac{\partial P_n}{\partial \delta_i} & \frac{\partial P_n}{\partial \phi_k} & \frac{\partial P_n}{\partial I_k} & \cdots & \frac{\partial P_n}{\partial \phi_n} & \frac{\partial P_n}{\partial I_n} \\
\frac{\partial Q_n}{\partial \delta_2} & \cdots & \frac{\partial Q_n}{\partial \delta_i} & \frac{\partial Q_n}{\partial \phi_k} & \frac{\partial Q_n}{\partial I_k} & \cdots & \frac{\partial Q_n}{\partial \phi_n} & \frac{\partial Q_n}{\partial I_n}
\end{bmatrix}
\begin{bmatrix} \Delta \delta_2 \\ \vdots \\ \Delta \delta_i \\ \Delta \phi_k \\ \Delta |I_k| \\ \vdots \\ \Delta \phi_n \\ \Delta |I_n| \end{bmatrix}
\qquad \text{-----(5.16)}
$$

Where, at the generator node i, we have

$$\frac{\partial P_i}{\partial \delta_2} = |E_i||E_2| \left[\alpha_{i2} \sin (\delta_i - \delta_2) - \beta_{i2} \cos (\delta_i - \delta_2) \right]$$

$$\frac{\partial P_i}{\partial \delta_i} = |E_i| \sum_{m=1}^{i-1} |E_m| \left[-\alpha_{im} \sin(\delta_i - \delta_m) + \beta_{im} \cos(\delta_i - \delta_m) \right]$$

$$+ |E_i| \sum_{l=k}^{n} |I_l| \left[-\alpha_{il} \sin(\delta_i - \phi_l) + \beta_{il} \cos(\delta_i - \phi_l) \right]$$

$$\frac{\partial P_i}{\partial \phi_k} = |E_i||I_k| \left[\alpha_{ik} \sin(\delta_i - \phi_k) - \beta_{ik} \cos(\delta_i - \phi_k) \right]$$

$$\frac{\partial P_i}{\partial I_k} = |E_i| \left[\alpha_{ik} \cos(\delta_i - \phi_k) + \beta_{ik} \sin(\delta_i - \phi_k) \right]$$

At the load bus k, we have

$$\frac{\partial P_k}{\partial \delta_i} = - |I_k||E_i| \left[\alpha_{ki} \sin(\delta_i - \phi_k) + \beta_{ki} \cos(\delta_i - \phi_k) \right]$$

$$\frac{\partial P_k}{\partial \phi_k} = |I_k| \sum_{m=1}^{i} |E_m| \left[\alpha_{km} \sin(\delta_m - \phi_k) + \beta_{km} \cos(\delta_m - \phi_k) \right]$$

$$+ |I_k| \sum_{l=k+1}^{n} |I_l| \left[\alpha_{kl} \sin(\phi_l - \phi_k) + \beta_{kl} \cos(\phi_l - \phi_k) \right]$$

$$+ \beta_{kk} |I_k|^2$$

$$\frac{\partial P_k}{\partial I_k} = \sum_{m=1}^{i} |E_m| \left[\alpha_{km} \cos(\delta_m - \phi_k) - \beta_{km} \sin(\delta_m - \phi_k) \right]$$

$$+ \sum_{l=k+1}^{n} |I_l| \left[\alpha_{kl} \cos(\phi_l - \phi_k) - \beta_{kl} \sin(\phi_l - \phi_k) \right]$$

$$+ 2\alpha_{kk} |I_k|$$

$$\frac{\partial Q_k}{\partial \delta_i} = |I_k||E_i| \left[\alpha_{ki} \cos(\delta_i - \phi_k) - \beta_{ki} \sin(\delta_i - \phi_k) \right]$$

$$\frac{\partial Q_k}{\partial \phi_k} = |I_k| \sum_{m=1}^{i} |E_m| \left[-\alpha_{km} \cos(\delta_m - \phi_k) + \beta_{km} \sin(\delta_m - \phi_k) \right]$$

$$+ |I_k| \sum_{l=k+1}^{n} |I_l| \left[-\alpha_{kl} \cos(\phi_l - \phi_k) + \beta_{kl} \sin(\phi_l - \phi_k) \right]$$

$$- \alpha_{kk} |I_k|^2$$

$$\frac{\partial Q_k}{\partial I_k} = \sum_{m=1}^{i} |E_m| \left[\alpha_{km} \sin(\delta_m - \phi_k) + \beta_{km} \cos(\delta_m - \phi_k) \right]$$

$$+ \sum_{l=k+1}^{n} |I_l| \left[\alpha_{kl} \sin(\phi_l - \phi_k) + \beta_{kl} \cos(\phi_l - \phi_k) \right]$$

$$+ 2\beta_{kk} |I_k|$$

D. Inversion of the Jacobian Matrix

The aforecited n x n Jacobian matrix may be considered in the following form:

$$\left[J^{(1)} \right] = \begin{bmatrix} J_{11}^{(1)} & J_{12}^{(1)} & \cdots & J_{1n}^{(1)} \\ J_{21}^{(1)} & J_{22}^{(1)} & \cdots & J_{2n}^{(1)} \\ \vdots & \vdots & & \vdots \\ J_{n1}^{(1)} & J_{n2}^{(1)} & \cdots & J_{nn}^{(1)} \end{bmatrix} \qquad (5.17)$$

This matrix is inverted by applying the four rules of the modified Shipley method, which are enumerated in Chapter III, to each of the n rows of the matrix. The hybrid matrix is formed by applying the rules fewer than n times.

The inversion process was used as follows:
The first diagonal element, $J_{11}^{(1)}$, was considered as the pivot element, provided that it was non-zero. The four rules of inversion used are repeated below.

1. Any element other than that in the pivot row and pivot column (say $J_{km}^{(1)}$) was replaced by

$$\left[J_{km}^{(1)} - \frac{J_{k1}^{(1)} J_{1m}^{(1)}}{J_{11}^{(1)}} \right]$$

2. The pivot column elements except the pivot element were divided by the pivot element.
3. The pivot row elements except the pivot element were divided by the pivot element with a change of sign.
4. The pivot element was replaced by its reciprocal.

The original matrix was transformed as

$$\left[J^{(2)}\right] = \begin{bmatrix} J_{11}^{(2)} & J_{12}^{(2)} & \cdots & J_{1n}^{(2)} \\ J_{21}^{(2)} & J_{22}^{(2)} & & J_{2n}^{(2)} \\ \vdots & \vdots & & \vdots \\ J_{n1}^{(2)} & J_{n2}^{(2)} & & J_{nn}^{(2)} \end{bmatrix}$$

Next, the diagonal element $J_{22}^{(2)}$ of the above matrix was considered as the pivot element which should be non-zero. The aforementioned four steps were carried out again. The inversion process was repeated with all the rest of the diagonal elements taken successively. When the final pivot element $J_{nn}^{(n)}$ was operated upon, the Jacobian matrix was inverted and had the following form:

$$\left[J^{(n+1)}\right] = \left[J^{(1)}\right]^{-1} = \begin{bmatrix} J_{11}^{(n+1)} & J_{12}^{(n+1)} & \cdots & J_{1n}^{(n+1)} \\ J_{21}^{(n+1)} & J_{22}^{(n+1)} & \cdots & J_{2n}^{(n+1)} \\ \vdots & \vdots & & \vdots \\ J_{n1}^{(n+1)} & J_{n2}^{(n+1)} & \cdots & J_{nn}^{(n+1)} \end{bmatrix} \qquad \text{-----(5.18)}$$

E. <u>Determination of Voltage and Current Angle Corrections and Current Magnitude Corrections</u>

The voltage and current angle corrections and current magnitude corrections are determined by multiplying the inverse of Jacobian matrix, $\left[J^{(1)}\right]^{-1}$, by the column vector of real and reactive power mismatches.

$$\begin{bmatrix} \Delta\delta_2 \\ \vdots \\ \Delta\delta_i \\ \Delta\phi_k \\ \Delta|I_k| \\ \vdots \\ \Delta\phi_n \\ \Delta|I_n| \end{bmatrix} = \begin{bmatrix} \\ \\ & J^{(1)} & \\ \\ \\ \end{bmatrix}^{-1} \begin{bmatrix} \Delta P_2 \\ \vdots \\ \Delta P_i \\ \Delta P_k \\ \Delta Q_k \\ \vdots \\ \Delta P_n \\ \Delta Q_n \end{bmatrix} \qquad (5.19)$$

F. Iterative Algorithm

The iterative process proceeds as follows:

1. As a first approximation, a flat voltage start, as described earlier, is made. Currents are assumed at the load busses, corresponding to the initially assumed values of voltages. Currents at the generator busses are computed from (5.13).
2. With the assumed values of voltages and currents, real power is calculated at each of the generator busses, and real and reactive powers are computed at each of the load busses.
3. Power mismatches, real and reactive, are found by subtracting the calculated values of power from the scheduled values.
4. The Jacobian matrix, (5.16), is formed, and its inverse is computed.
5. The voltage and current angle corrections and current magnitude corrections in polar form are found by multiplying the inverse of Jacobian matrix with the column matrix of power mismatches. The corrections are applied to node voltages and currents.

$$
\begin{aligned}
|I_k|^{(1)} &= |I_k|^{(0)} + \Delta|I_k|^{(0)} \\
\angle\phi_k^{(1)} &= \angle\phi_k^{(0)} + \angle\Delta\phi_k^{(0)} \\
|E_k|^{(1)} &= |E_k|^{(0)} + \Delta|E_k|^{(0)} \\
\angle\delta_k^{(1)} &= \angle\delta_k^{(0)} + \angle\Delta\delta_k^{(0)}
\end{aligned}
\qquad (5.20)
$$

6. The power mismatches, ΔP_k and ΔQ_k, are checked. If each of their absolute values is less than a preassigned precision index, the problem is solved. If not, the procedure is repeated by finding the new values of voltages and currents, starting with step 2.

CHAPTER VI

MODIFIED 6-BUS PROBLEM OF THE ORIGINAL WARD-HALE SYSTEM

The original 6-bus power-flow problem analysed by Ward and Hale (1) was modified in that bus 4 was interchanged with bus 6. This was done because bus 4 was a passive node having no load or generation on it, and the modification resulted in continuity in numbering the nodes and in ignoring node 6 (the last one) in the Jacobian matrix of (5.16). The loads were assumed to be inductive. The modified system is shown in figure 6.1.

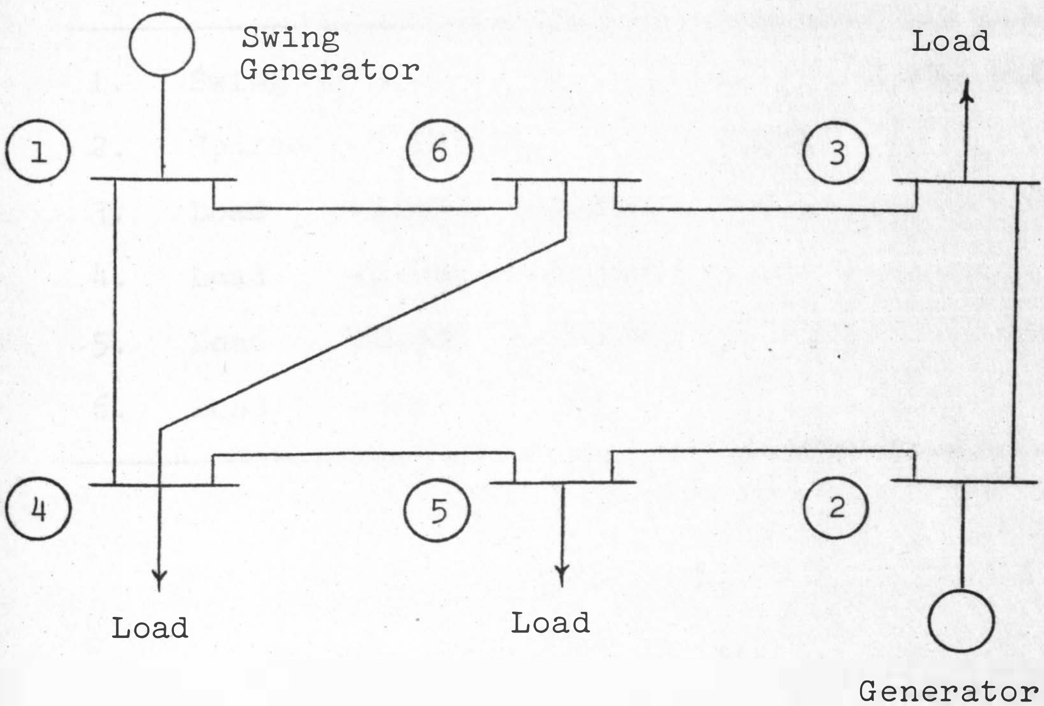

Figure 6.1. Sample 6-bus system.

The specified terminal conditions (in per unit) for the three kinds of busses - swing, generator, and load busses - are tabulated in Table 6.1. The elements of the bus admittance matrix representing the driving-point and transfer admittances of the 6-bus problem of figure 6.1 are given in Table 6.2.

Table 6.1

Specified Terminal Conditions

(Per Unit)

Bus	Type	P	Q	|E|	E
1.	Swing				1.050 0.0
2.	Source	0.500		1.100	
3.	Load	-0.550	-0.130		
4.	Load	-0.500	-0.050		
5.	Load	-0.300	-0.180		
6.	Load	0.0	0.0		

Table 6.2. Bus Admittance Matrix of the 6-Bus Problem of Figure 6.1.

Element k - m	G	B	Element k - m	G	B
1 1	0.992203	-4.375561	4 1	-0.433934	1.827462
1 2	0.0	0.0	4 2	0.0	0.0
1 3	0.0	0.0	4 3	0.0	0.0
1 4	-0.433934	1.827462	4 4	0.988036	-7.619402
1 5	0.0	0.0	4 5	0.0	3.416665
1 6	-0.558269	2.581996	4 6	-0.554102	2.324944
2 1	0.0	0.0	5 1	0.0	0.0
2 2	1.021400	-1.954525	5 2	-0.576541	1.308461
2 3	-0.444860	0.646063	5 3	0.0	0.0
2 4	0.0	0.0	5 4	0.0	3.416665
2 5	-0.576541	1.308461	5 5	0.576541	-4.641789
2 6	0.0	0.0	5 6	0.0	0.0
3 1	0.0	0.0	6 1	-0.558269	2.581996
3 2	-0.444860	0.646063	6 2	0.0	0.0
3 3	0.444860	-8.164860	6 3	0.0	8.270677
3 4	0.0	0.0	6 4	-0.554102	2.324944
3 5	0.0	0.0	6 5	0.0	0.0
3 6	0.0	8.270677	6 6	1.112370	-13.975357

CHAPTER VII

RESULTS AND CONCLUSIONS

Results

The 6-bus power-flow problem for the system shown in figure 6.1 was solved on an IBM 360 Model 40 digital computer with 128 K core memory. A program with single precision was written for the precision indices for real and reactive power mismatches having values of 1×10^{-3} and 1×10^{-5}. Double precision had to be used for the precision index of 5×10^{-7}.

First of all, the hybrid matrix was formed from the given bus admittance matrix shown in Table 6.2. The flow chart and the computer program for the formation of the hybrid matrix are given in the appendices A & B, respectively. The elements of the hybrid matrix are shown in Table 7.1.

With precision indices of 1×10^{-3} and 1×10^{-5} for the real and reactive power mismatches, the problem was solved in 6 and 7 iterations, respectively. The precision index of 5×10^{-7} also required 7 iterations.

Table 7.2 gives the values of voltages after the 1st, 6th, and 7th iterations at busses 2-6. These values are given separately for the three precision indices used in the problem. The flow chart and the computer program with double precision for the 6-bus

Table 7.1. Hybrid Matrix Formed From the Given Bus Admittance Matrix

Element k - m		α	β	Element k - m		α	β
1	1	0.475435	-1.198132	4	1	0.683189	-0.031716
1	2	-0.441680	1.208549	4	2	0.314003	0.026759
1	3	-0.735388	0.090973	4	3	0.032886	0.089911
1	4	-0.683189	0.031716	4	4	0.059040	0.223479
1	5	-0.492377	0.084501	4	5	0.062918	0.156680
1	6	-0.730873	0.050254	4	6	0.027629	0.090529
2	1	-0.441680	1.208549	5	1	0.492377	-0.084501
2	2	0.411233	-1.127419	5	2	0.522822	0.078965
2	3	-0.346483	-0.079373	5	3	0.031933	0.062214
2	4	-0.314003	-0.026759	5	4	0.062918	0.156680
2	5	-0.522822	-0.078965	5	5	0.086067	0.320071
2	6	-0.259666	-0.043207	5	6	0.028178	0.063136
3	1	0.735388	-0.090973	6	1	0.730873	-0.050254
3	2	0.346483	0.079373	6	2	0.259666	0.043207
3	3	0.088788	0.320664	6	3	0.070404	0.200428
3	4	0.032886	0.089911	6	4	0.027629	0.090529
3	5	0.031933	0.062214	6	5	0.028178	0.063136
3	6	0.070404	0.200428	6	6	0.058723	0.201651

Table 7.2. Voltage Solution

Precision Index	Bus	Iteration 1st		Iteration 6th		Iteration 7th	
		|E|	Degree	|E|	Degree	|E|	Degree
1×10^{-3}	2	1.099999	0.0	1.099991	- 3.358966		
	3	1.166800	-13.234833	1.000531	-12.786680		
	4	1.072420	-12.417873	0.919073	-12.240866		
	5	1.115445	-12.069226	0.919271	-12.337496		
	6	1.052283	-10.695078	0.929600	- 9.836724		
1×10^{-5}	2	1.099999	0.0			1.099989	- 3.358984
	3	1.166800	-13.234833			1.000529	-12.786666
	4	1.072420	-12.417873			0.919068	-12.240831
	5	1.115445	-12.069226			0.919262	-12.337479
	6	1.052283	-10.695078			0.929598	- 9.836741
5×10^{-7}	2	1.100000	- 0.0			1.100000	- 3.359151
	3	1.166803	-13.234823			1.000536	-12.786609
	4	1.072423	-12.417871			0.919074	-12.240795
	5	1.115449	-12.069224			0.919270	-12.337425
	6	1.052285	-10.695112			0.929603	- 9.836704

problem are given in Appendices C and D, respectively.

Conclusions

Newton's method coupled with hybrid matrix formation of the network-defining equations has decidedly reduced the number of iterations by 41.7%, compared with the Gauss-Seidel iterative technique adopted by Hale and Goodrich (4) who solved the problem in 12 iterations for a precision index of 5×10^{-7}.

The application of Newton's method to power-flow problems with the admittance matrix representation of the network equations results in 5 iterations for convergence to solution. The hybrid matrix suffers from an inherent disadvantage of being nearly a full matrix compared to the admittance matrix which has many zeroes in the off-diagonal spaces. On that account the hybrid matrix takes greater computer memory which may become prohibitive for large-size systems. Newton's method offers quadratic convergence which makes it faster than any known method for any size of problem. Forming the hybrid matrix takes more time than the admittance matrix. Also the time per iteration and the data preparation time for the hybrid matrix method are probably greater than those for the admittance matrix method. However, it is well to remember that there are certain kinds of problems which cannot be solved by the admittance matrix method, whereas the hybrid matrix method promises a possible solution.

Whereas this investigation is carried out on a specified 6-bus problem, it is suggested that the computer program be generalized so that any number of busses could be read with an arbitrary choice of generator and load busses, respectively.

APPENDIX A

FLOW CHART FOR FORMING A HYBRID MATRIX FROM THE BUS ADMITTANCE MATRIX

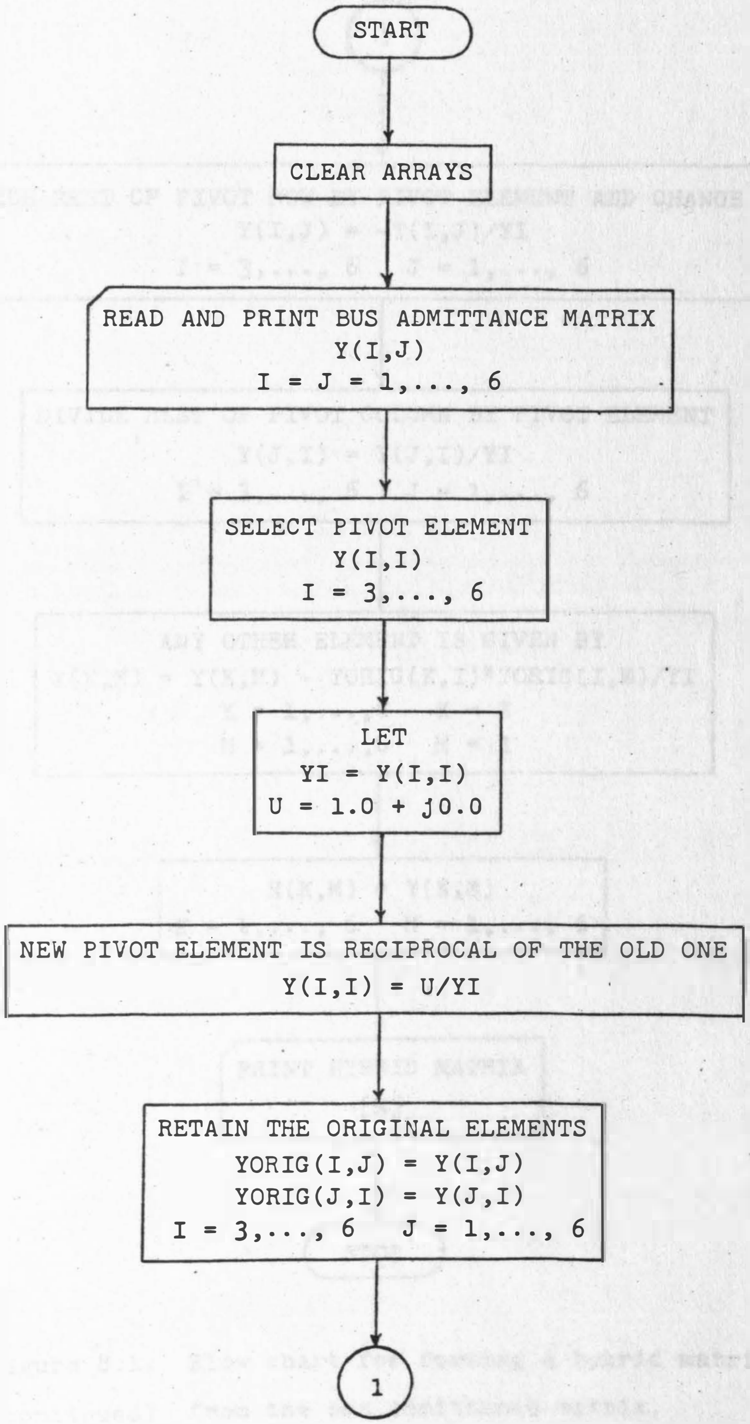

Figure 8.1. Flow chart for forming a hybrid matrix from the bus admittance matrix.

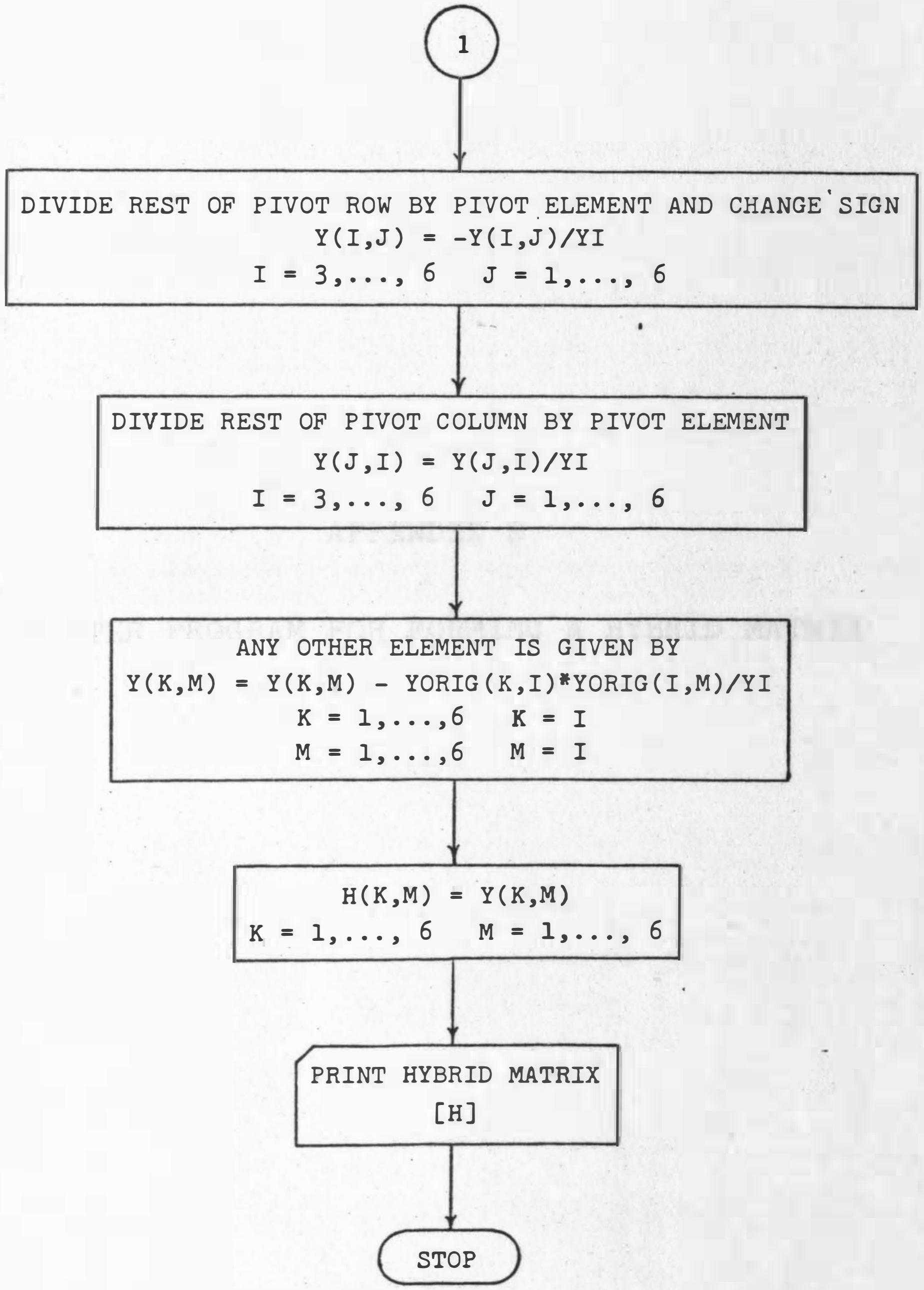

Figure 8.1. Flow chart for forming a hybrid matrix
(continued) from the bus admittance matrix.

APPENDIX B

COMPUTER PROGRAM FOR FORMING A HYBRID MATRIX

```
C       THIS PROGRAM FORMS A HYBRID MATRIX FROM THE GIVEN BUS ADMITTANCE
C       MATRIX.
C
        COMPLEX Y, H, YI, YOLD, C, PRODM, YORIG
        DIMENSION Y(10,20), H(10,20), YOLD(10,20), YORIG(10,20), C(10,20)
        DATA Y/200*(0.0,0.0)/, YOLD/200*(0.0,0.0)/, YORIG/200*(0.0,0.0)/,
       CH/200*(0.0,0.0)/, C/200*(0.0,0.0)/
        READ (11,15) ((Y(I,J), J = 1,6), I = 1,6)
     15 FORMAT (6F12.6)
C
C       WRITE ELEMENTS OF THE ADMITTANCE MATRIX
C
        WRITE (12,45)
     45 FORMAT (1H0, 19X, 18H ADMITTANCE MATRIX)
        WRITE (12,50)
     50 FORMAT (1H0, 10X, 34H ROW   COL        REAL          IMAG)
        DO 55 I = 1,6
        DO 55 J = 1,6
        WRITE (12,60) I,J, Y(I,J)
     60 FORMAT (1H0, 12X, I1, 5X, I1, 4X, F10.6, 3X, F10.6)
     55 CONTINUE
        U = CMPLX (1.0,0.0)
        DO 20 I = 3,6
        YI= Y(I,I)
        Y(I,I) = U/YI
        DO 25 J = 1,6
        IF (J. EQ. I) GO TO 25
        YORIG(I,J) = Y(I,J)
        YORIG(J,I) = Y(J,I)
        Y(I,J) = -Y(I,J)/YI
        Y(J,I) = Y(J,I)/YI
     25 CONTINUE
        DO 35 K = 1,6
        IF (K .EQ. I) GO TO 35
```

```
      DO 30 L = 1,6
      IF (L. EQ. I) GO TO 30
      Y(K,L) = Y(K,L) - YORIG(K,I)* YORIG(I,L)/YI
   30 CONTINUE
   35 CONTINUE
   20 CONTINUE
      DO 71 K = 1,6
      DO 71 L = 1,6
      H(K,L) = Y(K,L)
   71 CONTINUE
C
C     WRITE ELEMENTS OF THE HYBRID MATRIX
C
      WRITE (12,70)
   70 FORMAT (1H1, 22X, 14H HYBRID MATRIX)
      WRITE (12,50)
      DO 65 I = 1,6
      DO 65 J = 1,6
      WRITE (12,60) I, J, H(I,J)
   65 CONTINUE
      STOP
      END
```

APPENDIX C

FLOW CHART FOR POWER-FLOW SOLUTION BY NEWTON'S METHOD USING A HYBRID MATRIX

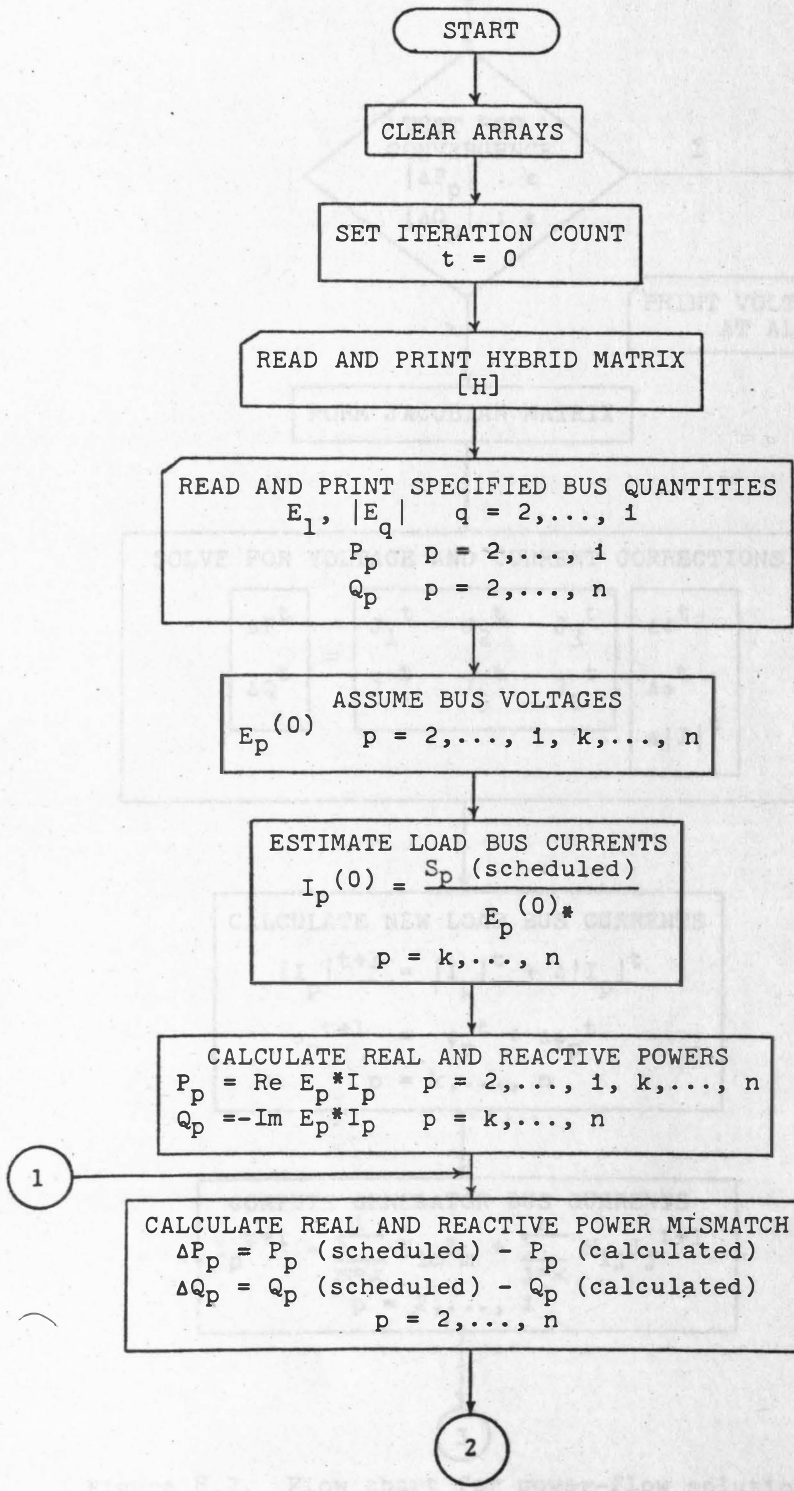

Figure 8.2. Flow chart for power-flow solution by Newton's method using hybrid matrix.

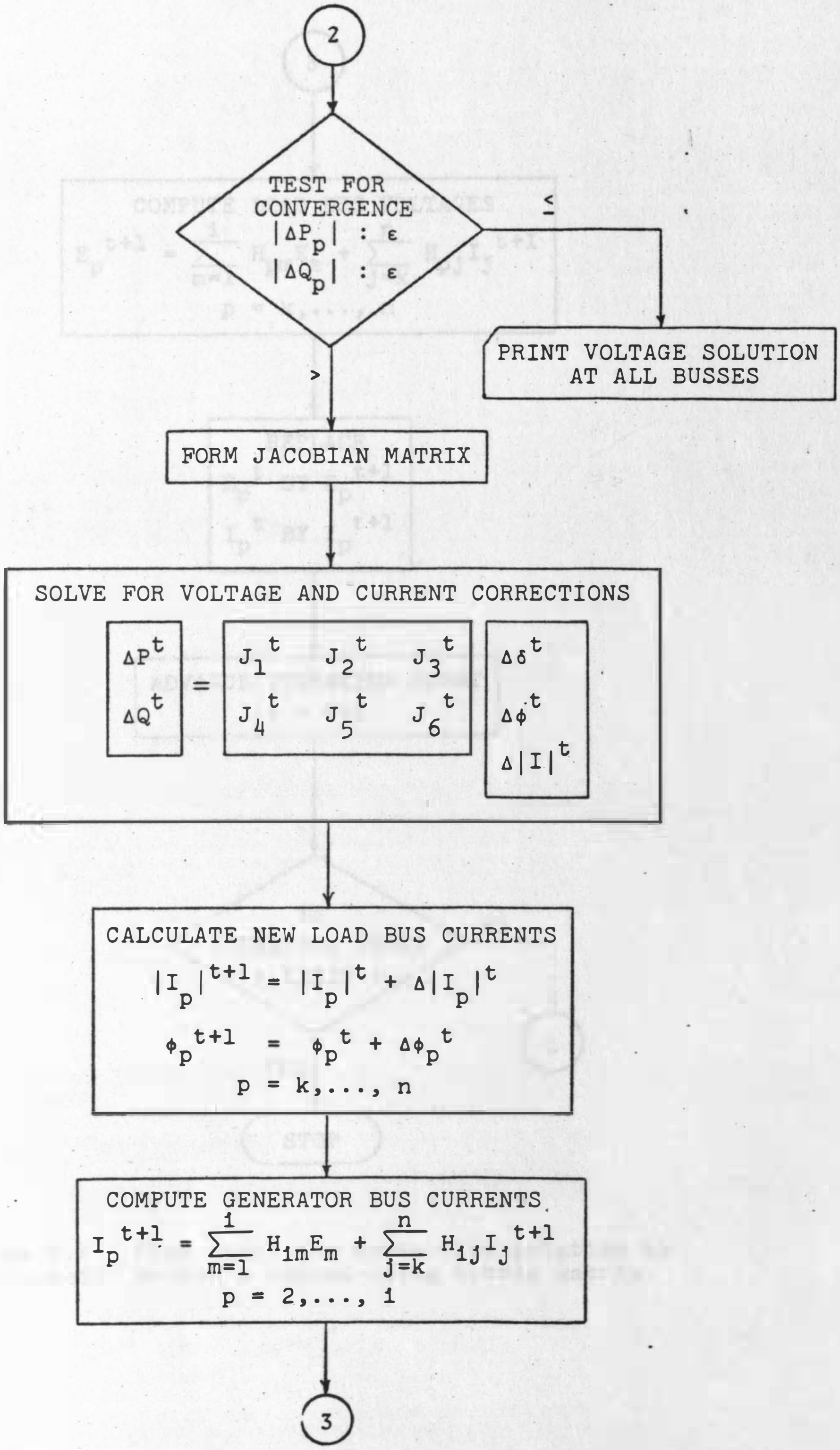

Figure 8.2. (continued) Flow chart for power-flow solution by Newton's method using hybrid matrix.

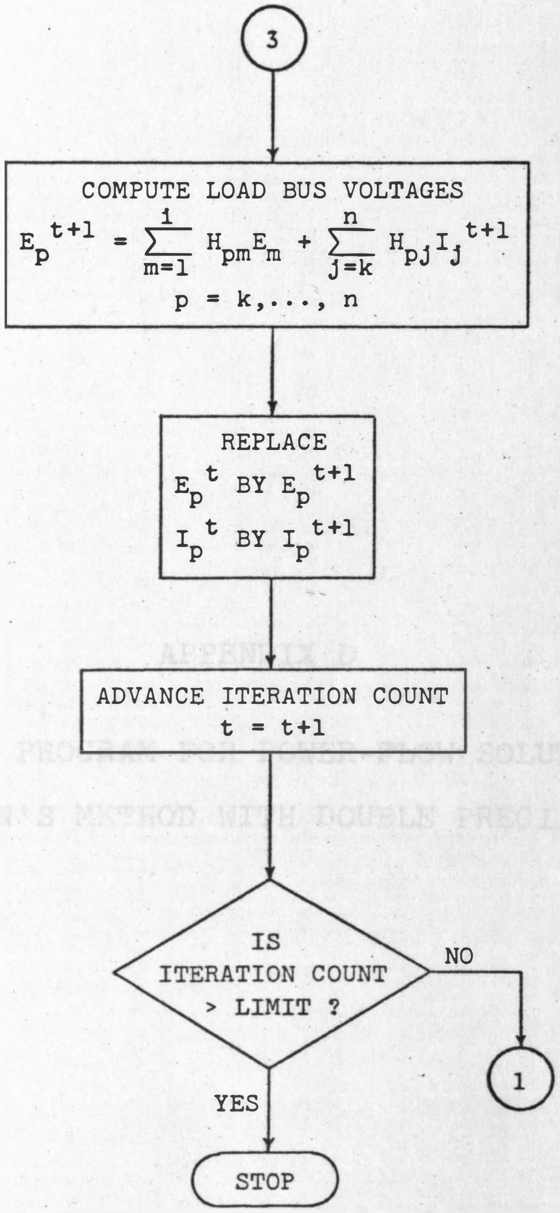

Figure 8.2. Flow chart for power-flow solution by
(continued) Newton's method using hybrid matrix.

APPENDIX D

COMPUTER PROGRAM FOR POWER-FLOW SOLUTION BY NEWTON'S METHOD WITH DOUBLE PRECISION

```
C     THIS PROGRAM SOLVES THE LOAD-FLOW PROBLEM - THE MODIFIED 6-BUS
C     WARD AND HALE SYSTEM BY NEWTON METHOD USING HYBRID MATRIX FOR THE
C     NETWORK-DEFINING EQUATIONS. POLAR COORDINATES HAVE BEEN USED.
C     GENERATOR BUSSES ARE VOLTAGE-CORRECTED AND LOAD BUSSES ARE CURRENT
C     -CORRECTED. INITIAL GUESS OF VOLTAGES IS MADE ON ALL BUSSES EXCEPT
C     THE SWING BUS. THE PROGRAM IS WRITTEN WITH DOUBLE PRECISION.
C
      REAL*8 IRL(10), IIM(10)
      COMPLEX XX, YY
      COMPLEX*16 E(20), I(20), H(10,20), S(20), HI, HE, SIGMA
      DOUBLE PRECISION AOLD(10,10), ADLAI(10), AP(10), BQ(10),
     CALPHA(10,10), BETA(10,10), DELP(10), ABSI(10), ADELTA(10),
     CDELQ(10), EP(10), EQ(10), DELTA(10), XI(10), DPDEL(10,10),
     CDPXI(10,10), DPI(10,10), DQDEL(10,10), DQXI(10,10), DQI(10,10),
     CAD(10,10), DP(10), CQ(10), A(10,10), AINV(10,10), AXI(10),
     CC(10,10), ABSE(20),
     CDLDEL(10), DLPWR(10), DLAI(10), DELXI(10), DELI(10),
     CEPS, SUM, A1, A2, A3, C1, C2, C3, D1, D2, D3, DP1, DP2, DP3, DQ1,
     CDQ2, DQ3, AB1, AB2, AB3, AC1, AD1, AE1, AF1, AF2, AF3, AF4, AF5,
     CAG1, AG2, AG3, AG4, AG5, AH1, AH2, AH3, AH4, AK1, AK2, AK3,
     CAK4, APIV, ADD, E1, E2, ERL2, EIM2
C
C      INITIALIZE THE ARRAYS
C
      DATA AXI/10*0.000/, ABSI/10*0.000/, ADELTA/10*0.0/
      DATA AOLD/100*0.000/, ABSE/20*0.000/
      DATA H/200*(0.000,0.000)/, ALPHA/100*(0.000)/, BETA/100*(0.000)/,
     CE/20*(0.000,0.000)/, I/20*(0.000,0.000)/, S/20*(0.000,0.000)/,
     CAP/10*(0.000)/, BQ/10*(0.000)/, DELP/10*(0.000)/, DELQ/10*(0.000)/
     C,EP/10*(0.000)/, EQ/10*(0.000)/, DELTA/10*(0.000)/, XI/10*(0.000)/
     C, DPDEL/10*(0.000)/, DPXI/100*(0.000)/, DPI/100*(0.000)/, DQDEL/10
     C *(0.000)/, DQXI/100*(0.000)/, DQI/100*(0.000)/
```

```
      DATA A0/100*0.0/, ADLAI/10*0.0/
      DATA C/100*0.000/, DP/10*0.0/, CQ/10*0.0/
      DATA A/100*0.000/, AINV/100*0.000/
      DATA DLDEL/10*0.000/, DLPWR/10*0.000/, DLAI/10*0.000/, DELI/10*0.0
     C00/, DELXI/10*0.000/
C
C     SET ITERATION COUNT
      NITER = 0
C
C     READ AND WRITE HYBRID MATRIX OF THE NETWORK-DEFINING EQUATIONS FOR
C     THE 6-BUS WARD AND HALE SYSTEM
C
      READ(11,72) ((H(IH,J), J = 1,6), IH = 1,6)
   72 FORMAT(6F12.6)
      WRITE (12,74)
   74 FORMAT(1H1, 22X, 14H HYBRID MATRIX)
      WRITE (12,73)
   73 FORMAT (1H0, 10X, 34H ROW     COL       REAL        IMAG)
      DO 75 IH = 1,6
      DO 75 J = 1,6
      WRITE (12,60) IH,J, H(IH,J)
   60 FORMAT(1H0, 12X, I1, 5X, I1, 4X, F10.6, 3X, F10.6)
   75 CONTINUE
      DO 65 K = 1,6
      DO 65 M = 1,6
      ALPHA(K,M) = REAL(H(K,M))
      BETA(K,M) = AIMAG(H(K,M))
   65 CONTINUE
C
C     READ AND WRITE SPECIFIED BUS QUANTITIES
      AP(2) = 0.500
      READ (11,76) (S(K), K = 3,6)
   76 FORMAT(8F8.3)
      DO 95 KK = 3,6
```

```
      AP(KK)= REAL(S(KK))
      BQ(KK) = AIMAG(S(KK))
   95 CONTINUE
      E(1) = DCMPLX(1.050D0,C.C00D0)
      E1 = CDABS(E(1))
C
C     ASSUME ANGLE OF VOLTAGE AT BUS 2 AND BOTH MAGNITUDE AND ANGLE OF
C     VOLTAGES AT BUSES 3 TC 6
C
      E(2) = DCMPLX(1.10OD0,0.000D0)
      E2 = CDABS(E(2))
      DO 85 K=3,6
      E(K) = DCMPLX(1.000D0,0.000D0)
   85 CONTINUE
      WRITE (12,15)
   15 FORMAT(1H1, 21X, 24HSPECIFIED BUS QUANTITIES)
      WRITE (12,78)
   78 FORMAT(1H0,10X,46H BUS  TYPE        P        Q       ABS(E)      E)
      WRITE (12,81) E(1)
   81 FORMAT(1H0, 12X, 1H1, 3X, 5HSWING, 29X, 2F6.3)
      WRITE (12,79) AP(2),E2
   79 FORMAT(1H0, 12X, 1H2, 3X, 6HSOURCE, 4X, F5.3, 12X, F5.3)
      DO 84 II = 3,6
      WRITE (12,83) II, S(II)
   83 FORMAT(1H0, 12X, I1, 3X, 4HLOAD, 3X, 2F8.3)
   84 CONTINUE
C
C     ESTIMATE LOAD BUS CURRENTS
C
      DO 87 KJ= 3,6
      I(KJ) = S(KJ)/DCONJG(E(KJ))
   87 CONTINUE
```

```
C      PERFORM THE FIRST ITERATION
C
   77 NITER = NITER & 1
C
C     COMPUTE CURRENT AT SOURCE BUS 2
C
      SIGMA = DCMPLX(0.000D0,0.000D0)
      DO 88 L = 3,6
      HI = H(2,L)*I(L)
      SIGMA = SIGMA + HI
   88 CONTINUE
      I(2) = H(2,1)*E(1) + H(2,2)*E(2) + SIGMA
      DO 130 K = 2,6
      ABSI(K) = CDABS(I(K))
  130 CONTINUE
  540 WRITE(12,190)
  190 FORMAT(1H1, 11X, 22HVALUES OF BUS CURRENTS)
      WRITE(12,395) NITER
  395 FORMAT(1H0, 15X, 15HITERATION NO. =,I3)
      DO 750 K = 2,6
      WRITE(12,20) K, I(K), K, ABSI(K)
   20 FORMAT(1H0, 10X, 2HI(, I1, 3H) =, 2F9.3, 10X, 5HABSI(, I1, 3H) =,
     CF6.3)
  750 CONTINUE
C
C     CALCULATE VOLTAGES AT LOAD BUSES
C
      DO9OLM=3,6
      SIGMA = DCMPLX(0.000D0,0.000D0)
      DO89M=3,6
      HE=H(LM,M)*I(M)
      SIGMA = SIGMA + HE
   89 CONTINUE
      E(LM) = H(LM,1)*E(1) + H(LM,2)*E(2) + SIGMA
```

```
   90 CONTINUE
      DO 430 K = 2,6
      ABSE(K) = CDABS(E(K))
  430 CONTINUE
      WRITE(12,425)
  425 FORMAT(1H0, 16X, 22HVALUES OF BUS VOLTAGES)
      WRITE(12,395) NITER
      DO 755 K = 2,6
      WRITE(12,400) K, E(K), K, ABSE(K)
  400 FORMAT(1H0, 15X, 2HE(, I1, 3H) =, 2F12.6, 10X, 5HABSE(, I1, 3H) =,
     CF12.6)
  755 CONTINUE
C
C     CALCULATE VOLTAGE AND CURRENT ANGLES
C
      WRITE(12,57)
   57 FORMAT(1H0, 13X, 26HVOLTAGE AND CURRENT ANGLES)
      DO 45 KW = 2,5
      YY = I(KW)
      XI(KW) = ATAN2(AIMAG(YY),REAL(YY))
      AXI(KW) = 57.29578*XI(KW)
      WRITE(12,44) KW, AXI(KW)
   44 FORMAT(1H0, 15X, 3HXI(, I1, 3H) =, F12.6)
   45 CONTINUE
      DO 58 KW = 2,6
      XX = E(KW)
      DELTA(KW) = ATAN2(AIMAG(XX),REAL(XX))
      ADELTA(KW) = 57.29578*DELTA(KW)
      WRITE(12,43) KW, ADELTA(KW)
   43 FORMAT(1H0, 15X, 6HDELTA(, I1, 3H) =, F12.6)
   58 CONTINUE
```

```
C      CALCULATE REAL AND REACTIVE POWERS
C
       WRITE(12,56)
    56 FORMAT(1H1, 10X, 35HCALCULATED REAL AND REACTIVE POWERS)
       WRITE (12,395) NITER
       DO 91 KL = 2,5
       S(KL) = DCONJG(E(KL))*I(KL)
       WRITE(12,657) KL, S(KL)
   657 FORMAT(1H0, 15X, 2HS(, I1, 3H) =, 2F8.3)
       DP(KL) = REAL(S(KL))
    91 CONTINUE
       DO 720 K = 3,5
       CQ(K) = -AIMAG(S(K))
   720 CONTINUE
C
C      CALCULATE DIFFERENCES BETWEEN SCHEDULED AND CALCULATED POWERS
C
       DO 92 K = 2,5
       DELP(K) = AP(K) - DP(K)
       EP(K)=DEL P(K)
       DELQ(K) = BQ(K) - CQ(K)
       EQ(K)= DEL Q(K)
    92 CONTINUE
       WRITE(12,656)
   656 FORMAT(1H0, 10X, 33HREAL AND REACTIVE POWER RESIDUALS)
       DO 55 K = 2,5
       WRITE(12,54) K, EP(K), K, EQ(K)
    54 FORMAT(1H0, 15X, 3HEP(, I1, 3H) =, F10.6, 12X, 3HEQ(, I1, 3H) =,
      CF10.6)
    55 CONTINUE
C
C      SET PRECISION INDEX - EPS
C
       EPS = 5.D-7
```

```
C     TEST FOR CONVERGENCE
C
      DO 50 J = 3,5
      IF(DABS(EP(2)) .GT. EPS) GO TO 30
      IF(DABS(EP(J)) .GT. EPS) GO TO 30
      IF(DABS(EQ(J)) .GT. EPS) GO TO 30
   50 CONTINUE
      GO TO 94
C
C     CALCULATE ELEMENTS OF JACOBIAN
C
C     DPDEL(K,M) = DP(K)/DDELTA(M) = PARTIAL DERIVATIVE OF P(K) WRT DELT
C     A(M)
C     DPXI(K,M) = DP(K)/DXI(M) = PARTIAL DERIVATIVE OF P(K) WRT XI(M)
C     DPI(K,M) = DP(K)/DI(M) = PARTIAL DERIVATIVE OF P(K) WRT I(M)
C     DQDEL(K,M) = DQ(K)/DDEL(M) = PARTIAL DERIVATIVE OF Q(K) WRT DELTA
C     (M)
C     DQXI(K,M) = DQ(K)/DXI(M) = PARTIAL DERIVATIVE OF Q(K) WRT XI(M)
C     DQI(K,M) = DQ(K)/ DI(M) = PARTIAL DERIVATIVE OF Q(K) WRT I(M)
C
   30 SUM = 0.0D0
      DO 100 M = 3,5
      A1 = -ALPHA(2,M)*DSIN(DELTA(2)-XI(M))
      A2 = BETA(2,M)*DCOS(DELTA(2)-XI(M))
      A3 = CDABS(I(M))*(A1 & A2)
      SUM = SUM & A3
  100 CONTINUE
C
      DPDEL(2,2) = E1*E2*(-ALPHA(2,1)*DSIN(DELTA(2)) + BETA(2,1)*
     CDCOS(DELTA(2))) + E2*SUM
C
      DO 102 NP = 3,5
      C1 = ALPHA(2,NP)*DSIN(DELTA(2)-XI(NP))
      C2 = BETA(2,NP)*DCOS(DELTA(2)-XI(NP))
```

```
      C3 = C1-C2
C
      DPXI(2,NP) = E2*CDABS(I(NP))*C3
C
      D1 = ALPHA(2,NP)*DCOS(DELTA(2)-XI(NP))
      D2 = BETA(2,NP)*DSIN(DELTA(2)-XI(NP))
      D3 = D1&D2
C
      DPI(2,NP) = E2*D3
C
  102 CONTINUE
      DO 105 KP = 3,5
      DP1 = ALPHA(KP,2)*DSIN(DELTA(2)-XI(KP))
      DP2 = BETA(KP,2)*DCOS(DELTA(2)-XI(KP))
      DP3=DP1&DP2
C
      DPDEL(KP,2) = -E2*CDABS(I(KP))*DP3
C
      DQ1 = ALPHA(KP,2)*DCOS(DELTA(2)-XI(KP))
      DQ2 = BETA(KP,2)*DSIN(DELTA(2)-XI(KP))
      DQ3 = DQ1-DQ2
C
      DQDEL(KP,2) = E2*CDABS(I(KP))*DQ3
C
  105 CONTINUE
      DO 106 KR = 3,5
      DO 107 KS = 3,5
      IF(KS .EQ. KR) GO TO 107
      AB1 = ALPHA(KR,KS)*DSIN(XI(KR)-XI(KS))
      AB2 = BETA(KR,KS)*DCOS(XI(KR)-XI(KS))
      AB3 = AB1-AB2
C
      DPXI(KR,KS) = CDABS(I(KR))*CDABS(I(KS))*AB3
```

```
      AC1 = ALPHA(KR,KS)*DCOS(XI(KR)-XI(KS)) & BETA(KR,KS)*DSIN
     C(XI(KR)-XI(KS))
C
      DQXI(KR,KS) = CDABS(I(KR))*CDABS(I(KS))*AC1
C
      AD1 = ALPHA(KR,KS)*DCOS(XI(KR)-XI(KS)) & BETA(KR,KS)*DSIN(XI(KR)
     C- XI(KS))
C
      DPI(KR,KS) = CDABS(I(KR))*AD1
C
      AE1 = -ALPHA(KR,KS)*DSIN(XI(KR)-XI(KS)) & BETA(KR,KS)*DCOS(XI(KR)
     C-XI(KS))
C
      DQI(KR,KS) = CDABS(I(KR))*AE1
C
  107 CONTINUE
  106 CONTINUE
      DO 108 KA = 3,5
      AF1 = ALPHA(KA,2)*DSIN(DELTA(2)-XI(KA)) & BETA(KA,2)*DCOS(DELTA(2)
     C-XI(KA))
      AF2 = E2*CDABS(I(KA))*AF1
      SUM = 0.0D0
      DO 109 M = 3,5
      IF (M .EQ. KA) GO TO 109
      AF3 = -ALPHA(KA,M)*DSIN(XI(KA)-XI(M)) & BETA(KA,M)*DCOS(XI(KA)-
     CXI(M))
      AF4 = CDABS(I(M))*AF3
      SUM = SUM & AF4
  109 CONTINUE
      AF5 = CDABS(I(KA))*SUM
C
      DPXI(KA,KA) = AF2 & AF5 & E1*CDABS(I(KA))*(-ALPHA(KA,1)*
     CDSIN(XI(KA)) & BETA(KA,1)*DCOS(XI(KA)))
```

```
      AG1 = -ALPHA(KA,2)*DCOS(DELTA(2)-XI(KA)) & BETA(KA,2)*
     CDSIN(DELTA(2)-XI(KA))
      AG2 = E2*CDABS(I(KA))*AG1
      SUM = 0.0D0
      DO 110 M = 3,5
      IF(M .EQ. KA) GO TO 110
      AG3 = ALPHA(KA,M)*DCOS(XI(KA)-XI(M)) & BETA(KA,M)*DSIN(XI(KA)-
     CXI(M))
      AG4 = CDABS(I(M))*AG3
      SUM=SUM&AG4
  110 CONTINUE
      AG5 = CDABS(I(KA))*SUM
C
      DQXI(KA,KA) = AG2 - AG5 - E1*CDABS(I(KA))* (ALPHA(KA,1)*
     CDCOS(XI(KA)) & BETA(KA,1)*DSIN(XI(KA)))
C
      AH1 = ALPHA(KA,2)*DCOS(DELTA(2)-XI(KA)) - BETA(KA,2)*DSIN(DELTA(2)
     C-XI(KA))
      AH2 = E2*AH1
      SUM = 0.0D0
      DO 111 M = 3,5
      IF(M .EQ. KA) GO TO 111
      AH3 = ALPHA(KA,M)*DCOS(XI(KA)-XI(M))+ BETA(KA,M)*DSIN(XI(KA)
     C-XI(M))
      AH4 = CDABS(I(M))*AH3
      SUM=SUM&AH4
  111 CONTINUE
C
      DPI(KA,KA) = 2.0*ALPHA(KA,KA)*CDABS(I(KA)) + AH2 + SUM +
     CE1*(ALPHA(KA,1)*DCOS(XI(KA)) + BETA(KA,1)*DSIN(XI(KA)))
C
      AK1 = ALPHA(KA,2)*DSIN(DELTA(2)-XI(KA)) + BETA(KA,2)*DCOS(DELTA(2)
     C-XI(KA))
      AK2 = E2*AK1
```

```
      SUM = 0.CDO
      DO 112 M = 3,5
      IF(M .EQ.KA) GO TO 112
      AK3 = -ALPHA(KA,M)*DSIN(XI(KA)-XI(M) ) + BETA(KA,M)*DCOS(XI(KA)-
     CXI(M))
      AK4 = CDABS(I(M))*AK3
      SUM= SUM&AK4
  112 CONTINUE
C
      DQI(KA,KA) = 2.0*BETA(KA,KA)*CDABS(I(KA)) + AK2 + SUM +
     CE1*(-ALPHA(KA,1)*DSIN(XI(KA)) + BETA(KA,1)*DCOS(XI(KA)))
C
  108 CONTINUE
C
C     WRITE ELEMENTS OF JACOBIAN
C
      WRITE(12,200)
  200 FORMAT(1H1, 13X, 31HELEMENTS OF THE JACOBIAN MATRIX)
      WRITE(12,395) NITER
      DO 135 JA = 2,5
      WRITE(12,140) JA, DPDEL(JA,2)
  140 FORMAT(1H0, 15X, 3HDP(, I1, 11H)/DDEL(2) =, F12.6)
  135 CONTINUE
      DO 145 JB = 3,5
      WRITE (12,150) JB, DQDEL(JB,2)
  150 FORMAT(1H0, 15X, 3HDQ(, I1, 11H)/DDEL(2) =, F12.6)
  145 CONTINUE
      DO 155 JC = 2,5
      DO 160 JD = 3,5
      WRITE(12,165) JC, JD, DPXI(JC,JD), JC, JD, DPI(JC,JD)
  165 FORMAT(1H0, 15X, 3HDP(, I1, 6H)/DXI(, I1, 3H) =, F12.6, 15X, 3HDP(
     C, I1, 5H)/DI(, I1, 4H) =, F12.6)
  160 CONTINUE
  155 CONTINUE
```

```
      DO 170 JE = 3,5
      DO 175 JF = 3,5
      WRITE(12,180) JE, JF, DQXI(JE,JF), JE, JF, DQI(JE,JF)
  180 FORMAT(1H0, 15X, 3HDQ(, I1, 6H)/DXI(, I1, 3H) =, F12.6, 15X,
     C3HDQ(, I1, 5H)/DI(, I1, 4H)  =, F12.6)
  175 CONTINUE
  170 CONTINUE
C
C     FORM EQUIVALENT JACOBIAN MATRIX
C
      DO 205 K = 2,5
  205 A(K-1,1) = DPDEL(K,2)
      DO 210 M = 3,5
  210 A(M+2,1) = DQDEL(M,2)
      DO 220 L = 2,5
      DO 215 N = 3,5
      A(L-1,N-1) = DPXI(L,N)
  215 CONTINUE
  220 CONTINUE
      DO 230 J = 3,5
      DO 225 JA = 3,5
      A(J+2,JA-1) = DQXI(J,JA)
  225 CONTINUE
  230 CONTINUE
      DO 240 LA = 2,5
      DO 235 LB = 3,5
      A(LA-1,LB+2) = DPI(LA,LB)
  235 CONTINUE
  240 CONTINUE
      DO 250 MA = 3,5
      DO 245 MB = 3,5
      A(MA+2,MB+2) = DQI(MA,MB)
  245 CONTINUE
  250 CONTINUE
```

```
C     WRITE THE JACOBIAN MATRIX IN THE EQUIVALENT FORM
C     A(K,M) = A(K,M) - AO(K,L)*AO(L,M)/APIV
      WRITE(12,254)
  254 FORMAT(1H1, 10X, 49HELEMENTS OF THE JACOBIAN MATRIX (EQUIVALENT FO
     CRM))
      WRITE(12,395) NITER
      WRITE(12,255)
  255 FORMAT(1H0, 15X, 24H ROW   COL        A(K,M),/,17X,'(K)', 3X,
     C'(M)')
      DO 270 K = 1,7
      DO 265 M = 1,7
      WRITE(12,260) K,M, A(K,M)
  260 FORMAT(1H0, 17X, I1, 5X, I1, 4X, F12.6)
  265 CONTINUE
  270 CONTINUE
      DO 590 K = 1,7
      DO 590 M = 1,7
  590 AOLD(K,M) = A(K,M)
C
C     INVERT THE JACOBIAN MATRIX
C
      DO 275 L = 1,7
      APIV = A(L,L)
      A(L,L) = 1.0/APIV
      DO 280 N = 1,7
      IF(N .EQ. L) GO TO 280
      AO(L,N) = A(L,N)
      AO(N,L) = A(N,L)
      A(L,N) = -A(L,N)/APIV
      A(N,L) = A(N,L)/APIV
  280 CONTINUE
      DO 285 K = 1,7
      IF(K .EQ. L) GO TO 285
      DO 284 M = 1,7
```

```
      IF (M .EQ. L) GO TO 284
      A(K,M) = A(K,M) - AC(K,L)*AC(L,M)/APIV
  284 CONTINUE
  285 CONTINUE
  275 CONTINUE
      DO 600 K = 1,7
      DO 600 M = 1,7
  600 AINV(K,M) = A(K,M)
C
C     WRITE ELEMENTS OF THE INVERTED JACCBIAN MATRIX
C
      WRITE(12,289)
  289 FORMAT(1H1, 10X, 40HELEMENTS OF THE INVERTED JACOBIAN MATRIX)
      WRITE(12,395) NITER
      WRITE(12,290)
  290 FORMAT(1H0, 15X, 25H ROW   COL      AINV(K,M),/,17X,'(K)', 3X,
     C'(M)')
      DO 305 K = 1,7
      DO 300 M = 1,7
      WRITE(12,295) K, M, AINV(K,M)
  295 FORMAT(1H0, 17X, I1, 5X, I1, 4X, F12.6)
  300 CONTINUE
  305 CONTINUE
C
C     (DLAI) REPRESENTS COLUMN VECTOR OF CHANGES IN VOLTAGE ANGLE AT BUS
C     2, CURRENT ANGLES AT BUSES 3 TO 5, AND CURRENTS AT BUSES 3 TO 5
C     (DLPWR) REPRESENTS COLUMN VECTOR OF CHANGES IN REAL POWER AT BUSES
C     2 TO 5, AND IN REACTIVE POWER AT BUSES 3 TO 5
C
C     WRITE VOLTAGE, CURRENT, AND POWER ERRORS IN THE EQUIVALENT FORM
C
  700 DLAI(1) = DLDEL(2)
      DO 115 K = 3,5
      DLAI(K-1) = DELXI(K)
```

```
      DLAI(K&2) = DELI(K)
  115 CONTINUE
      DO 116 L = 2,5
  116 DLPWR(L-1) = EP(L)
      DO 117 N = 3,5
  117 DLPWR(N&2) = EQ(N)
C
C     SOLVE FOR VOLTAGE AND CURRENT CORRECTIONS
C
      DO 119 J = 1,7
      SUM = 0.0D0
      DO 118 M = 1,7
      ADD = AINV(J,M)*DLPWR(M)
      SUM = SUM & ADD
  118 CONTINUE
      DLAI(J) = SUM
  119 CONTINUE
      DO 621 K = 1,4
      ADLAI(K) = 57.29578*DLAI(K)
  621 CONTINUE
C
C     WRITE THE ERRORS IN VOLTAGE AND CURRENT ANGLES AND IN CURRENTS
C
      WRITE(12,120)
  120 FORMAT(1H1,  8X, 36HERRORS IN VOLTAGE AND CURRENT ANGLES)
      WRITE(12,395) NITER
      WRITE(12,725)
  725 FORMAT(1H0, 20X, 9H(RADIANS))
      WRITE(12,315) DLAI(1)
  315 FORMAT(1H0, 15X, 10HDLDEL(2) =, F12.6)
      DO 122 K = 2,4
      L = K&1
      WRITE(12,320) L, DLAI(K)
  320 FORMAT(1H0, 15X, 6HDELXI(, I1, 3H) =, F12.6)
```

```
  122 CONTINUE
      WRITE(12,59)
   59 FORMAT(1H0, 20X, 9H(DEGREES))
      WRITE(12,622) ADLAI(1)
  622 FORMAT(1H0, 15X, 11HADEL(2) =, F12.6)
      DO 624 K = 2,4
      L = K+1
      WRITE(12,623) L, ADLAI(K)
  623 FORMAT(1H0, 15X, 7HADELXI(, I1, 3H) =, F12.6)
  624 CONTINUE
      WRITE(12,740)
  740 FORMAT(1H0, 17X, 18HERRORS IN CURRENTS)
      DO 330 K = 5,7
      L = K-2
      WRITE(12,325) L, DLAI(K)
  325 FORMAT(1H0, 15X, 5HDELI(, I1, 4H)  =, F12.6)
  330 CONTINUE
C
C     COMPUTE NEW VOLTAGE AND CURRENT ANGLES
C
      DELTA(2) = DELTA(2) & DLAI(1)
      ADELTA(2) = DELTA(2)*57.29578
      DO 123 KL = 3,5
      XI(KL) = XI(KL) & DLAI(KL-1)
      AXI(KL) = XI(KL)*57.29578
  123 CONTINUE
      WRITE(12,510)
  510 FORMAT(1H0, 11X, 30HNEW VOLTAGE AND CURRENT ANGLES,/, 22X,
     C9H(DEGREES))
      WRITE(12,335) ADELTA(2)
  335 FORMAT(1H0, 15X, 10HDELTA(2) =, F12.6)
      DO 545 KL = 3,5
      WRITE(12,340) KL, AXI(KL)
  340 FORMAT(1H0, 15X, 3HXI(, I1, 6H)    =, F12.6)
```

```
C     CALCULATE NEW LOAD BUS VOLTAGES AND CURRENTS
C
      ABSI(KL) = ABSI(KL) & DLAI(KL&2)
      IRL(KL) = ABSI(KL)*DCOS(XI(KL))
      IIM(KL) = ABSI(KL)*DSIN(XI(KL))
      I(KL) = DCMPLX(IRL(KL),IIM(KL))
  545 CONTINUE
      ERL2=E2*DCOS(DELTA(2))
      EIM2 = E2*DSIN(DELTA(2))
      E(2) = DCMPLX(ERL2,EIM2)
      E2 = CDABS(E(2))
C
C     IF ITERATION COUNTER EXCEEDS MAXIMUM ALLOWABLE, GIVE UP
C
      IF(NITER .GT. 12) GO TO 410
C
C     ADVANCE ITERATION COUNT
C
      GO TO 77
  410 WRITE(12,415)
  415 FORMAT(1H1, 15X, 35HPROBLEM NOT CONVERGED TO A SOLUTION)
      GO TO 515
   94 WRITE(12,455)
  455 FORMAT(1H0, 15X, 14HPROBLEM SOLVED)
  515 STOP
      END
```

BIBLIOGRAPHY

1. Ward, J. B. and Hale, H. W., "Digital Computer Solution of Power- Flow Problems," AIEE Transactions, vol. 75, Part 3, pp. 398-404, 1956.

2. Brown, R. J. and Tinney, W. F., "Digital Solutions for Large Power Networks," AIEE Transactions, vol. 76, Part 3, pp. 347-351,1957.

3. Van Ness, James E., "Iteration Methods for Digital Load Flow Studies," AIEE Transactions, vol. 78, Part 3, pp. 583-588, 1959.

4. Hale, H. W. and Goodrich, R. W., "Digital Computation of Power Flow - Some New Aspects," AIEE Transactions, vol. 78, Part 3, pp. 919-924, 1959.

5. Shipley, Bruce R. and Coleman, Dorothy, "A New Direct Matrix Inversion Method," AIEE Transactions, vol. 78, Part 1, pp. 568-572, 1959.

6. Van Ness, James E. and Griffin, John H., "Elimination Methods for Load Flow Studies," AIEE Transactions, vol. 80, Part 3, pp. 299-303, 1961.

7. Storry, Junis O., Digital Analysis of Power Systems using Hybrid Parameters, Ph.D. Dissertation, Iowa State University of Science and Technology, Ames, Iowa, 1967.

8. Tinney, William F. and Hart, Clifford E., "Power Flow Solution by Newton's Method," IEEE Transactions, vol. 86, Part 3, pp. 1449-1460, 1967.

9. Stagg, Glenn W., and Ahmed H. El-Abiad, Computer Methods in Power System Analysis, McGraw-Hill Book Company, New York, 1968.

10. Stevenson, William D., Jr., Elements of Power System Analysis, McGraw-Hill Book Company, New York, 1962.

11. Neuenswander, John R., Modern Power Systems, International Textbook Company, Scranton, 1971.

12. Stiefel, Eduard L., An Introduction To Numerical Mathematics, Academic Press Inc., New York, 1966.