OPTIMAL POWER FLOW STUDIES

THESIS
SUBMITTED IN PARTIAL FULFILMENT OF
THE REQUIRMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

(ELCTRICAL ENGINEERING)

BY

S. K. Agarwal

UNDER THE SUPERVISION OF

Prof. I. J. Nagrath

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING
BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
PILANI (RAJASTHAN)

1970

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE

PILANI (RAJASTHAN)

SUPERVISOR'S CERTIFICATE

Certified that the research work described in the thesis bearing the title "Optimal Power Flow Studies' was carried out by Mr. Santosh Kumar Agarwal under my supervision.

(I.J. Nagrath)

Professor of Electrical Engineering,
Department of Electrical & Electronics E
Birla Institute of Technology & Science,
PILANI, (Raj.) India.

ACKNOWLEDGEMENTS

The author expresses his deep sense of gratitude to Professor I.J. Nagrath, his thesis supervisor, without whose constant guidance, supervision and encouragement the work would not have been possible.

The author is very thankful to Professor A.H. El-Abiad, Professor of Electrical Engineering, Purdue University for his help and encouragement during the initial stages of the work. He feels indebted to Professor David C. White, Professor of Electrical Engineering, Massachusetts Institute of Technology and formerly Visiting Professor at B.I.T.S. for his encouragement and useful suggestions.

The author thanks Professor M.M. Mukerji and Dr. H.C. Misra of Information Processing Centre for making the facilities of the Centre available to the author.

The author thanks Dr. W.F. Tinney and Dr. H.W. Dommel of BPA for certain useful suggestions and encouragement. Thanks are also due to Dr. G.W. Stagg formerly of American Electric Power for supplying the system test data and to Dr. John Peschon, formerly of Stanford Research Institute for making some of the SRI reports[36,39] available to the author.

The author thanks the authorities of B.I.T.S. for the facilities provided for the work.

SYNOPSIS

In recent years a great deal of work has been done on the optimal power flow studies making use of nonlinear programming formulation with full a.c. simulation of the network. Important advances made include application of various optimization techniques in conjunction with efficient computation methods like triangular factorization, compact storage and optimal renumbering. These methods recently introduced to power system engineers by W.F. Tinney and his associates of Bonneville Power Administration[1,4] have made possible the study of very large power systems without partitioning and resulted in considerable saving of computer time. Chapter II discusses mainly these.

In Chapter III algorithm of forming the admittance matrix and simultaneously storing the non zero elements in the compact form is given. Power flow equations are discussed in Chapter IV.

In Chapter V various available load flow techniques are lumped into general categories and compared through fixed point analysis. To date not much use has been made of such analysis to power system studies. The author is familiar with only one paper[3] making its use. This paper mainly discusses Newton's method through Kantrovich's theorem which makes use of Jacobian and the three dimensional Hessian matrix, computation of which is very hard. By making use of fixed point analysis, convergence properties of various algorithms which up to now were experimentally established are analytically established.

A discussion of computer programme for load flow solution by Newton's method is given in Chapter VI. In Chapter VII criteria and techniques for identifying the optimal point of the nonlinear programming problem and convergence of sequential optimization through penalty and barrier methods are discussed.

In Chapter VIII various published techniques for optimal flow solutions are discussed and some are studied through fixed point analysis, which indicate the doubtful convergence of many of these. Dommel and Tinney's[21] technique is found to be the most promising. In Chapter IX the computer programme for optimal power flow problem making use of a method which is a modification of the above is presented. It uses a modified Lagrangian, a different gradient and penalty method and handles the inequality constraints on voltages for the buses with reactive power generation on different line. In Chapter X algorithms developed by the author for optimal hydrothermal operations are presented.

++++++++++++

CONTENTS

Page

>>>>>>>>>>>>>>>

NOTATIONS

Following is the list of more commonly used symbols. In rare cases, some of these symbols represent a different quantity, explained in the text.

$V_i \angle \delta_i = e_i + j f_i$	=	complex voltage of node i
I_i	=	nodal current
$y_{ij} \angle \psi_{ij} = g_{ij} + j b_{ij}$	=	element of admittance matrix
V	=	column vector of bus voltages
I	=	column vector of bus currents
Y	=	admittance matrix
$P_i + j Q_i$	=	net injected power at node i
$P_i^G + j Q_i^G$	=	net generation at node i
$P_i^L + j Q_i^L$	=	net load at node i
$\overline{x}$	=	upper limit of x
$\underline{x}$	=	lower limit of x
M, A	=	Jacobian matrix
n	=	number of nodes
r	=	step size
R^k	=	constant with the sequential minimization
t_h	=	off nominal turn ratio of hth transformer
ϕ_h	=	angular shift of hth phase shifting transformer
x^*	=	the required value of x
P_L	=	system transmission losses
P_D	=	system load demand

subscript	s	=	refers to the slack bus
	G	=	set of nodes with active power generations
	R	=	set of nodes with reactive power generations
	T	=	set of nodes with thermal generations
	H	=	set of nodes with hydrogenerations.
	$S_1 + S_2$	=	union of sets S_1 and S_2
	$S_1 * S_2$	=	intersection of sets S_1 and S_2

Chapter I

INTRODUCTION

Optimal power flow studies were made as early as 1940's when the transmission losses were neglected and the generation powers were scheduled on the basis of equal incremental costs. With the development of interconnected power systems, need for consideration of these losses was felt. These losses initially were approximated by a quadratic loss formula. Further work was aimed on the improvement in the method of obtaining the loss formula and generator scheduling initially on network analyser and later on digital computer. This method optimises for real powers only. Limitations of the loss formula led to search for better methods. A partial break through was first achieved in 1962 by Carpentier of Electricite' de France who made use of non-linear programming formulation through Kuhn Tucker's theorem with full a.c. representation of the network, however the algorithm needed further improvements. Further work in this direction published in English speaking literature was carried out at Bonneville Power Administration and Stanford Research Institute for the BPA system which has 100% hydrogeneration, for which transmission losses are to be minimised by reactive powers and tap adjustments. Carpentier's algorithm was not found suitable for the system. BPA had meanwhile perfected the computer programme for load flow studies and published in 1967. Making use of this programme a method for reactive power optimization was suggested[19,36]. A lot of work meanwhile in this direction was also reportedly carried out in U.S.S.R. most of which is not yet available in English literature. Thereafter a number of papers in this direction have appeared, most claiming some advantages over other methods. An entirely different approach making use of penalty functions was developed by Sasson at Imperial College, London, and Instituto Technologico y de Estudios Superiores de Monterrey, Mexico, and published in 1969-1970[4,15,16].

The author's aim was to make a comparative study of the existing published methods and to develop an efficient and reliable computer programme suitable for large power systems. This needs that the sparsity of system matrices be fully exploited. Application of matrix inverse is out of question for larger systems. Fortunately the technique of triangular factorization recently introduced to power system engineers replaces the use of matrix inverse. This along with compact storage for matrices and optimal renumbering of nodes makes the computer logic complicated.

Performance of any technique is better judged from a bigger system. Because of availability of small digital computer IBM 1130 with 16 K, 16 bit storage, the study was restricted to 30 buses only. Findings of Dommel and Tinney indicate that conclusions drawn on a system of 25 buses or so usually hold good for larger system as well[14].

After some experimentation over the existing published techniques for load flow solutions, Newton's method as perfected by Tinney and Hart[33] with some modifications, was finally adopted. Chapter VI, discussing the author's programme for load flow studies is intended as a supplement to this reference. The accuracy of this method was found restricted only by round off errors. Though a tolerance of 10^{-4} pu was prescribed, mismatch of most buses was found much less than this at the end of fourth iteration. A need for very accurate load flow programme was felt for the optimal power flow studies in the absence of which the relative costs with different schedule do not give a realistic comparison since the improvement in the operating cost is expected between 1 to 3% only with optimal scheduling. Improvement in transmission loss reported by Peschon et al. is about 4% with reactive power scheduling[35]. A comparison of various published algorithms for optimal power flow studies in Chapter VIII indicates the supremacy of Dommel and Tinney's method over others. While trying this method for the system described in Section 9.8, a need for further improvement was felt of which the details are given in Chapter IX. Application of the optimal flow programme apart from the routine fuel cost minimization was also made for planning Mvar requirements of power systems under constraints on line voltages in Section 9.10.

In Chapter X algorithms based on Lagrangian formulation and gradient techniques for optimal hydrothermal scheduling of system consisting of multireservoirs (with and without head variation) and multithermal plants are given. Initially the algorithm is based on Bryson and Denham's technique[48] for which the solution remains in the feasible domain of power flow equations and water storage. Application of decomposition techniques, comparative advantages and disadvantages are also given. It was found that application of maximum principle and Lagrangian formulation essentially give the same mathematical equations, on which the algorithm is to be based.

++++++++++++++

CHAPTER II

SOLUTION OF LINEAR EQUATIONS

This chapter presents some of the recently introduced techniques namely triangular factorization, compact storage scheme and optimal renumbering scheme, for solving a set of linear equations. Section 2.1 presents triangular factoriza-tion technique, in which an array of numbers, is obtained from a non singular matrix A; that can be used to obtain the effect of A, A^{-1}, A^{t}, $(A^{-1})^{t}$ over a column vector. This technique replaces the use of matrix inverse, which is very inefficient for large sparse system of equations such as occur in power network problems, since while the system matrix is sparse, its inverse is a full matrix. Most material of this section is from reference (31). However in addition a few computer logics and few additional proofs which include 2.42, 2.51 and 2.54 have been added. Few of the compact storage schemes are des-cribed in section 2.2. Near optimal renumbering schemes are discussed in section 2.3. Description of various numbering schemes is from reference (1). In addition description and computer logic of authors sub-routine is provided. These two techniques greatly improve the efficiency of traingular factorization.

2.1 Triangular Factorization

A matrix is usually triangularized by eliminating elements of successive columns below the diagonal. From the point of computer speed and storage, it is more efficient to eliminate elements of each row up to the diagonal before proceeding to the next row, since only one row need be formed and operated upon at one time.

Further discussion is based on the following equation

$$Ax = b \tag{2.1}$$

where A is n x n non singular matrix, x is a column vector of unknowns and b is the known vector.

This equation written explicitly is as follows -

$$\begin{bmatrix} a_{11} & a_{21} \cdots & a_{1n} \\ a_{21} & a_{22} & a_{2n} \\ \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \tag{2.2}$$

The first step in Gaussian elimination is to divide the elements of the first row by a_{11} as follows:

$$d_{11} = 1/a_{11} \tag{2.3}$$

$$a_{1j}^{(1)} = d_{11}\, a_{1j}; \quad j = 1, \ldots, n \tag{2.4}$$

$$u_{1j} = a_{1j}^{(1)}; \quad j = 2, \ldots, n \tag{2.5}$$

$$b_1^{(1)} = d_{11}\, b_1 \qquad (2.6)$$

a_{21} is eliminated from the second row by linear combination with the derived first row, and then to divide the remaining derived elements of the second row by its derived diagonal element as follows:

$$l_{21} = a_{21} \qquad (2.7)$$

$$a_{2j}^{(1)} = a_{2j} - l_{21}\, a_{1j}^{(1)} \; ; \quad j = 1, \ldots, n \qquad (2.8)$$

$$b_2^{(1)} = b_2 - l_{21}\, b_1^{(1)} \qquad (2.9)$$

$$d_{22} = 1/a_{22}^{(1)} \qquad (2.10)$$

$$a_{2j}^{(2)} = d_{22}\, a_{2j}^{(1)} ; \; j = 2, \ldots, n \qquad (2.11)$$

$$u_{2j} = a_{2j}^{(2)} \; ; \quad j = 3, \ldots, n \qquad (2.12)$$

$$b_2^{(2)} = d_{22}\, b_2^{(1)} \qquad (2.13)$$

a_{31} and a_{32} are eliminated from third row and remaining elements of this row are now divided by the diagonal element.

Thus elements to the left of diagonal are eliminated and the diagonal element is made unity. This process illustrated for kth row is as follows.

$$l_{k1} = a_{k1} \qquad (2.14)$$

$$a_{kj}^{(1)} = a_{kj} - l_{k1}\, a_{1j}^{(1)} ; \; j = 1, \ldots, n \qquad (2.15)$$

$$b_k^{(1)} = b_k - l_{k1}\, b_1^{(1)} \tag{2.16}$$

$$l_{k2} = a_{k2}^{(1)} \tag{2.17}$$

$$a_{k2}^{(2)} = a_{kj}^{(1)} - l_{k2}\, a_{2j}^{(2)} \quad ; \quad j = 2,\ldots,n \tag{2.18}$$

$$b_k^{(2)} = b_k^{(1)} - l_{k2}\, b_2^{(2)} \tag{2.19}$$

$$\vdots \qquad \vdots$$

$$l_{k,k-1} = a_{k,k-1}^{(k-2)} \tag{2.20}$$

$$a_{kj}^{(k-1)} = a_{kj}^{(k-2)} - l_{k,k-1}\, a_{k-1,j}^{(k-1)} \quad ; j=k-1,\ldots,n \tag{2.21}$$

$$b_k^{(k-1)} = b_k^{(k-2)} - l_{k,k-1}\, b_{k-1}^{(k-1)} \tag{2.22}$$

$$d_{k,k} = 1/a_{k,k}^{(k-1)} \tag{2.23}$$

$$a_{kj}^{(k)} = d_{kk}\, a_{kj}^{(k-1)} \quad ; \quad j \geq k \tag{2.24}$$

$$b_k^{(k)} = d_{kk}\, b_k^{(k-1)} \tag{2.25}$$

$$u_{kj} = a_{kj}^{(k)} \quad ; \quad j > k \tag{2.26}$$

This process represented by equations 2.15 to 2.22 amounts to premultiplying the derived matrix A' and b' by a n x n non-singular matrix L_k which differ from the unit matrix only in the kth row as follows:

matrix L_k ; row k = $[-l_{k1}, -l_{k2}, \ldots, -l_{k,k-1}, 1, 0, \ldots, 0]$

(2.27)

The process represented by equations 2.24 and 2.25 amounts to premultiplying the derived matrix A' and b' by an elementary matrix D_k which differs from the unit matrix in the kth diagonal element as follows:

matrix D_k ; row k = $[0, \ldots, 0, d_{kk}, 0, \ldots, 0]$ (2.28)

After all the elements on the left side of the diagonal are eliminated and all the diagonal element made unity; the derived system of equation is as follows:

$$\begin{bmatrix} 1 & u_{12} & \cdots & & u_{1n} \\ & 1 & \cdots & & u_{2n} \\ & & 1 & \cdots & u_{3n} \\ & & & & \vdots \\ & & & 1 & u_{n,n-1} \\ & & & & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \vdots \\ \\ x_n \end{bmatrix} = \begin{bmatrix} b_1^{(1)} \\ b_2^{(2)} \\ \cdot \\ \vdots \\ \\ b_n^{(n)} \end{bmatrix} \quad (2.29)$$

Solution is now obtained as follows by back substitution.

$$x_n = b_n^n$$

$$x_{n-1} = b_n^{(n-1)} - u_{n-1,n}^{(n-1)} \; x_n$$

$$x_i = b_i^{(i)} - \sum_{j>i} u_{ij}\, x_j \tag{2.30}$$

Back substitution amounts to premultiplying the derived matrix b' successively by U_{n-1}, U_{n-2}, ... U_2, U_1, where U_k is a n x n non-singular matrix differing from the matrix as follows.

$$U_k \text{ ; row } k = [0, \ldots, 0, 1, -u_{k,k+1}, \ldots, -u_{kn}] \tag{2.31}$$

Thus

$$U_1 U_2 \ldots U_{n-1}\, D_n\, L_n \ldots D_2\, L_2\, D_1 = A^{-1} \tag{2.32}$$

The non trivial element of the matrices U, D and L are stored as follows:

$$\begin{bmatrix} d_{11} & u_{12} & u_{13} & \cdots\cdots & u_{1n} \\ l_{21} & d_{22} & u_{23} & \cdots\cdots & u_{2n} \\ l_{31} & l_{32} & d_{33} & \cdots\cdots & u_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ l_{n1} & l_{n2} & & & d_{nn} \end{bmatrix} \tag{2.33}$$

This table will henceforth be called as table of L. U. factors.

Inverse of these matrices are trivial. Inverse of D_i

matrix involves only the reciprocal of the element d_{ii}. The inverse of matrix L_i and U_i involve only reversal of algebraic sign of the off diagonal elements.

Operations A^{-1}, A, $(A^t)^{-1}$, (A^t) on a column vector are represented as follows:

$$A^{-1} b = U_1 U_2 \cdots U_{n-1} D_n L_n \cdots D_2 L_2 D_1 b \qquad (2.34)$$

$$A b = L_1^{-1} L_2^{-1} D_2^{-1} \cdots L_n^{-1} D_n^{-1} U_{n-1}^{-1} \cdots U_2^{-1} U_1^{-1} b \qquad (2.35)$$

$$(A^t)^{-1} b = L_1^t L_2^t D_2^t \cdots L_n^t D_n^t U_{n-1}^t \cdots U_2^t U_1^t b \qquad (2.36)$$

$$(A^t) b = (U_1^{-1})^t (U_2^{-1})^t \cdots (U_{n-1}^{-1})^t (D_n^{-1})^t (L_n^{-1})^t \cdots (D_2^{-1})^t (L_2^{-1})^t (D_1^{-1})^t b \qquad (2.37)$$

2.1.1 <u>Numerical example</u>

The following example is based on the non symmetrical matrix A

$$A = \begin{bmatrix} 2 & 4 & 6 \\ 3 & 8 & 13 \\ 4 & 10 & 19 \end{bmatrix}$$

Table of L.U. factors for A is obtained as follows.

$$\begin{bmatrix} 2 & 4 & 6 \\ 3 & 8 & 13 \\ 4 & 10 & 19 \end{bmatrix} \to \begin{bmatrix} \frac{1}{2} & 2 & 3 \\ 3 & 8 & 13 \\ 4 & 10 & 19 \end{bmatrix} \to \begin{bmatrix} \frac{1}{2} & 2 & 3 \\ 3 & 2 & 4 \\ 4 & 10 & 19 \end{bmatrix} \to \begin{bmatrix} \frac{1}{2} & 2 & 3 \\ 3 & \frac{1}{2} & 2 \\ 4 & 10 & 19 \end{bmatrix}$$

$$\begin{bmatrix} \frac{1}{2} & 2 & 3 \\ 3 & \frac{1}{2} & 2 \\ 4 & 2 & 7 \end{bmatrix} \to \begin{bmatrix} \frac{1}{2} & 2 & 3 \\ 3 & \frac{1}{2} & 2 \\ 4 & 2 & 3 \end{bmatrix} \to \begin{bmatrix} \frac{1}{2} & 2 & 3 \\ 3 & \frac{1}{2} & 2 \\ 4 & 2 & \frac{1}{3} \end{bmatrix}$$

With $b = \begin{bmatrix} 20 & 38 & 51 \end{bmatrix}^t$; x is obtained from table of factors as follows:

$$\begin{bmatrix} 20 \\ 38 \\ 51 \end{bmatrix} \to \begin{bmatrix} 10 \\ 38 \\ 51 \end{bmatrix} \to \begin{bmatrix} 10 \\ 8 \\ 51 \end{bmatrix} \to \begin{bmatrix} 10 \\ 4 \\ 51 \end{bmatrix} \to \begin{bmatrix} 10 \\ 4 \\ 3 \end{bmatrix} \to \begin{bmatrix} 10 \\ 4 \\ 1 \end{bmatrix} \to \begin{bmatrix} 10 \\ 2 \\ 1 \end{bmatrix} \to \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$$

With $x = \begin{bmatrix} 3 & 2 & 1 \end{bmatrix}^t$; b = Ax can be obtained from table of factors as follows:

$$\begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} \to \begin{bmatrix} 10 \\ 2 \\ 1 \end{bmatrix} \to \begin{bmatrix} 10 \\ 4 \\ 1 \end{bmatrix} \to \begin{bmatrix} 10 \\ 4 \\ 3 \end{bmatrix} \to \begin{bmatrix} 10 \\ 4 \\ 51 \end{bmatrix} \to \begin{bmatrix} 10 \\ 8 \\ 51 \end{bmatrix} \to \begin{bmatrix} 10 \\ 38 \\ 51 \end{bmatrix} \to \begin{bmatrix} 10 \\ 38 \\ 51 \end{bmatrix}$$

With $b = \begin{bmatrix} 16 & 38 & 63 \end{bmatrix}^t$; $(A^{-1})^t\ b$ is obtained as follows:

$$\begin{bmatrix}16\\38\\63\end{bmatrix} \rightarrow \begin{bmatrix}16\\6\\15\end{bmatrix} \rightarrow \begin{bmatrix}16\\6\\3\end{bmatrix} \rightarrow \begin{bmatrix}16\\6\\1\end{bmatrix} \rightarrow \begin{bmatrix}12\\4\\1\end{bmatrix} \rightarrow \begin{bmatrix}12\\2\\1\end{bmatrix} \rightarrow \begin{bmatrix}6\\2\\1\end{bmatrix} \rightarrow \begin{bmatrix}3\\2\\1\end{bmatrix}$$

With $b = \begin{bmatrix} 3 & 2 & 1 \end{bmatrix}^T$; $A^t\ b$ is obtained as follows:

$$\begin{bmatrix}3\\2\\1\end{bmatrix} \rightarrow \begin{bmatrix}6\\2\\1\end{bmatrix} \rightarrow \begin{bmatrix}12\\2\\1\end{bmatrix} \rightarrow \begin{bmatrix}12\\4\\1\end{bmatrix} \rightarrow \begin{bmatrix}16\\6\\1\end{bmatrix} \rightarrow \begin{bmatrix}16\\6\\3\end{bmatrix} \rightarrow \begin{bmatrix}16\\6\\15\end{bmatrix} \rightarrow \begin{bmatrix}16\\38\\63\end{bmatrix}$$

2.1.2 <u>Hybrid Operations</u>

Method of traingular factorization can be extended for the hybrid operations as follows.

Let the hybrid column g is defined as follows -

$$g^t = (b_1, b_2, \ldots, b_k, x_{k+1}, \ldots, x_n) \tag{2.38}$$

An intermediate column vector h representing forward substitution for the first k rows is as follows:

$$h^t = (b_1^{(1)}, b_2^{(2)}, \ldots, b_k^{(k)}, x_{k+1}, \ldots, x_n$$

$$= D_k\ L_k\ D_{k-1}\ L_{k-1}, \ldots, D_2\ L_2\ D_1\ g \tag{2.39}$$

Backward substitution for the first k rows will give $x_k, x_{k-1}, \ldots, x_2, x_1$.

$$x = U_1\, U_2 \cdots U_{k-1}\, U_k\, h$$

$$x = U_1\, U_2 \cdots U_{k-1}\, U_k\, D_k\, L_k \cdots D_3\, L_3\, D_2\, L_2\, D_1\, g \tag{2.40}$$

$$b = D_1^{-1}\, L_2^{-1}\, D_2^{-1} \cdots L_n^{-1}\, D_n^{-1}\, U_{n-1}^{-1} \cdots U_{k+1}^{-1}\, D_k\, L_k \cdots D_3\, L_3\, D_2\, L_2\, D_1\, g \tag{2.41}$$

In order to obtain the complete vector; complete operation represented by (2.41) is not necessary since first k elements are known. Last $n-k$ elements may be obtained from a matrix b' defined as follows:

$$b' = L_{k+1}^{-1}\, D_{k+1}^{-1} \cdots L_n^{-1}\, D_n^{-1}\, U_{n-1}^{-1} \cdots U_{k+1}^{-1}\, D_k\, L_k \cdots D_3\, L_3\, D_2\, L_2\, D_1\, g \tag{2.42}$$

2.1.3 Numerical Example

Given $g^t = (b_1, b_2, x_3) = (20, 38, 1)$ to obtain $x = (x_1, x_2, x_3)$ for the matrix of example 2.1.1

$$\begin{bmatrix} 20 \\ 38 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 10 \\ 38 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 10 \\ 8 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 10 \\ 4 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 10 \\ 2 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$$

To obtain (b_1, b_2, b_3) alone

$$\begin{bmatrix} 20 \\ 38 \\ 1 \end{bmatrix} \to \begin{bmatrix} 10 \\ 38 \\ 1 \end{bmatrix} \to \begin{bmatrix} 10 \\ 8 \\ 1 \end{bmatrix} \to \begin{bmatrix} 10 \\ 4 \\ 1 \end{bmatrix} \to \begin{bmatrix} 10 \\ 4 \\ 3 \end{bmatrix} \to \begin{bmatrix} 10 \\ 4 \\ 51 \end{bmatrix}$$

$b = (20, 38, 51)^t$

Computer Logic

2.1.4 To obtain the L.U. factors of n x n matrix A

In this programme, elements of n x n matrix A are replaced by the L.U. factors of matrix A.

(i) Perform up to (vi) for $i = 1, \ldots, n$.

(ii) Perform (iii) for $k = 1, \ldots, i-1$.

(iii) $a_{ij} = a_{ij} - a_{ik}\, a_{kj}$; $j = k+1, \ldots, n$

(iv) $d = 1/a_{ii}$

(v) $a_{ij} = d\, a_{ij}$; $j = i+1, \ldots, n$.

(vi) $a_{ii} = d$

2.1.5 To simulate A^{-1} b

(i) Set $x_i = b_i$; $i = 1, \ldots, n$.

Forward substitution

(ii) Perform up to (iv) for $i = 1, \ldots, n$.

(iii) $x_i = x_i - \sum_{k=1}^{i-1} a_{ik} x_k$

(iv) $x_i = a_{ii} x_i$

Backward substitution

(v) Perform up to (vii) for $i = 1, \ldots, n-1$.

(vi) $j = n - i$

(vii) $x_j = x_j - \sum_{k=j+1}^{n} a_{jk} x_k$

2.1.6 <u>To simulate Ax</u>

(i) Set $b_i = x_i$ for $i = 1, \ldots, n$.

(ii) $b_i = b_i + \sum_{j=i+1}^{n} a_{ij} b_j$; $i = 1, \ldots, n-1$

(iii) Perform up to (vi) for $i = 0, \ldots, n-1$.

(iv) $j = n - i$.

(v) $b_j = b_j / a_{jj}$

(vi) $b_j = b_j + \sum_{k=1}^{j-1} a_{jk} b_k$

2.1.7 To simulate $(A^{-1})^{T}$ b

(i) Set $x_i = b_i$; $i = 1, \ldots, n$.

(ii) Perform (iii) for $i = 1, \ldots, n-1$.

(iii) $x_j = x_j - x_i a_{ij}$; $j = i+1, \ldots, n$.

(iv) Perform up to (vii) for $i = 0, \ldots, n-1$.

(v) $j = n - i$.

(vi) $x_j = x_j a_{jj}$

(vii) $x_k = x_k - a_{jk} x_j$; $k = 1, \ldots, j-1$.

2.1.8 To simulate A^t x

(i) Set $b_i = x_i$; $i = 1, \ldots, n$.

(ii) Perform up to (iv) for $i = 1, \ldots, n$.

(iii) $x_k = x_k + a_{ik} x_i$; $k = 1, \ldots, i-1$.

(iv) $x_i = x_i / a_{ii}$

(v) Perform up to (vii) for $i = 1, \ldots, n-1$.

(vi) $k = n - i$.

(vii) $x_j = x_j + x_k a_{kj}$; $j = k+1, \ldots, n$.

2.1.9 Simulation of hybrid matrix operations

(a) Given vector $g^t = (b_1, \ldots, b_k, x_{k+1}, \ldots, x_n)$ to obtain (x).

(i) Set $x_i = b_i$; $i = 1, \ldots, k$.

(ii) Perform up to (iv) for $i = 1, \ldots, k$.

(iii) $x_i = x_i - \sum_{j=1}^{i-1} a_{ij} x_j$

(iv) $x_i = a_{ii} x_i$

(v) Perform (vi) for $i = 0, \ldots, k-1$.

(vi) $x_{(k-i)} = x_{(k-i)} - \sum_{j=k-i+1}^{n} a_{k-i,j} x_j$

(b) To obtain vector (b) alone.

(i) Set $x_i = b_i$; $i = 1, \ldots, k$.

(ii) Perform up to (iv) for $i = 1, \ldots, k$.

(iii) $x_i = x_i - \sum_{j=1}^{i-1} a_{ij} x_j$

(iv) $x_i = a_{ii} x_i$

(v) Perform (vi) for $i = k+1, \ldots, n-1$.

(vi) $x_i = x_i + \sum_{j=i+1}^{n} a_{ij} x_j$

(vii) Perform up to (x) for $i = 0, \ldots, n-1+k$

(viii) $j = n - i$.

(ix) $x_j = x_j/a_{jj}$

(x) $x_j = x_j + \sum_{k=1}^{j-1} a_{jk}\, x_k$

(xi) $b_i = x_i$ for $i > k$.

2.1.10 Symmetrical matrices

If the matrix A is symmetrical; its inverse is also symmetrical for which the elements on one side of the diagonal do not have to be stored. In this section, it is shown how symmetry could be exploited in the table of factors.

$$u_{ji} = a_{ji}^{(j)} = d_{jj}\, a_{ji}^{(j-1)} \quad ; \quad i > j \tag{2.43}$$

$$l_{ij} = a_{ij}^{(j-1)} \tag{2.44}$$

For a symmetrical matrix;

$$a_{ji} = a_{ij}$$

$$a_{ji}^{(1)} = a_{ji} - a_{j1}\, a_{1i}^{(1)} \;;$$

$$= a_{ji} - a_{j1}\, d_{11}\, a_{1i}; \quad \text{for } i > j \geq 2 \tag{2.45}$$

$$a_{ij}^{(1)} = a_{ij} - a_{i1}\, a_{1j}$$

$$= a_{ij} - a_{i1}\, d_{11}\, a_{1j} \tag{2.46}$$

Hence

$$a_{ji}^{(1)} = a_{ij}^{(1)} \;;\; i > j \geq 2 \tag{2.47}$$

$$a^{(2)}_{ji} = a^{(1)}_{ji} - a^{(1)}_{j2}\ a^{(2)}_{21}$$

$$= a^{(1)}_{ji} - a^{(1)}_{j2}\ d_{22}\ a^{(1)}_{21} \ ; \text{ for } i > j \geq 3 \qquad (2.48)$$

$$a^{(2)}_{ij} = a^{(1)}_{ij} - a^{(1)}_{i2}\ a^{(2)}_{2j}$$

$$= a^{(1)}_{ij} - a^{(1)}_{i2}\ d_{22}\ a^{(1)}_{2j}$$

Hence

$$a^{(2)}_{ij} = a^{(2)}_{ji} \qquad i > j \geq 3$$

$$a^{(k)}_{ij} = a^{(k-1)}_{ij} - a^{(k-1)}_{ik}\ a^{(k)}_{kj}$$

$$= a^{(k-1)}_{ij} - a^{(k-1)}_{ik}\ d_{kk}\ a^{(k-1)}_{kj} \qquad (2.49)$$

$$\text{for } i > j > k$$

$$a^{(k)}_{ji} = a^{(k-1)}_{ji} - a^{(k-1)}_{jk}\ a^{(k)}_{ki}$$

$$= a^{(k-1)}_{ji} - a^{(k-1)}_{jk}\ d_{kk}\ a^{(k-1)}_{ki}$$

Hence, if

$$a^{(k-1)}_{ij} = a^{(k-1)}_{ji} \ ; \ i > j > k$$

$$a^{(k)}_{ij} = a^{(k)}_{ji} \qquad (2.50)$$

Since (2.50) is valid for $k = 2$; it is valid for all $k < j$.

$$a_{ij}^{(j-1)} = a_{ji}^{(j-1)} \quad ; \quad i > j$$

$$u_{ji} = d_{jj}\, a_{ji}^{(j-1)} = d_{jj}\, l_{ij} \tag{2.51}$$

Equation (2.51) indicates that it is not necessary to store both l and u terms separately. l terms can be obtained from d and u terms, thus advantages of symmetry are reflected in the table of factors. If u terms are stored row by row; l terms can be obtained column by column; by dividing the corresponding row of u by 'd' term. The formulation of A^{-1} of equation (2.34) need complete row of l terms. It appears desirable to modify the formulation so that complete column of l terms is needed instead.

TH-146

Matrix L_k defined in equation (2.27) differs from the unit matrix only in kth row. This is broken in (k-1) 'elementary matrices' E_{kj}^{l} where E_{kj}; $j < k$ has its kjth element as an essential element with the value '$-l_{kj}$'.

$$L_k = E_{k1}^{l}\, E_{k2}^{l}\, E_{k3}^{l} \cdots E_{k,k-1}^{l} \tag{2.52}$$

Since $A^{-1} = U_1\, U_2 \cdots U_{n-1}\, D_n\, L_n \cdots D_2\, L_2\, D_1$

$$= U_1\, U_2 \cdots U_{n-1}\, D_n\, E_{n1}^{l}\, E_{n2}^{l} \cdots E_{n,n-1}^{l}$$

$$\cdots D_{n-1}\, E_{n-1,1}^{l}\, E_{n-1,2}^{l} \cdots E_{n-1,n-3}^{l}\, E_{n-1,n-2}^{l}$$

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

$$\cdots\cdots D_3\, E_{31}^{l}\, E_{32}^{l}\, D_2\, E_{21}^{l}\, D_1 \tag{2.53}$$

Two elementary matrices E_{ij} and E_{kl} will commutate if $i \neq l$ and also $j \neq k$; and E_{ij}, D_k will commutate if $i \neq k$ or $j \neq k$. Equation (2.54) is regrouped as follows:

$$A^{-1} = U_1\, U_2 \,\ldots\, U_{n-1}\, D_n\, E^1_{n,n-1}\, E^1_{n,n-2}\, E^1_{n-1,n-2}$$
$$D_{n-2}\, E^1_{n,n-3}\, E^1_{n-1,n-3}\, E^1_{n-2,n-3} \,\ldots\ldots$$
$$D_k\, E^1_{n,k-1}\, E^1_{n-1,k-1} \,\ldots\ldots\, E^1_{k,k-1} \,\ldots\ldots$$
$$D_2\, E^1_{n1}\, E^1_{n-1,1} \,\ldots\ldots\, E^1_{21}\, D_1$$

Let $\quad L'_k = E^1_{nk}\, E^1_{n-1,k} \,\ldots\ldots\, E^1_{k+1,k}$

Matrix L'_k differ from the unit matrix in kth column as follows:

$$L'_k : \quad \text{col } k = \left[0, \ldots\ldots 0, 1, -1_{k+1,k}, -1_{k+2,k} \ldots\ldots, -1_{nk}\right]$$

$$A^{-1} = U_1 U_2 \ldots U_{n-1}\, D_n\, L_{n-1}\, D_{n-1}\, L'_{n-2} \ldots\ldots$$
$$D_2\, L'_1\, D_1 \qquad (2.54)$$

This equation, true for symmetrical and non-symmetrical matrices amounts to elimination by columns.

From relation (2.51);

$L'_1 = D_1\, U^t_1\, D^{-1}_1$ for symmetrical matrices;

$$A^{-1} = U_1\, U_2 \cdots U_{n-1}\, D_n\, D_{n-1}\, U^t_{n-1}\, D^{-1}_{n-1}\, D_{n-1}$$
$$D_{n-2}\, U^t_{n-2}\, D^{-1}_{n-2} \quad \cdots\cdots\cdots$$
$$\cdots\cdots\cdots\cdots\cdots\cdots$$
$$\cdots\cdots D_2\, U^t_2\, D^{-1}_2\, D_2\, D_1 U^t_1\, D^{-1}_1\, D_1$$

$$= U_1\, U_2 \cdots U_{n-1}\, D_n\, D_{n-1}\, U^t_{n-1}\, D_{n-2}\, U^t_{n-2} \cdots$$
$$\cdots D_1\, U^t_1$$

Since $U^t_i\ D_j$ commutate for $i \quad j$;

$$A^{-1} = U_1\, U_2 \cdots U_{n-1}\, D_n\, D_{n-1} \cdots D_1\, U^t_{n-1} \cdots U^t_2\ U^t_1$$

for symmetrical matrix (2.55)

2.1.11 Numerical example

Let the symmetrical matrix A is the following:

$$\begin{bmatrix} 2 & -1 & -1 & 0 \\ -1 & 4 & -2 & -1 \\ -1 & -2 & 5 & -2 \\ 0 & -1 & -2 & 4 \end{bmatrix}$$

Arithmatic steps for factorization are as follows:

$a_{11} = 1/a_{11} = 1/2$

$B = a_{11}\ a_{12} = (1/2)\ (-1) = -1/2$

$a_{22} = a_{22} - B\ a_{12} = 4-(-1/2)\ (-1) = 7/2$

$a_{12} = B \qquad = -1/2$

$a_{23} = a_{23} - B\ a_{13} = -2-(-1/2)\ (-1) = -5/2$

$a_{24} = a_{24} - B\ a_{14} = -1-(-1/2)\ (0) = -1$

$a_{22} = 1/a_{22} \qquad = 2/7$

$B = a_{11}\ a_{13} = (1/2)\ (-1) = -1/2$

$a_{33} = a_{33} - B(a_{13}) = 5-(-1/2)\ (-1) = 9/2$

$a_{34} = a_{34} - B(a_{14}) = -2-(-1/2)(0) = -2$

$a_{13} = B \qquad = -1/2$

$B = a_{22}\ a_{23} = (2/7)(-5/2) = -5/7$

$a_{33} = a_{33} - B\ a_{23} = 9/2-(-5/7)(-5/2) = 19/7$

$a_{34} = a_{34} - B\ a_{24} = -2-(-5/7)(-1) \qquad = -19/7$

$a_{23} = B = -5/7$

$a_{33} = 1/a_{33} = 7/19$

$B = a_{22}\ a_{24} = (2/7)(-1) = -2/7$

$a_{44} = a_{44} - B\ a_{24} = 4-(-2/7)(-1) = 26/7$

$a_{24} = B = -2/7$

$B = a_{33}\ a_{34} = (7/19)(-19/7) = -1$

$a_{44} = a_{44} - B\ a_{34} = 26/7-(-1)(-19/7) = 1$

$a_{34} = B = -1$

$a_{44} = 1/a_{44} = 1.$

Omitting the terms on the left of the diagonal, table of factors for the matrix is as follows:

$$\begin{bmatrix} 1/2 & -1/2 & -1/2 & 0 \\ & 2/7 & -5/7 & -2/7 \\ & & 7/19 & -1 \\ & & & 1 \end{bmatrix}$$

For a given vector $b^t = (1, 4, -1, -4)$; x may be obtained as follows:

$$\begin{bmatrix} 1 \\ 4 \\ -1 \\ -4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 9/2 \\ -1/2 \\ -4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 9/2 \\ 19/7 \\ -19/7 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 9/2 \\ 19/7 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1/2 \\ 9/7 \\ 1 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 1/2 \\ 2 \\ 1 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 2 \\ 2 \\ 1 \\ 0 \end{bmatrix}$$

2.1.12 Computer logic for symmetrical matrix

(a) Computation of table of factors for the symmetrical matrix:

(i) Perform up to (viii) for $i = 1, \ldots, n$.

(ii) Perform up to (vii) for $j = 1, \ldots, i-1$.

(iii) $B = a_{ji}\, a_{jj}$

(iv) $a_{ii} = a_{ii} - B\, a_{ji}$

(v) $a_{ji} = B$

(vi) Perform up to (vii) for $k = i+1, \ldots, n$.

(vii) $a_{ik} = a_{ik} - B\, a_{jk}$

(viii) $a_{ii} = 1/a_{ii}$

(b) Computation of $x = A^{-1}b$ from the table of factors. (Equation 2.55):

(i) Set $x_i = b_i$; $i = 1, \ldots, n$.

(ii) Perform up to (iv) for $i = 1, \ldots, n-1$.

(iii) Perform up to (iv) for $j = i+1, \ldots, n$

(iv) $x_j = x_j - x_i\, a_{ij}$

(v) $x_i = d_{ii}\, x_i$; $i = 1, \ldots, n$

(vi) Perform up to (viii) for $i = 1, \ldots, n-1$

(vii) $j = n - i$

(viii) $x_j = x_j - \sum_{k=j+1}^{n} a_{jk} x_k$

2.1.13 <u>Comparison between the method of triangular factorisation and inverse for a full matrix</u>

The two methods are comparable in the following respects:

(i) Once the inverse matrix and table of factors are obtained, computation of $A^{-1}b$ and $(A^t)^{-1}b$ needs the same number of multiplication and addition (n^2).

(ii) Elements on one side of the diagonal need not be stored for a symmetrical matrix in both the methods. Thus storage requirement is the same.

The method of triangular factorisation has the following advantages over the method of inverse:

(i) The table of factors can be obtained in one-third the operations of the inverse.

(ii) The method of triangular factorisation gives the effect of (A); A^t; and certain hybrid operations; whereas the method of inverse does not.

The advantages of inverse are:

(i) Complete solution needs k n multiplications and additions if the vector 'b' has k non zero elements.

(ii) Under some circumstances, the inverse may be more easily modified to reflect changes in the original matrix.

2.2 Compact storage scheme for sparse matrix

Although explicit matrix storage can be justified in some cases of relatively small sparse matrices, it is not practical for the matrices of most power network problems. A compact matrix storage scheme in which only the non-zero elements are retained is essential. A properly programmed compact storage scheme also results in considerable saving of computer time during matrix operations.

One possible scheme for a general matrix stores the non-zero elements of successive rows in a linear array. The column location of these non-zero elements and the location where the next row starts (row index) is stored separately. Following example explains the scheme.

2.2.1 Example

If the matrix A expressed explicitly is as follows:

d_1			u_1	
	d_2			u_2
l_1		d_3		
			d_4	u_3
		l_2		d_5

One possible compact storage scheme is as follows:

Table 2.2.1

S.No.	1	2	3	4	5	6
Row Index	1	3	5	7	9	11

S.No.	1	2	3	4	5	6	7	8	9	10
Nonzero element	d_1	u_1	d_2	u_2	l_1	d_3	d_4	u_3	l_2	d_5
Column location	1	4	2	5	1	3	4	5	3	5

Many variation of this scheme suitable for different situations, depending upon the nature of matrix or the nature of matrix operation expected are possible. An efficient storage scheme is again a trade off between the storage requirements and the computer time during the matrix operations. In this project the sparse matrix encountered are (a) Admittance matrix (b) L.U. factors of the jacobian matrix. Methods of storing these matrices are discussed below.

2.2.2 <u>Admittance matrix</u>

This matrix is complex and symmetrical and can be stored in polar or rectangular form. Elements on the left side of the diagonal need not be stored. A feasible scheme is as follows.

If the symmetrical matrix Y expressed explicitly is as follows:

d_1	a_1		a_2	
a_1	d_2	a_3		
	a_3	d_3		a_4
a_2			d_4	
		a_4		d_5

Compact storage scheme is as follows:

Table 2.2.2 a

S.No.	1	2	3	4	5
Diagonal element	d_1	d_2	d_3	d_4	d_5
Row index for right off sub-diagonal elements	1	3	4	5	5
S.No.	1	2	3	4	5
Nonzero right sub-diagonal element	a_1	a_2	a_3	a_4	-
Column location	1	4	3	5	-

It is sometimes desired to obtain all the non-zero elements of one rwo of the admittance matrix (y_{ij}; $j=1, \ldots, n$). For $j > i$; non-zero elements can be obtained by obtaining the row index for i and $i+1$. For $j < i$, a search is needed from all the row indices from 1 to i. This may be quite time consuming and due to the fact that many of these elements are likely to be zero; this approach has not been considered very efficient.

In the computer programme; additional array of row index, column location and location of the element in the array of non-zero right sub-diagonal elements are formulated.

Additional table may be as follows.

Table 2.2.2 b

S.No.	1	2	3	4	5	6
Row index for left sub-diagonal elements	1	1	2	3	4	5

S.No.	1	2	3	4	5
Column location for left sub-diagonal element	1	2	1	3	-
Locaion in the array of non-zero right sub-diagonal element	1	3	2	4	-

The last array of this table represents the location of the non-zero element in the main array (Table 2.2.2 a). If instead the non-zero element itself is represented, additional storage is needed since these elements are complex and are in real mode.

2.2.3 Table of L.U. factors

Equations (2.34) to (2.37) indicate that a complete row of elements on the left or right of diagonal is needed simultaneously. Hence the scheme similar to that of storing the symmetrical matrix has been adopted. The last array of table (2.2.2 b) however has the actual non-zero element on the left of diagonal. This is explained through Table(2.2.3) for example of (2.2.1).

Table 2.2.3:

Compact storage scheme ofor example 2.2.1:

S.No.	1	2	3	4	5	6
Diagonal element	d_1	d_2	d_3	d_4	d_5	d_6
Row index for right sub-diagonal elements	1	2	3	3	4	4
Row index for left sub-diagonal elements	1	1	1	2	2	3

S.No.	1	2	3	4
Non-zero right sub-diagonal element	u_1	u_2	u_3	-
Non-zero left sub-diagonal element	l_1	l_2	-	-
Column of non-zero right sub-diagonal element	4	5	5	-
Column of non-zero left sub-diagonal element	1	3	-	-

In case the table of factors has a symmetrical pattern of non-zero elements; as is the case with the conventional load flow solution through Newton's method, a more efficient storage scheme in terms of storage stores the complete row on the right of diagonal and complete column below the diagonal, through common index and integer array listing. Equation(2.54) may be used for A^{-1}. This needs the complete column below the diagonal at one time, instead of complete row on the left of the diagonal. This is explained through the following example.

2.2.4 Example

Let the matrix (A) with symmetrical pattern of non-zero elements written explicitly be the following:

d_1		u_1		u_2
	d_2		u_3	
l_1		d_3		u_4
	l_3		d_4	
l_2		l_4		d_5

The compact storage scheme may be as follows.

Table 2.2.4

S.No.	1	2	3	4	5	6
Diagonal element	d_1	d_2	d_3	d_4	d_5	-
Index	1	3	4	5	5	5

S.No.	1	2	3	4	5
Integer listing	3	5	4	5	-
Non-zero right sub-diagonal element	u_1	u_2	u_3	u_4	-
Non-zero sub-diagonal element below diagonal	l_1	l_2	l_3	l_4	-

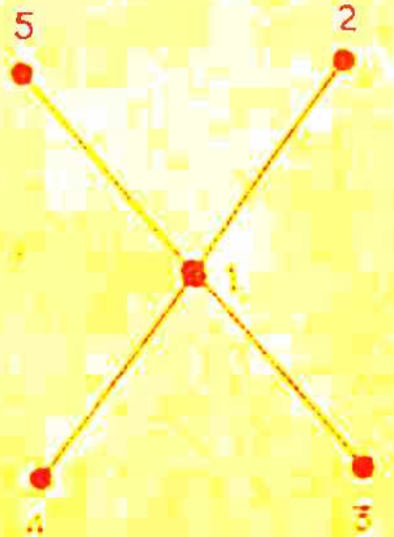
Most versions of Fortran IV compilers store a rectangular matrix 'A' of dimension M x N by column in block of memory named A. The first column is followed by the next column and so on until the N columns are exhausted. Thus the double subscripted variable A is stored as a single subscripted varaible. The element A(I, J) is interpreted as A(M * J + I). This rule is built into the Fortran compiler and the translation is automatic whenever the ordered pair is encountered in the coding. In the compact storage scheme non-zero elements are not stored. If a sub-programme could be written such that the programmes written in conventional way could with little modification use the compact storage; this should be of great value.

2.3 Optimal renumbering of nodes

There are two primary objectives[1] for an ordered Gaussian elimination. The first and the oldest seeks to control numerical accuracy through a pivoting scheme. The second aims at conservation of matrix sparsity, since the order in which the Gaussian elimination is performed on sparse matrices affects the total number of new non-zero elements generated in the course of elimination and hence the total computation time. The physical nature of large scale electrical networks and the numerical accuracy of modern computers tend to eliminate the need for pivoting and to enhance the need for the exploitation of sparsity. Survey of this section is restricted to sparsity conserving ordering scheme.

Need for a reasonable renumbering scheme can be explained by the help of Fig. 2.3.1a. With the numbering schemes the non-zeros of Y-matrix and table of L.U. factors are shown respectively in Figures 2.3.1b and 2.3.1c.

If the same system is renumbered as shown in Fig. 2.3.2a; the total number of non-zero terms of the Y-matrix shown in



(a) LINEAR GRAPH

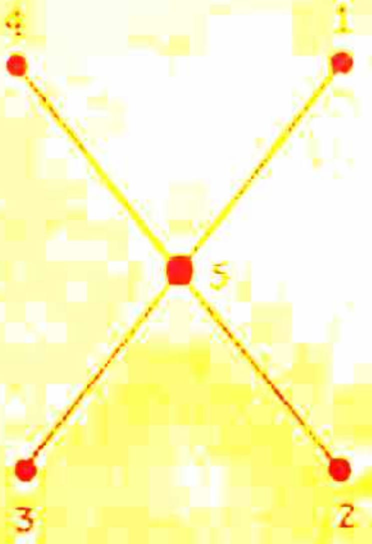
×	×	×	×	×
×	×			
×		×		
×			×	
×				×

(b) NON ZERO ELEMENTS OF Y-MATRIX.

×	×	×	×	×
×	×	×	×	×
×	×	×	×	×
×	×	×	×	×
×	×	×	×	×

(c) NON ZERO ELEMENTS OF TABLE OF FACTORS.

FIG. - 2.3.1



LINEAR GRAPH.

×				×
	×			×
		×		×
			×	×
×	×	×	×	×

(b) NON ZERO ELEMENTS OF Y-MATRIX.

×				×
	×			×
		×		×
			×	×
×	×	×	×	×

(c) NON ZERO ELEMENTS OF TABLE OF FACTORS.

FIG.-2·3·2.

Fig. 2.3.2b, remain unchanged but the number of non-zero elements of the table of factors shown in Fig. 2.3.2c is very much less compared to that shown for the same system in Fig. 2.3.1c.

For a n node system; there are of course n! permutations for numbering available. The absolute optimal order or elimination would result in the least possible non-zero elements in the table of factors. The only known method for determining the absolute optimal order is to examine all n! possibilities. This seems impractical particularly if n is large. However, several satisfactory schemes have been developed for obtaining near optimal orders.

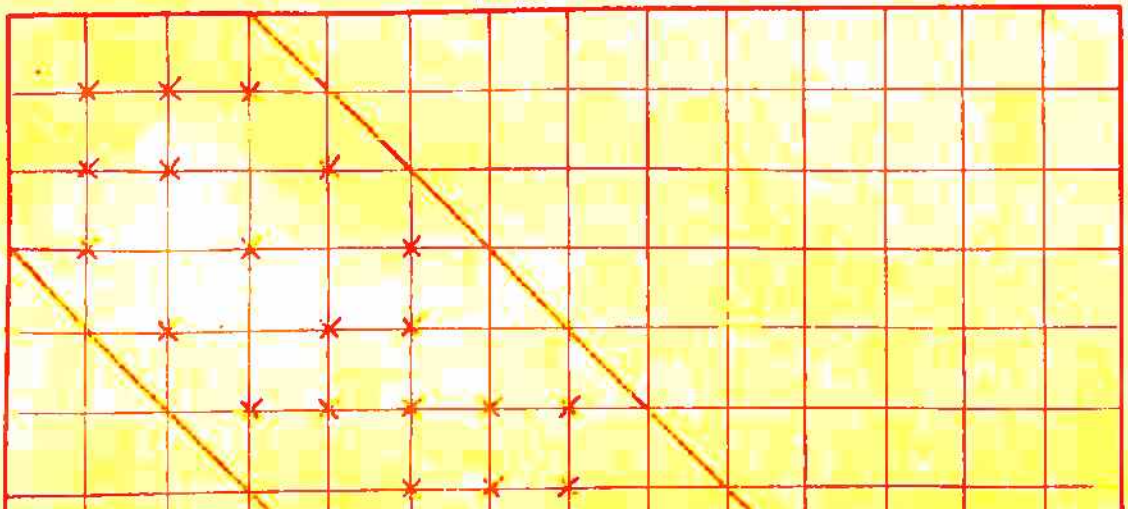
2.3.1 Matrix banding schemes

The objective of the banding schemes is to restrict the elements of a matrix to a narrow band about the major diagonal (Fig. 2.3.3.a) or about the minor diagonal (Fig.2.3.3b).

2.3.2 Minimally adjacent cut sets (MACS)

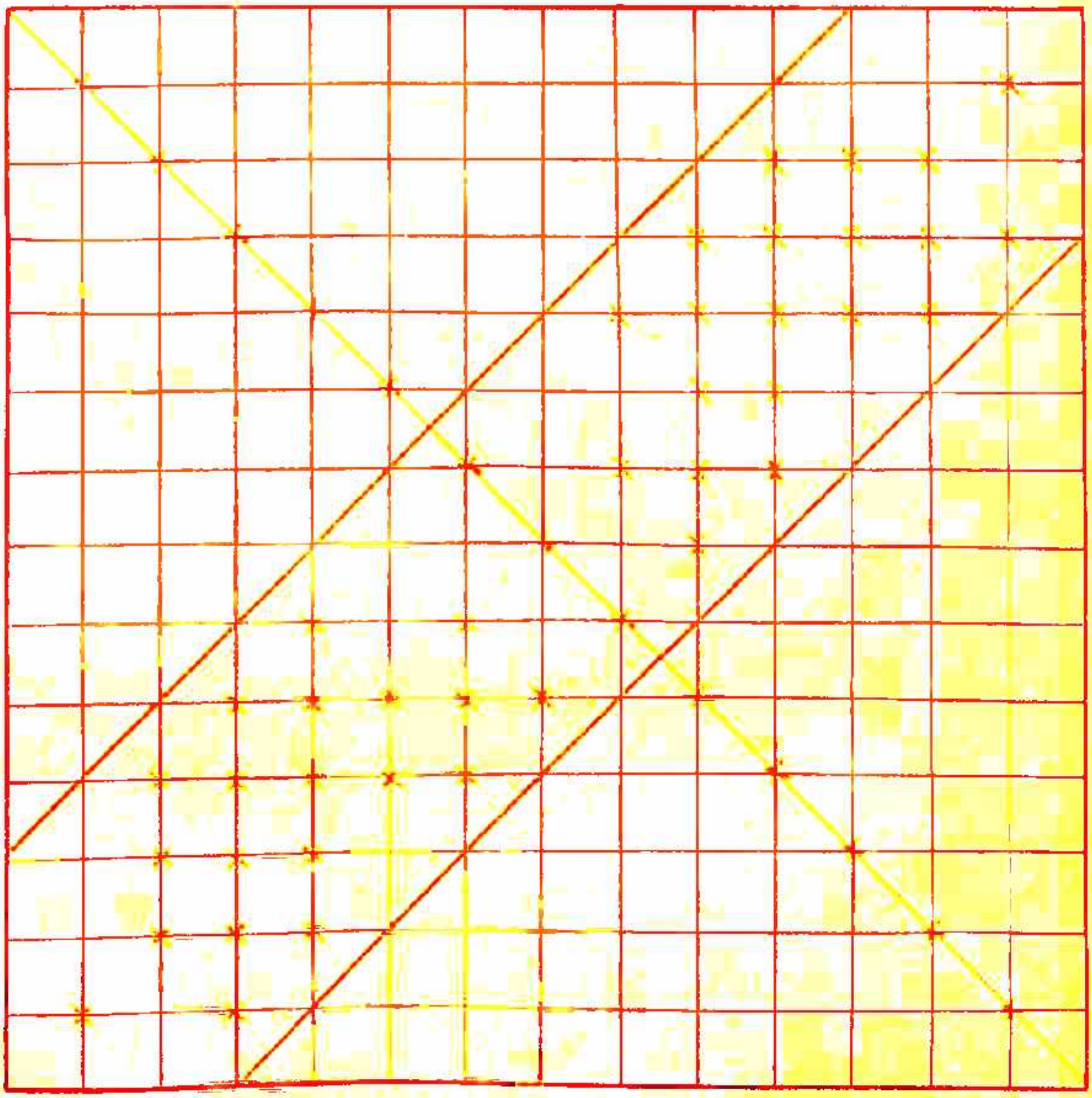
Given a set S of nodes of a connected graph G and a subset S_1 of nodes; the minimally adjacent cut sets (MACS) of S_1 are nodes from S not in S_1 but which are neighbours of nodes in S_1 .

2.3.3 An algorithm for major diagonal banding is as follows:



MAJOR DIAGONAL BANDING.

FIG. - 2·3·3 (a).



MINOR DIAGONAL BANDING.

FIG - 2·3·3(b).

'Starting with an arbitrary node as number 1, the MACS of node 1 are assigned the next sequence of numbers; the MACS of the numbered nodes are subsequently assigned the next sequence of numbers and the process is continued until all the nodes of the network are numbered'.

Linear graph of AEP 14 bus test system, numbered as above and the non-zero elements of 'Y' matrix and table of factors are shown respectively in Fig. 2.3.4 and Fig. 2.3.5.

The algorithm is weak because the choice of initial guess is critical.

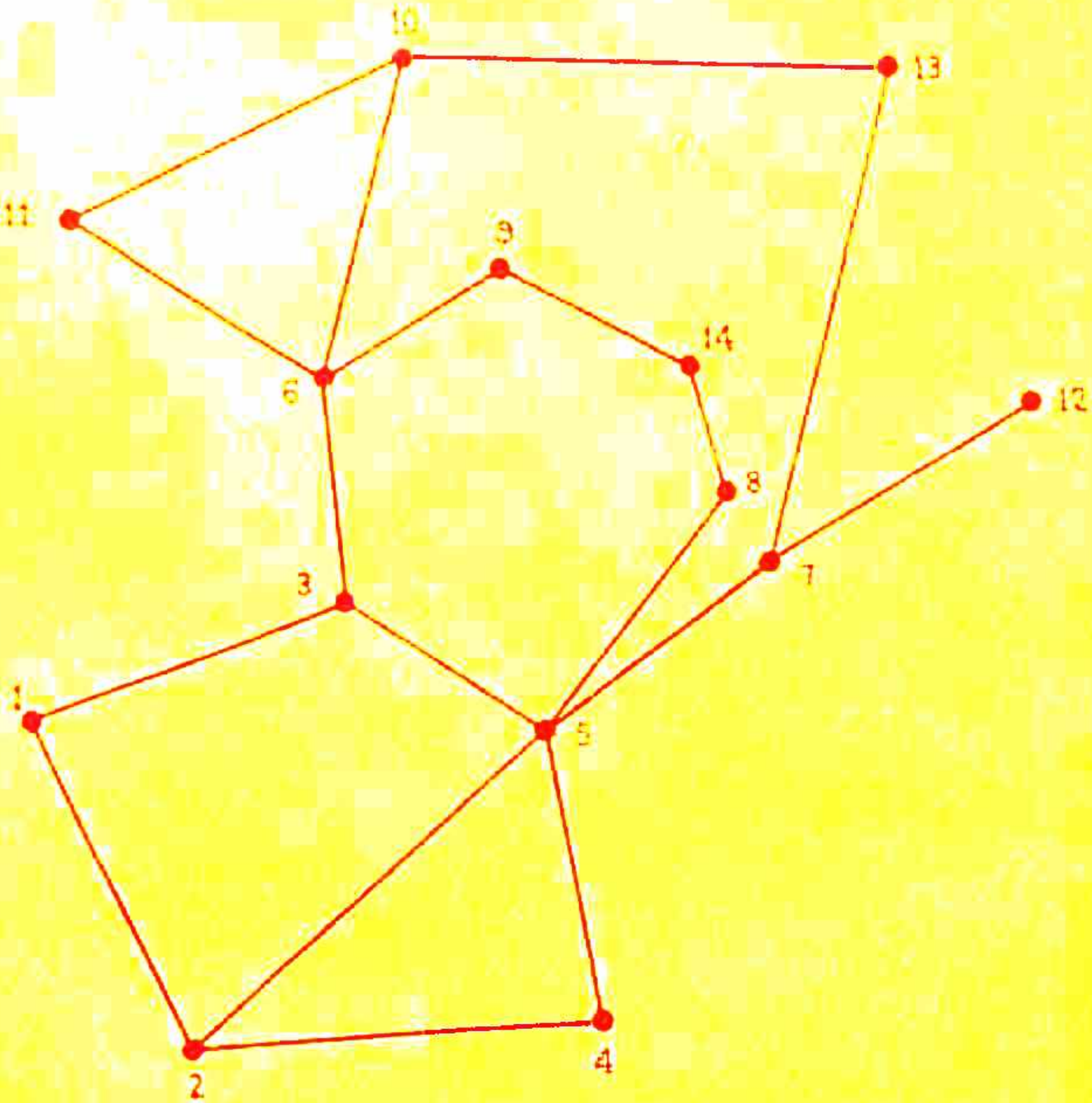
Carre[23] has suggested the following technique:

'First one node is arbitrarily chosen as last point and numbered n. Its MACS are assigned the next high valued numbers; the MACS of these numbered nodes are assigned, the next high valued numbers'.

The author does not understand how this method of numbering is effectively the same as the 'Scheme II' described by Tinney[45] (2.3.5), as claimed in the paper[23]. This is essentially the major diagonal banding scheme.

2.3.4 An algorithm for minor diagonal banding is as follows:

(i) Start with the bus having only one line and list it as node 1. In the absence of a node with only one incident branch; the node with least number of incident branches may be numbered as 1.



LINEAR GRAPH OF AEP 14 BUS TEST SYSTEM; NUMBERED THROUGH MAJOR DIAGONAL BANDING SCHEME.

FIG. - 2·3·4

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	x	x	x											
2	x	x	o	x	x									
3	x	o	x	o	x	x								
4		x	o	x	x	o								
5		x	x	x	x	o	x	x						
6			x	o	o	x	o	o	x	x	x			
7					x	o	x	o	o	o	o	x	x	
8					x	o	o	x	o	o	o	o	o	x
9						x	o	o	x	o	o	o	o	x
10						x	o	o	o	x	x	o	x	o
11						x	o	o	o	x	x	o	o	o
12							x	o	o	o	o	x	o	o
13							x	o	o	x	o	o	x	o
14								x	x	o	o	o	o	x

X NON ZERO ELEMENTS OF Y-MATRIX

O ADDITIONAL NON ZERO ELEMENTS OF TABLE OF FACTORS

NON ZERO ELEMENTS OF AEP 14 BUS TEST SYSTEM, NUMBERED THROUGH MAJOR DIAGONAL BANDING SCHEME.

FIG.- 2.3.5.

(ii) List the MACS of node 1 as the last numbered nodes.

(iii) Assign the next lower valued numbers to the MACS of the numbered nodes.

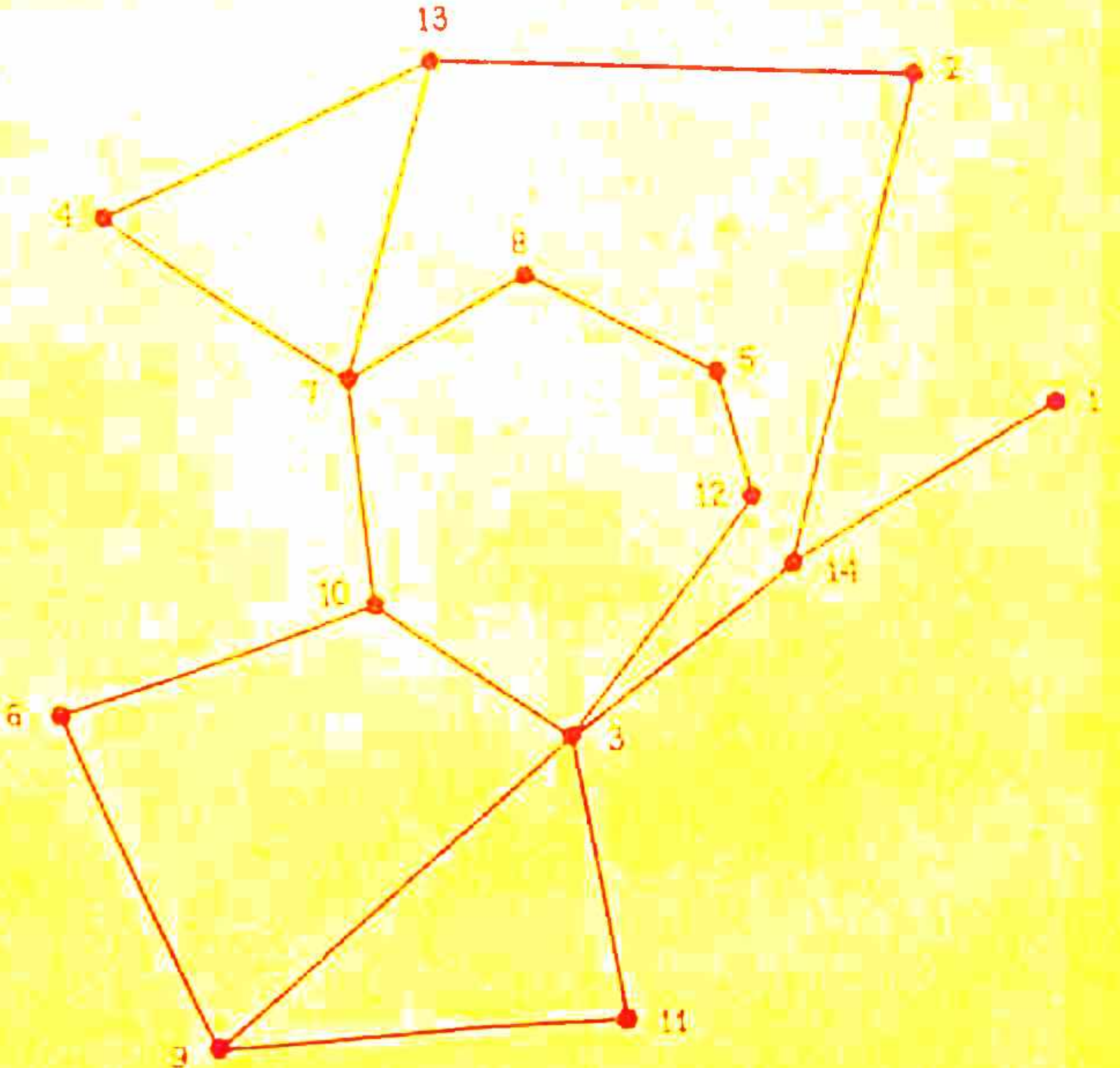
(iv) Assign the next high valued numbers to the MACS of the numbered nodes and return to step (iii)

Linear graph of AEP 14 bus test system and the non-zero elements of the Y-matrix and table of factors are shown respectively in Fig. 2.3.6 and Fig. 2.3.7.

Obviously the diagonal banding schemes do not work for the system shown in Fig. 2.3.1a of which the absolute optimal order is in Fig. 2.3.2a in which the non-zero element of the table of factors remain the same as the non-zero elements of Y-matrix.

2.3.5 Tinney[1,31,33,45] has given three schemes for near optimal renumbering. These schemes are claimed to be generally more effective than the banding schemes. These are listed in increasing order of complexity, execution time and optimality. It is assumed that the nodes are originally numbered according to some external criterion and then renumbered according to the schemes.

I. Number the rows according to the number of non-zero off diagonal elements before elimination. In this scheme the rows with only one off diagonal element are numbered first, those with two elements second, etc., and those



LINEAR GRAPH OF AEP 14 BUS TEST SYSTEM, NUMBERED THROUGH MINOR DIAGONAL BANDING SCHEME.

FIG. - 2.3.6

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	x													x
2		x											x	x
3			x						x	x	x	x		x
4				x			x						x	
5					x			x				x		
6						x			x	x				
7				x			x	x		x			x	
8					x		x	x		O		O		
9			x			x			x	O	x	O		O
10			x			x	x	O	O	x	O	O		O
11			x						x	O	x	O		O
12			x		x			O	O	O	O	x		O
13		x		x			x						x	O
14	x	x	x						O	O	O	O	O	x

X NON ZERO ELEMENTS OF Y- MATRIX

O ADDITIONAL NON ZERO ELEMENTS OF TABLE OF FACTORS

NON ZERO ELEMENTS OF AEP 14 BUS TEST SYSTEM ; NUMBERED THROUGH MINOR DIAGONAL BANDING SCHEME.

FIG.- 2·3·7

with the most elements last. This scheme does not take into account any of the subsequent effects of the elimination process. The only information needed is the number of non-zero element in each row of the original matrix.

II. Number the rows such that at each step of the process the next row,to be operated upon is the one with the fewest non-zero elements. If more than one row meet this criterion, select any one. This scheme needs a simulation of the effects on the accumulation of non-zero elements of elimination by columns. Required input information is a list by rows of the column numbers of the non-zero off diagonal elements.

III. Number the rows such that at each step of the process, the next row to be operated upon is the one that will introduce the fewest new non-zero elements.

This involves a trial simulation of every feasible alternative at each step. Input information is the same as for scheme (II).

It is hard to evaluate the three schemes since the performance of any scheme depends upon the type of system, However scheme I, though simplest, does not appear to be efficient; since subsequent built up of non-zero elements are not at all taken care of, if many solutions of the matrix are needed.

Scheme III needs large computer time since it involves trying many possible permutations.

2.3.6 At present author's subroutine works as follows:

> 'Number the rows such that the next row is the one having fewest non-zero elements in the table of factors of that row',

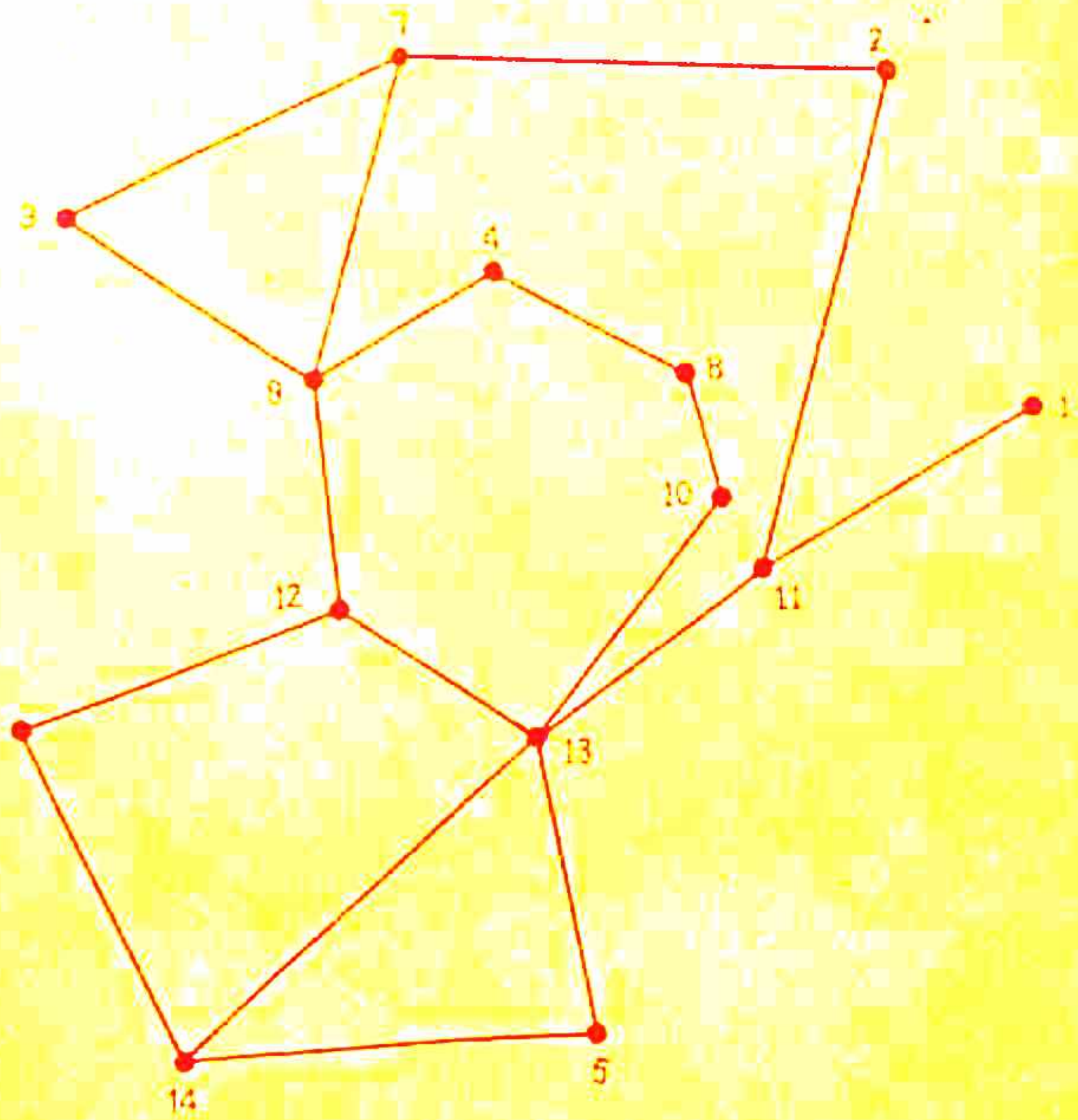
The author is now sure if this is the Tinney's II scheme.

A comparison of some of the techniques applied on AEP 14 bus test system is as follows:

Non zero elements in the Y-matrix	=	50
Additional non-zero elements with major diagonal banding scheme (Fig. 2.3.5)	=	54
Additional non-zero elements with minor diagonal banding scheme (Fig. 2.3.7)	=	24
Additional non-zero elements with author's subroutine (Fig.2.3.9)	=	16

Computer logic of author's subroutine for renumbering the buses is as follows:

For the set 'S' of n nodes; the nodes already renumbered from 1, ..., n_1 be in the subset S_1; where n_1 is the total number of nodes already renumbered; remaining nodes are in subset S_2 such that $S = S_1 + S_2$.



LINEAR GRAPH OF AEP 14 BUS TEST SYSTEM, NUMBERED THROUGH AUTHORS SUBROUTINE.

FIG.- 2·3·8

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	x										x			
2		x					x				x			
3			x				x		x					
4				x				x	x					
5					x								x	x
6						x						x		x
7		x	x				x		x		O			
8				x				x	O	x				
9			x	x			x	O	x	O	O	x		
10								x	O	x	O	O	x	
11	x	x					O		O	O	x	O	x	
12						x			x	O	O	x	x	O
13					x					x	x	x	x	x
14					x	x						O	x	x

X NON ZERO ELEMENTS OF Y-MATRIX

O ADDITIONAL NON ZERO ELEMENTS OF TABLE OF FACTORS

NON ZERO ELEMENTS OF AEP 14 BUS TEST SYSTEM MATRIX ; NUMBERED THROUGH AUTHOR'S SUBROUTINE.

FIG.-2.3.9.

(i) Let $n_1 = 0$; and all the nodes be in S_2.

(ii) For all the remaining nodes, having their old numbers as $i = 1, \ldots, n$; $i \notin S_1$; perform up to (vi).

(iii) Obtain the columns of non-zero elements of ith row from the row indices and column locations; and let them be in set C_1. Other nodes are in set C_2 such that $S = C_1 + C_2$.

(iv) Perform up to (v) for $j = 1, \ldots, n_1$, where j is the new number of buses already renumbered. This way all the columns of non-zero elements of table of factor is obtained if the next node is i.

(v) Obtain the old number for j and if it lies in set C_1, all the columns of the elements of the table of factors on the right hand side of the diagonal for the jth row obtained in step (viii) are also brought in set C_1.

(vi) Obtain the total number of elements n_2 in the set $C_1 \cdot S_2$. If they are less than the least such numbers obtained for other values of i; let $m = n_2$ and $k = i$.

(vii) $n_1 = n_1 + 1$

(viii) The next row to be operated upon is k and its new number is n_1. Obtain for this row the old column numbers of the non-zero elements on the right side in the table of factors as obtained in step (iv) and (v) and store them in the compact storage scheme for n_1th row, as shown in table 2.2.2 a.

(ix) If $n_1 < n$; repeat from step (ii). If $n_1 = n$, renumbering job is complete.

-x-

CHAPTER III

FORMULATION OF NETWORK ADMITTANCE MATRIX

In this chapter algorithm for forming the admittance matrix and storing it in the compact form is described. Effects of tap changing transformer and phase shifting transformer on the admittance matrix are described but the algorithm takes care of only tap changing transformer. With phase shifting transformer, the admittance matrix is no more symmetrical. Hence it is considered desirable to account for additional terms due to the phase shifting transformer separately.

3.1 Admittance matrix of a line connected between two nodes (Fig. 3.1)

$$\begin{bmatrix} I_i \\ I_j \end{bmatrix} = \begin{bmatrix} y+jB & -y \\ -y & y+jB \end{bmatrix} \begin{bmatrix} V_i \\ V_j \end{bmatrix} \tag{3.1}$$

where $y = 1/(r+jx)$ (3.2)

$$= \frac{r}{(r^2+x^2)} - \frac{jx}{(r^2+x^2)}$$

3.2 Admittance matrix of a line connected between two nodes through an adjustable transformer (Fig. 3.2)

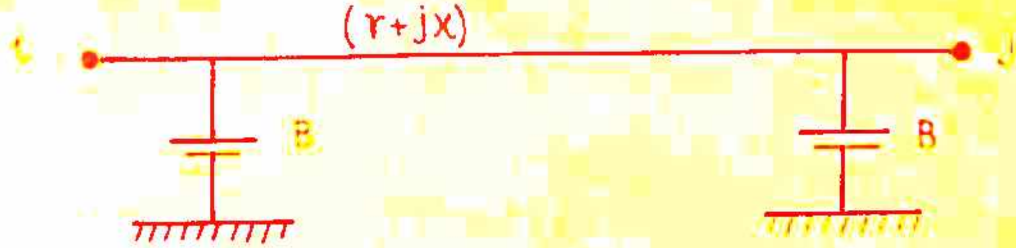


FIG. 3·1

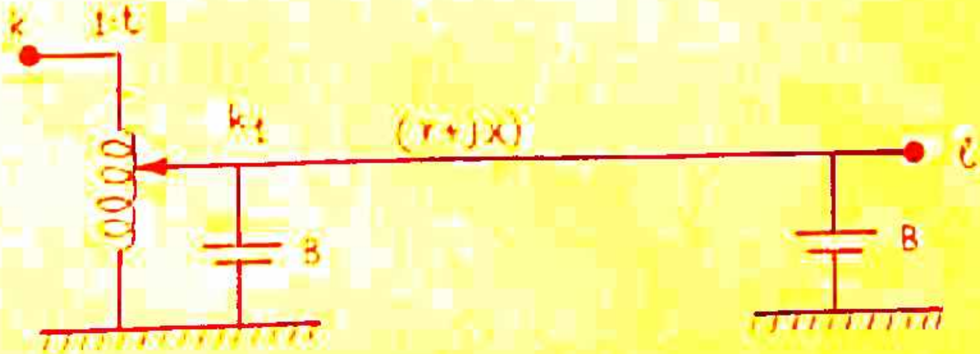


FIG. 3·2

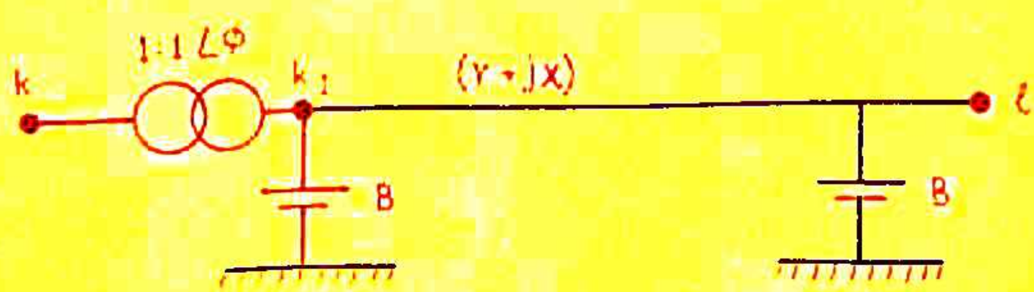


FIG. 3·3

$$\begin{bmatrix} I_{k1} \\ \\ I_l \end{bmatrix} = \begin{bmatrix} y+jB & -y \\ \\ -y & y+jB \end{bmatrix} \begin{bmatrix} V_{k1} \\ \\ V_l \end{bmatrix} \qquad (3.3)$$

If transformer losses are ignored;

$$V_{k1} = t\,V_k$$

$$I_k = t\,I_{k1}$$

$$\begin{bmatrix} I_k \\ \\ I_l \end{bmatrix} = \begin{bmatrix} t^2(y+jB) & -ty \\ \\ -ty & (y+jB) \end{bmatrix} \begin{bmatrix} V_k \\ \\ V_l \end{bmatrix} \qquad (3.4)$$

where y is defined by equation 3.2

k = near node of the transformer

l = far node of the transformer

t = off nominal turn ratio

For every adjustable tap setting transformer, its near node, far node, (y+jB), (-y) and location in the admittance array are stored separately.

If the transformer tap setting is modified; only one diagonal element and only one stored off diagonal element is modified as follows.

$$y^2_{kk} = y^1_{kk} + (t_2 - t_1)\,(y + jB) \qquad (3.5)$$

$$y^2_{kl} = y^1_{kl} - (t_2 - t_1)\,(y) \qquad (3.6)$$

where y^1_{kk} and y^1_{kl} refer the tap setting t_1 and y^2_{kk} and y^2_{kl} refer to the tap setting t_2.

3.3 Admittance matrix of a line connected between two nodes through phase shifting transformer (Fig. 3.3)

$$\begin{bmatrix} I_{k1} \\ I_l \end{bmatrix} = \begin{bmatrix} y+jB & -y \\ -y & y+jB \end{bmatrix} \begin{bmatrix} V_{k1} \\ V_l \end{bmatrix} \tag{3.7}$$

$$\begin{aligned} V_{k1} &= V_k \; \varepsilon^{j\phi} \\ I_k &= I_{k1} \; \varepsilon^{-j\phi} \end{aligned} \tag{3.8}$$

$$\begin{bmatrix} I_k \\ I_l \end{bmatrix} = \begin{bmatrix} (y+jB) & -y \; \varepsilon^{-j\phi} \\ -y \; \varepsilon^{j\phi} & (y+jB) \end{bmatrix} \begin{bmatrix} V_k \\ V_l \end{bmatrix} \tag{3.9}$$

Thus with the presence of phase shifting transformer, the admittance matrix is no more symmetrical.

3.4 Computer logic for forming the admittance matrix and storing it in compact form

The admittance matrix is formed by first setting the diagonal and off diagonal elements to zero. Data for one line which include the two nodes i and j to which the line is connected, line resistance and reactance, shunt admittance and off nominal turn ratio and near node of the transformer (if present) are read and its admittance matrix formed and the elements added to the corresponding elements of compactly stored matrix. It is assumed that the number of first bus is smaller than the number of second bus and data are read in the ascending order of i. Data need not be arranged in the ascending order of j.

<u>Notations</u>

y_d^i = $g_d^i + j\, b_d^i$ = diagonal element of ith node

I_1^i = index sequence of ith row right off diagonal elements

I_2^i = index sequence of ith row, left off diagonal elements

L_1^i = column of ith right off diagonal element

L_2^i = column of ith left off diagonal element

y^i = $g^i + j\, b^i$ = ith right off diagonal element

L_3^i = corresponding location of ith left off diagonal element in the array of right off diagonal elements

$\bar{m}$ = expected maximum number of lines

n_t = total number of adjustable tap setting transformers

y_{td}^i = $g_{td}^i + j\, b_{td}^i$ = $(y+jB)$ of equation 3.3, for ith adjustable tap setting transformer

k_1 = count of rows

k_2 = count of non-zero elements

t^i = tap setting of ith adjustable tap setting transformer

y_t^i = $g_t^i + j\, b_t^i$ = $(-y)$ of equation 3.3 corresponding to ith adjustable tap setting transformer

I_n^i and I_f^i = near and far nodes of ith adjustable tap setting transformer respectively

L_t^k = location of i - jth right off diagonal element in array y if i - jth element are connected through kth adjustable transformer.

The logic is as follows :

(i) Set y_t^l and y_{td}^l; $l = 1, \ldots, \bar{n}_t$, y_d^l ; $l = 1, \ldots, n$ and y^l; $l = 1, \ldots, \bar{m}$; n_t to zero.

(ii) Set $k_1 = 1$ and $k_2 = 0$

(iii) Set the real variables y_1^l and y_2^l; and integer variables i_1^l to zero for $l = 1, \ldots, n$.

(iv) Read i, j, r, x, B, tap setting and near node of the transformer (if present). If i is equal to k_1, proceed from (x), if i is greater than k_1, proceed from (v), and i less than k_1 indicates data not arranged in ascending order of i and the programme is terminated.

(v) $I_1^{k1} = k_2 + 1$

(vi) Perform (vii) for $l = k_1, \ldots, n$; such that $i_1^l \neq 0$.

(vii) $k_2 = k_2 + 1$, $L_1^{k2} = l$, $g^{k2} = y_1^l$, $b^{k2} = y_2^l$, $y_1^l = 0$, $y_2^l = 0$, $i_1^l = 0$.

(viii) $k_1 = k_1 + 1$ and if k_1 is less than i (of step iv) go to (ix); otherwise to (x).

(ix) $I_1^{k1} = k_2 + 1$ and proceed from (viii).

(x) Check if j is greater than i. If not, data are not as required and the programme is terminated.

(xi) $i_1^j = 1$, $b_3 = 1/(r^2 + x^2)$, $b_1 = rb_3$, and $b_2 = -xb_3$.

(xii) If there is no transformer tap, proceed from (xiv), if the transformer tap is adjustable, proceed from (xiii), if not adjustable, proceed from (xv).

(xiii) $n_t = n_t + 1$

$y_{td}^{nt} = b_1 + jb_2 + jB$

$y_t^{n_t} = -b_1 - jb_2$

$I_n^{nt} = i_n$ = near end

$I_f^{n_t} = i_f$ = far end

t^{n_t} = off nominal turn ratio.

Proceed from (xvi).

(xiv) $y_d^i = y_d^i + b_1 + jb_2 + jB$

$y_d^j = y_d^j + b_1 + jb_2 + jB$

$y_1^j = y_1^j - b_1$

$y_2^j = y_2^j - b_2$

Proceed from (xvii).

(xv) Obtain the near end and far ends i_n and i_f respectively.

(xvi) $c_1 = b_1 t$

$c_2 = b_2 t$

$y_1^j = y_1^j - c_1$

$y_2^j = y_2^j - c_2$

$y_d^{i_f} = y_d^{i_f} + b_1 + jb_2 + jB$

$y_d^{i_n} = y_d^{i_n} + t(c_1 + jc_2) + jBt^2$

(xvii) Check if this is the last cord of line data, if not proceed from (iv).

(xviii) Store the non-zero right off diagonal elements of ith row as in (vi) and (vii).

(xix) Search for L_t^k , $k = 1,\ldots, n_t$, L_2^i , L_3^i for $i = 1,\ldots, k_2$ and I_2^k , $k = 1,\ldots, n$.

(xx) Read shunt admittance data and modify the diagonal elements.

-x-

CHAPTER IV

POWER FLOW EQUATIONS

In this chapter some properties of power flow equations are discussed. Using some of the identities, a lot of computer time can be saved in computing the system jacobian and powers.

4.1 Steady state relations

Under steady state power flow equations in terms of the admittance matrix may be written as follows:

$$P_i = V_i \sum_j V_j\, y_{ij} \cos(\theta_i - \theta_j - \Psi_{ij}) \tag{4.1}$$

$$Q_i = V_i \sum_j V_j\, y_{ij} \sin(\theta_i - \theta_j - \Psi_{ij}) \tag{4.2}$$

Using the notation of Vanness and Griffin[50]:

$$H_{ij} = \frac{\partial P_i}{\partial \theta_j} \tag{4.3}$$

$$N_{ij} = V_j \frac{\partial P_i}{\partial V_j} \tag{4.4}$$

$$J_{ij} = \frac{\partial Q_i}{\partial \theta_j} \tag{4.5}$$

$$L_{ij} = V_j \frac{\partial Q_i}{\partial V_j} \tag{4.6}$$

Thus

$$H_{ij} = V_i\ V_j\ y_{ij}\ \sin(\theta_i - \theta_j - \psi_{ij}) \tag{4.7}$$

$$N_{ij} = V_i\ V_j\ y_{ij}\ \cos\ (\theta_i - \theta_j - \psi_{ij}) \tag{4.8}$$

$$J_{ij} = -V_i\ V_j\ y_{ij}\ \cos\ (\theta_i - \theta_j - \psi_{ij}) \tag{4.9}$$

$$L_{ij} = V_i\ V_j\ y_{ij}\ \sin(\theta_i - \theta_j - \psi_{ij}) \tag{4.10}$$

$$H_{ii} = -V_i \sum_{j\neq i} V_j\ y_{ij}\ \sin(\theta_i - \theta_j - \psi_{ij}) \tag{4.11}$$

$$N_{ii} = 2\ V_i^{\ 2}\ g_{ii} + V_i \sum_{j\neq i} V_j\ y_{ij}\ \cos(\theta_i - \theta_j - \psi_{ij}) \tag{4.12}$$

$$J_{ii} = V_i \sum_{j\neq i} V_j\ Y_{ij}\ \cos(\theta_i - \theta_j - \psi_{ij}) \tag{4.13}$$

$$L_{ii} = -2\ V_i^{\ 2}\ b_{ii} + V_i \sum_{j\neq i} V_j\ y_{ij}\ \sin(\theta_i - \theta_j - \psi_{ij}) \tag{4.14}$$

From these equations, following identities result :

$$H_{ij} = s_{ij}\ g_{ij} - c_{ij}\ b_{ij} \tag{4.15}$$

$$N_{ij} = c_{ij}\ g_{ij} + s_{ij}\ b_{ij} \tag{4.16}$$

where,

$$s_{ij} = V_i\ V_j\ \sin(\theta_i - \theta_j) = f_i\ e_j - e_i\ f_j \tag{4.17}$$

$$c_{ij} = V_i\ V_j\ \cos(\theta_i - \theta_j) = e_i\ e_j + f_i\ f_j \tag{4.18}$$

$$J_{ij} = -N_{ij} \tag{4.19}$$

$$L_{ij} = H_{ij} \tag{4.20}$$

$$H_{ii} = - \sum_{j \neq i} H_{ij} \tag{4.21}$$

$$J_{ii} = - \sum_{j \neq i} J_{ij} \tag{4.22}$$

$$P_i = V_i^2 \, g_{ii} + J_{ii} \tag{4.23}$$

$$Q_i = -V_i^2 \, b_{ii} - H_{ii} \tag{4.24}$$

$$N_{ii} = V_i^2 \, g_{ii} + P_i \tag{4.25}$$

$$L_{ii} = -V_i^2 \, b_{ii} + Q_i \tag{4.26}$$

The jacobian 'M' of the set of equations 4.3 to 4.6 is a square matrix of order 2n :

$$M = \begin{bmatrix} H & N \\ J & L \end{bmatrix} \tag{4.27}$$

where H, N, J and L are n x n square matrix whose ijth elements are H_{ij}, N_{ij}, J_{ij} and L_{ij}, respectively.

4.2 <u>Theorem</u>

Whatever the values of V and θ; matrices H, J and M are singular.

<u>Proof</u>:

From equations 4.21 and 4.22; sum of elements of any row of matrix H and J is zero. Hence the matrix H, J and M are singular.

4.3 Theorem

There do not exist in general vectors V and Θ such that the equations 4.1 and 4.2 are satisfied for any given value of P_i and Q_i; i = 1, ..., n.

Proof:

Let a set of variables V_o, θ_o, P_o and Q_o satisfy equations 4.1 and 4.2. If the active and reactive powers are perturbed by $\varepsilon \eta_p$ and $\varepsilon \eta_q$ respectively where the scaler $\varepsilon \to 0$ and η_p and η_q are vector quantities, such that

$$P = P_o + \varepsilon \eta_p$$

$$Q = Q_o + \varepsilon \eta_q$$

If corresponding per unit variation in voltage magnitude and variation in phase angles are respectively ΔV and $\Delta\theta$; these are given by the following equations

$$\varepsilon \begin{bmatrix} \eta_p \\ \\ \eta_q \end{bmatrix} = M \begin{bmatrix} \Delta\theta \\ \\ \Delta V \end{bmatrix} \qquad (4.28)$$

Since M is singular; $\Delta\theta$ and ΔV cannot be obtained for any arbitrary η_p and η_q. Hence θ and V cannot be obtained for any arbitrary P and Q.

4.4 Load flow problem

In order to be able to solve for bus bar voltages and angles; given the complex bus bar powers; phase angle of one bus bar must be taken as reference. This deletes one column of H and J. By taking active power of one bus as dependent variable, one row of M is deleted. This makes the matrix non-singular. However in order to be able to fix the order of bus bar voltage magnitudes; it is desirable that voltage magnitude of one bus be fixed. Thus three types of buses are represented in the load flow calculations.

Slack bus (v θ): V and θ are prescribed and P and Q are the dependent variables. Selection of this bus is necessary for the reasons already mentioned.

Voltage regulated bus (p v): P and V are prescribed; Q and θ are the dependent variables.

Load bus (p q): P and Q are prescribed; V and θ are dependent variables.

4.5 Compatibility relations

Since M is a singular matrix it has zero as one of the eigen values; and let $(\lambda^t, \mu^t)^t$ be the eigen vector of M^t corresponding to this eigen value; where λ and μ are n x 1 column vectors.

$$M \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix} = \begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} \qquad (4.29)$$

Premultiplying by (λ^t, μ^t) on both sides; following compatibility relations are satisfied:

$$\sum_i \lambda_i \Delta P_i + \sum_i \mu_i \Delta Q_i = 0 \qquad (4.30)$$

4.6 General compatibility relation

Variational equations of the power flow problem can be written as follows:

$$\begin{bmatrix} \Delta P \\ \hline \vdots \\ \Delta Q_k \\ \vdots \\ \Delta V_l \end{bmatrix} = \left[\begin{array}{ccc|ccc} & H & & & N & \\ \hline J_{k1} & \cdots\cdots & J_{kn} & L_{k1} & \cdots\cdots & L_{kn} \\ 0 & & 0 & 0 & \ldots 1 \ldots & 0 \end{array}\right] \begin{bmatrix} \Delta\theta \\ \hline \Delta V \end{bmatrix} \qquad (4.31)$$

where $k \in$ set of bus bars with Q as independent variable

$l \in$ set of bus bars with V as independent variable.

The jacobian of equation 4.31 is also singular, hence the compatibility relation is as follows:

$$\sum_i \lambda_i \Delta P_i + \sum_k \mu_k \Delta Q_k + \sum_l u_l \Delta V_l = 0 \qquad (4.32)$$

where $(\lambda^T, \mu^T, u^T)^T$ is the eigen vector of the transposed jacobian corresponding to zero eigen value.

4.7 Comments

All λ 's in equations 4.30 and 4.32 will have the same sign.

If only powers at two nodes i and j are changed by ΔP_i and ΔP_j and other incremental quantities do not undergo any change;

$$\Delta P_i \lambda_i + \Delta P_j \lambda_j = 0$$

$$\frac{P_i}{P_j} = - \frac{\lambda_j}{\lambda_i} \tag{4.33}$$

It is intuitively expected for a power system under normal loading that if power at one node increases, the power at the other node decreases, if all other powers remain unchanged. Hence $\Delta P_i/\Delta P_j < 0$. This is possible if all λ's have the same sign.

4.8 Correlation between compatibility and loss formula relation

Through the application of tensor theory, the total system losses can be expressed in terms of all plant outputs, after a number of assumptions are made, as follows[53] :

$$P_L = \sum_{i \in G} \sum_{j \in G} P_i B_{ij} P_j \tag{4.34}$$

Incremental changes in generation, with demand unchanged will result in the following incremental change in loss:

$$\sum_{i \in G} \Delta P_i = \Delta P_L$$

$$= 2 \sum_{i \in G} \left[\sum_{j \in G} B_{ij} \; P_j \right] \Delta P_i$$

or $$\sum_{i \in G} \left(1 - 2 \sum_{j \in G} B_{ij} \; P_j\right) \Delta P_i = 0 \tag{4.35}$$

Comparing equations 4.32 and 4.35;

$$\lambda_i = 1 - 2 \sum_{j \in G} B_{ij} \, P_j \tag{4.36}$$

-x-

CHAPTER V

LOAD FLOW SOLUTIONS

In this chapter, various load flow programmes are lumped into very general categories and compared. Sufficiency conditions for convergence of iterative procedures are discussed through the fixed point principle and applied to admittance matrix, impedence matrix and Newton's method for load flow.

5.1 Classification

Load flow programmes can be broken into the following broad categories[9]:

(a) Low accuracy special purpose load flows[30]: Such load flow programmes are needed where emphasis is not an accurate detailed modelling of power systems. Emphasis instead is on simple criteria such as what variables are out of limit and for what reasons. Usually this is followed by performing a series of input data changes to simulate possible system contingencies. For each such contingency the operator will like to know if any other components will be affected which may cause another system failure and eventually lead to a cascading system breakdown. The contingencies studied are usually in the form of line outages, generator outages, short circuits and various other effects.

(b) Conventional load flows: Such load flows need accurate set of input data and are run to appropriately small tolerance. The trade offs include programming convenience,,speed and memory requirement.

(c) Very accurate load flows: Many applications such as transient stability studies and static optimisation need in many cases a very accurate load flow programme.

5.2 Programming algorithms[9]

These algorithms for load flow solutions are almost as numerous as the number of authors who have written about load flows. These can broadly be lumped in the following categories:

(a) Admittance matrix method: These methods use Gauss or Gauss Seidal iterative technique and system admittance matrix with ground as reference. Chief advantages of this method are ease of programming and most efficient utilization of core memory of any presently known load flow methods. It is usually used when a very large system need be studied on a computer with limited core memory. However it needs long running time and has possible non-convergence.

(b) Impedence matrix method: This method also uses Gauss or Gauss Seidal iterative technique and system impedence matrix with one of the nodes as reference. Convergence of this method is better behaved than the admittance matrix method but Z-matrix is a full matrix while Y-matrix sparse. The author has been able to overcome the disadvantages of the full matrix by making use of L.U. factors, of the admittance matrix formed by taking slack bus as reference. This works much faster and needs much smaller storage if compact storage is used, compared to conventional Z-matrix. Many authors have suggested the application of 'piece-wise methods'[6,22,23,24] to overcome the disadvantages of excessive storage requirement. The real advantages of this method are apparent when the user wishes to study the effect of short circuits, line and generator outages or changes in tie line flows.

(c) Newton's method: This technique has the best convergence characteristics of all the presently available methods. Solution accuracy is restricted only by the round off errors. This method if efficiently programmed and buses optimally renumbered and compact storage used for the Jacobian and admittance matrix, can be made as the fastest, This method is probably best suited for optimal power flow studies because of very high accuracy and the programme needs small modifications for obtaining incremental costs of adjustable parameters. Main disadvantages of the method are (i) the programming logic is considerably more complex (ii) large memory requirement even if compact storage is used. Credit for making this technique practical goes to Tinney[33] and his associates of B.P.A.

(d) Non-linear programming methods: Load flow problem is treated as the problem of optimization where the objective function is the sum of squares of the total real and reactive mismatch. Optimization is achieved through any of the gradient techniques. The author's experience with gradient techniques is that an accurate solution is hard to obtain. Fletchers Powell's gradient method[44] may yield a better convergence but needs a very large storage and time for the non-sparse Hessian matrix which is built iteratively. Chief advantages of this method appear to be only small modification needed for optimal power flow solution and less computer logic compared to Newton's method. Credit for perfecting this method goes to Sasson[4,15,16].

Most of the research work in the field of load flow has been concerned with variation on these methods, which include better correction and acceleration methods, better ways to utilize the matrix by tearing, triangular factorization, compact storage, near optimal renumbering and various methods of representing the system as a d.c. circuit and dealing only with real power flows to obtain approximate solution.

5.3 Convergence of Gauss and Gauss Seidal iterative techniques for linear simultaneous equations

If n x n linear simultaneous equation under consideration is as follows:

$$Ax = b \tag{5.1}$$

Let $$A = D + L + U \tag{5.2}$$

where D, L and U are n x n matrices. D contain the diagonal elements, L the elements on the left side of the diagonal and U the elements on the right hand side of diagonal.

Formulation of Gauss iterative procedure is as follows:

$$D\,x^{k+1} = b - (L + U)\,x^{k} \tag{5.3}$$

if $$x^{k} = x^{*} + \varepsilon\eta \tag{5.4}$$

where η is n x 1 column vector and x^{*} is the solution of equation 5.1

$$D\,x^{k+1} = A\,x^{*} - (L + U)\,(x^{*} + \varepsilon\eta)$$

$$= D\,x^{*} - \varepsilon(L + U)\,\eta$$

$$x^{k+1} = x^{*} - \varepsilon\,D^{-1}(L + U)\,\eta \tag{5.5}$$

If absolute value of any eigen value of $D^{-1}(L + U)$ is more than 1, the Gauss iterative method will not converge, does not matter how close the initial guess is to the actual solution.

Formulation of Gauss Seidal iterative technique is as follows:

$$x^{k+1} = x^{\circ} - \varepsilon(L + D)^{-1} U \eta \qquad (5.6)$$

Necessary and sufficient condition for the technique to converge is that the absolute value of no eigen value of $(L + D)^{-1} U$ be greater than 1.

Convergence may sometimes be improved by the application of acceleration or damping, as follows:

$$x^{k+1} \text{ (modified)} = x^{k} + \alpha(x^{k+1} - x^{k}) \qquad (5.7)$$

where $\alpha > 0$

$\alpha > 1$ represents acceleration and $\alpha < 1$ represent damping.

$$x^{k} = x^{\circ} + \varepsilon \eta$$

$$x^{k+1} = x^{\circ} - \varepsilon B \eta \qquad (5.8)$$

where B is the principal matrix and is as follows:

$B = D^{-1}(L + U)$ for Gauss method

and $B = (L + D)^{-1} U$ for Gauss Seidal method

$$x^{k+1} \text{ (modified)} = x^{\circ} + \varepsilon[(1-\alpha) I - \alpha B]\eta \qquad (5.9)$$

where I is the identity matrix.

For the modified solution to converge:

$$|1 - \alpha(1 + \beta)| < 1 \tag{5.10}$$

where β is an eigen value of B.

Speed of convergence for unaccelerated solution depends upon β and for the accelerated solution upon $|1 - \alpha(1 + \beta)|$.

If $\beta > 1$, unaccelerated solution will not converge, but the damped solution satisfying equation (5.10) will converge.

If $\beta < -1$, no acceleration or damping can make it converge.

If $\beta = -1 + \delta$; for small positive δ ; convergence is poor but the accelerated solution will have better convergence.

5.4 Convergence of iteration process for non-linear functions[37,40]

If the equations are written in the following form,

$$x_i = F_i(x) \; ; \; i = 1, \ldots, m \tag{5.11}$$

where $x = (x_1, x_2, \ldots, x_m)^t$

First iteration is as follows:

$$x_i^2 = F_i(x^1) \tag{5.12}$$

$$x_i^* - x_i^2 = F_i(x^*) - F_i(x^1)$$

$$= \int_0^1 \frac{\partial}{\partial\theta}\left[F_i(x^1 + \theta(x^*-x^1))\right] d\theta$$

$$= \int_0^1 \sum_j \frac{\partial F_i}{\partial x_j}(x^1 + \theta(x^*-x^1))(x_j^* - x_j^1)\, d\theta$$

$$= \sum_j (x_j^* - x_j^1) \int_0^1 \frac{\partial F_i}{\partial x_j}(x^1 + \theta(x^*-x^1))\, d\theta$$

Let $$g_{ij}(x^*, x^1) = \int_0^1 \frac{\partial F_i}{\partial x_j}(x^1 + \theta(x^*-x^1))\, d\theta \tag{5.13}$$

$$x_i^* - x_i^2 = \sum_j (x_j^* - x_j^1)\, g_{ij} \tag{5.14}$$

Adding equation 5.14;

$$\sum_i (x_i^* - x_i^2) = \sum_i \sum_j (x_j^* - x_j^1)\, g_{ij} \tag{5.15}$$

$$= \sum_j (x_j^* - x_j^1) \sum_i g_{ij} \tag{5.16}$$

Considering only the absolute values

$$\sum_j |x_j^* - x_j^2| \leq \sum_j |x_j^* - x_j^1| \sum_i |g_{ij}| \tag{5.17}$$

If $\sum_i \left|\frac{\partial F_i}{\partial x_j}\right| \leq \delta < 1$ for all i in the region

(x^1, x^*)

$$\sum_j |x_j^o - x_j^2| \leq \delta \sum_j |x_j^o - x_j^1| \tag{5.18}$$

This relation holds good for the first iteration. For successive iterations; similarly

$$\sum_j |x_j^o - x_j^{k+1}| \leq \delta \sum_j |x_j^o - x_j^k|$$

$$\sum_j |x_j^o - x_j^{k+1}| \leq \delta^k \sum_j |x_j^o - x_j^1| \tag{5.19}$$

If δ is less than 1; right hand side of this inequality can be made as small as desirable, by repeating the iteration process sufficient number of times.

Thus the iteration process converges if

$$\sum_j \left|\frac{\partial F_j}{\partial x_i}\right| < 1 \tag{5.20}$$

for all i.

5.5 Uniqueness of solution

If x^* and y satisfy equation 5.13;

since $y_i = F_i(y)$

$$\sum_j |x_j^o - F_j(y)| \leq \delta \sum_j |x_j^o - y_j|$$

or $$\sum_j |x_j^o - y_j| \leq \delta \sum_j |x_j^o - y_j|$$

This is a contradiction since $\delta < 1$.

5.6 Convergence of admittance matrix method using Gauss iterative technique

$$V_i = \frac{1}{y_{ii}} \left[\frac{P_i - jQ_i}{V_i^*} - \sum_{\substack{j \\ j \neq i}} y_{ij} V_j \right] \tag{5.21}$$

for all i except slack bus.

Let $g_{ij} + j\, b_{ij} = y_{ij}/y_{ii}$ (5.22)

$c_i + j\, d_i = (P_i - j\, Q_i)/y_{ii}$ (5.23)

$e_i + j\, f_i = V_i$ (5.24)

$$e_i = \frac{e_i c_i - f_i d_i}{e_i^2 + f_i^2} - \sum_{\substack{j \\ j \neq i}} (g_{ij}\, e_j - b_{ij}\, f_j) \tag{5.25}$$

$$f_i = \frac{f_i c_i + e_i d_i}{e_i^2 + f_i^2} - \sum_{\substack{j \\ j \neq i}} (b_{ij}\, e_j + g_{ij}\, f_j) \tag{5.26}$$

Criteria of convergence are

$$\delta_e = \sum_{\substack{i \\ i \neq s}} \left|\frac{\partial E_i}{\partial e_k}\right| + \sum_{\substack{i \\ i \neq s}} \left|\frac{\partial F_i}{\partial e_k}\right| < 1; \quad k \neq s \tag{5.27}$$

$$\delta_f = \sum_{\substack{i \\ i \neq s}} \left|\frac{\partial E_i}{\partial f_k}\right| + \sum_{\substack{i \\ i \neq s}} \left|\frac{\partial F_i}{\partial f_k}\right| < 1; \quad k \neq s \tag{5.28}$$

where E_i and F_i are the right hand side of equations 5.25 and 5.26.

$$\delta_e = \sum_{\substack{i \\ i \neq k,s}} \left\{ |g_{ik}| + |b_{ik}| \right\} \cdot \frac{\left| f_k^2 c_k - e_k^2 c_k + 2e_k f_k d_k \right| + \left| -e_k^2 d_k + f_k^2 d_k - 2e_k f_k d_k \right|}{(e_k^2 + f_k^2)^2} \qquad (5.29)$$

$$\delta_f = \sum_{\substack{i \\ i \neq k,s}} \left\{ |b_{ik}| + |g_{ik}| \right\} \cdot \frac{\left| f_k^2 d_k - e_k^2 d_k - 2f_k e_k c_k \right| + \left| e_k^2 c_k - f_k^2 c_k - 2e_k f_k d_k \right|}{(e_k^2 + f_k^2)^2} \qquad (5.30)$$

For a reliable convergence δ_e and δ_f should be sufficiently less than 1. However as we know

$$- \sum_j y_{ij} + y_i' = y_{ii} \qquad (5.31)$$

where y_i' is the equivalent shunt admittance at i node. this suggest

$$\left| \sum_{\substack{j \\ j \neq i}} \frac{y_{ij}}{y_{ii}} \right|$$

is nearly equal to 1.

Hence $\sum_{\substack{k \\ k \neq i}} |b_{ik}| + |g_{ik}|$ and subsequently

$$\sum_{\substack{i \\ i \neq k,s}} |b_{ik}| + |g_{ik}|$$

may be on either side of 1. Over and above it, there will be contribution due to additional term depending on system loading. This means that the sufficiency condition for convergence is either not satisfied or satisfied by a low margin. This explains for poor and less reliable convergence of the method.

5.7 Convergence of impedence matrix method with Gauss iterative technique

The fixed point formulation is as follows:

$$I_i = \frac{P_i - j\,Q_i}{V_i^*} - y_i\,V_i \tag{5.32}$$

$$V_i = V_s + \sum_{\substack{j \\ j \neq s}} z_{ij}\,I_j \tag{5.33}$$

These equations may also be written as follows:

$$V_i = V_s + \sum_{\substack{j \\ j \neq s}} z_{ij}\left[\frac{P_j - j\,Q_j}{(V_j)^*} - y_j\,V_j\right] \tag{5.34}$$

for all i except slack.

If $$r_{ij} + j\,x_{ij} = z_{ij} \tag{5.35}$$

$$e_j + j\,f_j = V_j \tag{5.36}$$

$$g_{ij} + j\,b_{ij} = z_{ij}\,y_j \tag{5.37}$$

$$e_i = e_s + \sum_{\substack{j \\ j \neq s}} \frac{e_j(r_{ij}P_j + x_{ij}Q_j) + f_j(x_{ij}P_j - r_{ij}Q_j)}{(e_j^2 + f_j^2)}$$

$$- \sum_{\substack{j \\ j \neq s}} (r_{ij}\, e_j - b_{ij}\, f_j) \tag{5.38}$$

$$f_i = f_s + \sum_{\substack{j \\ j \neq s}} \frac{e_j(x_{ij}P_j - r_{ij}Q_j) - f_j(r_{ij}P_j + x_{ij}Q_j)}{(e_j^2 + f_j^2)}$$

$$- \sum_{\substack{j \\ j \neq s}} (b_{ij}\, e_j + r_{ij}\, f_j) \tag{5.39}$$

Let

$$a'_{ik} = r_{ik}\, P_k + x_{ik}\, Q_k \tag{5.40}$$

$$b'_{ik} = x_{ik}\, P_k - r_{ik}\, Q_k \tag{5.41}$$

$$c_{ik} = e_k\, a'_{ik} + f_k\, b'_{ik} \tag{5.42}$$

$$d_{ik} = e_k\, b'_{ik} - f_k\, a'_{ik} \tag{5.43}$$

$$\varepsilon_e = \sum_{\substack{i \\ i \neq s}} \left| \frac{(e_k^2 + f_k^2)\, a'_{ik} - 2e_k\, c_{ik}}{(e_k^2 + f_k^2)^2} - g_{ik} \right|$$

$$+\sum_{\substack{i \\ i\neq s}} \left| \frac{(e_k^2 + f_k^2)\, b'_{ik} - 2e_k\, d_{ik}}{(e_k^2 + f_k^2)^2} + b_{ik} \right| \tag{5.44}$$

$$\delta_f = \sum_{\substack{i \\ i\neq s}} \left| \frac{(e_k^2 + f_k^2)\, b'_{ik} - 2f_k\, c_{ik}}{(e_k^2 + f_k^2)^2} + b_{ik} \right|$$

$$+\sum_{\substack{i \\ i\neq s}} \left| \frac{-(e_k^2 + f_k^2)\, e'_{ik} - 2\, f_k\, d_{ik}}{(e_k^2 + f_k^2)^2} - g_{ik} \right| \tag{5.45}$$

g_{ik} and b_{ik} are small quantities due to small shunt admittance at node k. Other elements in δ_e and δ_f are proportionate to loading. If loading is small, δ_e and δ_f will be less than 1. This assures the convergence. This explains for better convergence of Z-bus method over Y-bus method.

5.8 Convergence of Newton's method[38]

Consider a set of n equations with n unknowns

$$f_i(x_1, x_2, \ldots, x_n) = 0$$

The fixed point formulation through Newton's method is as follows:

$$x = x - J^{-1}(x)\, f(x) = \phi(x) \tag{5.46}$$

where $J(x) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots\cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \\ \frac{\partial f_n}{\partial x_1} & \cdots\cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$ (5.47)

Because of the presence of J(x); analysis through 5.4 will need a three dimensional Hessian matrix; explanation of resulting expression is hard to understand. Kantorovich's[40] analysis, attempted by Meisel and Barnard[3] also has the same drawback. A simplified analysis is as follows.

$$x^{k+1} - x^* = x^k - J^{-1}(x^k)\, f(x^k) - x^*$$

$$= x^k - x^* - J^{-1}(x^k)\left[f(x^k) - f(x^*)\right] \qquad (5.48)$$

where x^* is solution of the equation

$$f(x^k) - f(x^*) = G(x^*, x^k)\,(x^k - x^*) \qquad (5.49)$$

where G is n x n matrix whose ijth element is given by equation 5.13

$$x^{k+1} - x^* = x^k - x^* - J^{-1}(x^k)\, G(x^*, x^k)\,(x^k - x^*)$$

$$= \left\{ I - J^{-1}(x^k)\, G(x^*, x^k) \right\} (x^k - x^*) \qquad (5.50)$$

Since

$$J^{-1}(x^*)\, G(x^*, x^*) = I \qquad (5.51)$$

If x^1 is close enough to x^*, matrix $J^{-1}(x^1)G(x^*,x^1)$ is quite close to I and $I-J^{-1}(x^1)G(x^*,x^1)$ is quite close to zero.

If the norm of a column vector y is defined as follows

$$\|y\| = \sum_j |y_j| \tag{5.52}$$

and norm of matrix A as follows

$$\|A\| = \max_j \sum_i |a_{ij}| \tag{5.53}$$

$$\|Ay\| = \sum_i \left|\sum_j a_{ij}\, y_j\right| \leq \sum_i \sum_j |a_{ij}|\, |y_j|$$

$$\leq \sum_j |y_j| \quad \max_j \sum_i |a_{ij}|$$

$$\leq \|y\|\, \|A\| \tag{5.54}$$

Thus

$$\|x^2 - x^*\| \leq \|I - J^{-1}(x^1)\, G(x^*,x^1)\| \cdot \|x^1 - x^*\|$$

Since norm of a zero matrix is zero and since

$I - J^{-1}(x^1)\, G(x^*, x^1)$ is quite close to zero, there therefore exists $\varepsilon > 0$ such that if $\|x - x^*\| \leq \varepsilon$

Then

$$\|I - J^{-1}(x)\, G(x^*, x)\| \leq \delta < 1$$

If $\|x^1 - x^*\| \leq \varepsilon$ then

$$\|x^2 - x^*\| \leq \delta \|x^1 - x^*\| \leq \delta\varepsilon < \varepsilon \tag{5.55}$$

Therefore,

$$\|x^{k+1} - x^*\| \leq \delta^k \|x^1 - x^*\|$$

and $$\lim_{k \to \infty} \|x^{k+1} - x^*\| = 0 \tag{5.56}$$

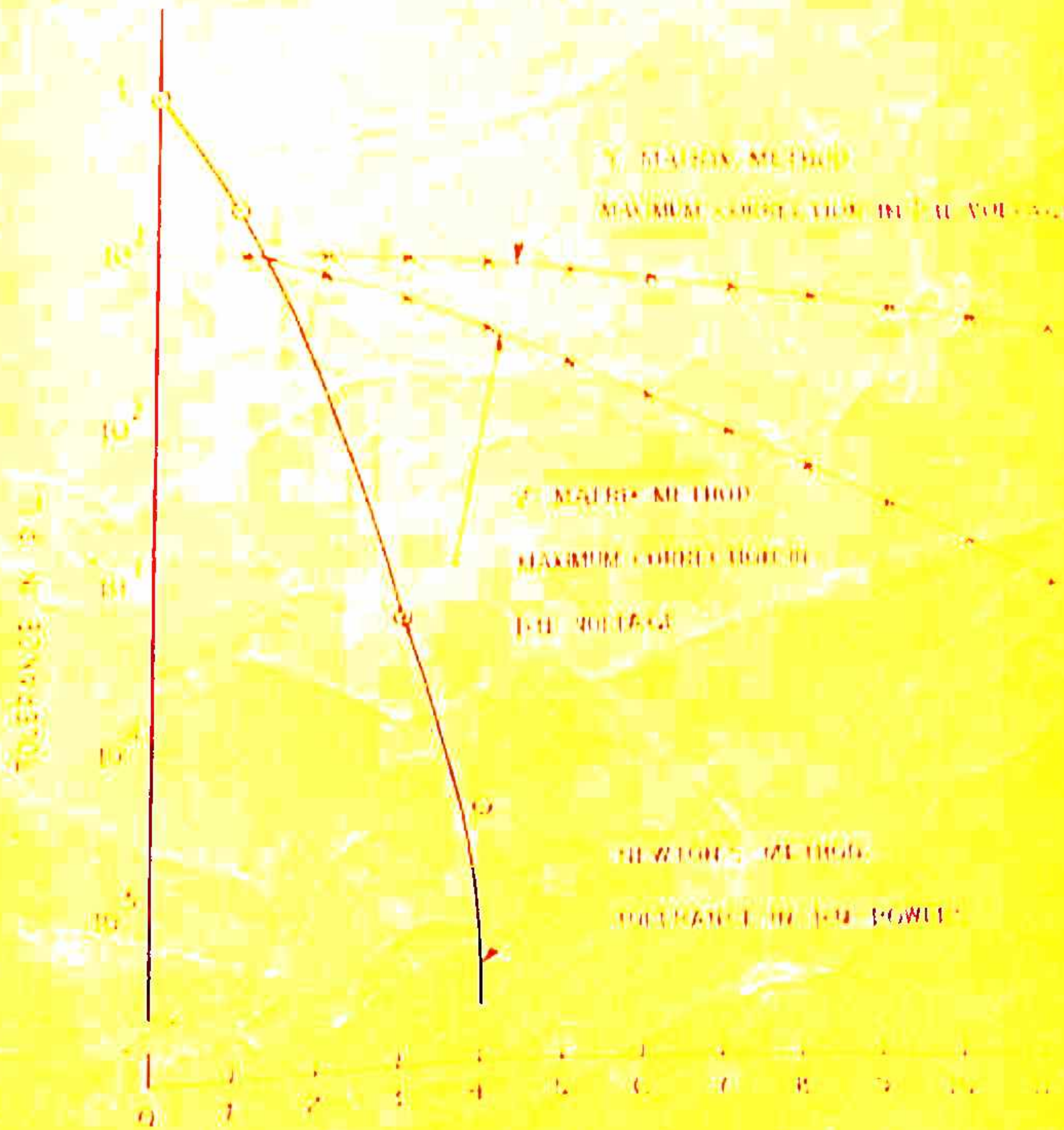
5.9 Example

The value of δ obtained on AEP 30 bus system for Y-matrix method was 1.126. This indicates doubtful convergence. However it converged to a voltage tolerance of 0.001 pu in 44 iterations with the initial voltages of 1 + jo for all busses except 1.06 + jo for the slack bus.

The same value for Z-matrix method was 0.869. This indicates sure convergence. The solution converged in 11 iterations within the tolerance indicated above.

For Newton's method the value of δ and maximum mismatch for successive iterations was as follows:

Iteration	δ	Maximum mismatch
Initial start	.3863	.94199
1	.1254	.20980
2	.0327	$.26968 \times 10^{-1}$
3	.0043	$.73812 \times 10^{-4}$
4	.0002	$.53589 \times 10^{-5}$



CONVERGENCE CHARACTERISTICS OF LOAD FLOW METHODS

FIG. [illegible]

Maximum mismatch of fourth iteration is not considered reliable since the word size of IBM 1130 computer with standard precision is about 6.9 decimal digits. This should be much lower as judged from the trend and also from the corresponding δ.

5.10 Conclusion

Hence following conclusions are drawn through fixed point principles to which the experimental results also agree.

(1) Convergence of Y-matrix method is less reliable even if the initial point is close enough and loading is small.

(2) Convergence of Z-matrix method is more reliable if the loading and shunt admittance is small.

(3) Convergence of Newton's method is very reliable if the initial point is sufficiently close to the final solution.

It has been experimentally established in literature that the convergence of Newton's method in polar form is most reliable and fast. Convergence characteristics of load flow methods for AEP 30 bus system is as given in Fig. 5.1. A similar characteristics is also given by Dommel in correspondence for reference 33 for a 17 bus system.

-x-

CHAPTER VI

COMPUTER PROGRAMME FOR LOAD FLOW SOLUTION BY NEWTON'S METHOD

In this chapter the load flow programme written by the author is discussed. It has not been possible for us to compare the computational time per iteration with that of BPA programme because of availability of a small computer and non-availability of BPA system test data to us. Time per iteration for AEP 30 bus system is about 10 seconds.

The programme is based on formulating and triangularising the equation 6.1 node by node. This equation is not stored in the computer. The notation followed is that of Vanness and Griffin[50] given in Chapter IV. If incremental costs of maintaining the variables to the prescribed value need not be computed directly, L_{is} and N_{is} for $i \neq s$ are replaced by zeros. Similarly for decidedly pv node numbered k; L_{ik} and N_{ik} are also replaced by zero. Since the programme has been used for optimal power flow studies, this has not been done since incremental cost of maintaining the voltage to a particular value also need be computed.

6.1 Changing bus bar type

In addition to the conventional buses of p.q, p.v and slack; the computer programme provides for the following additional types of buses:

$$
\begin{array}{r}
\\ \\ \\
\text{Slack bus (s)} \\ \\
\\ \\ \\
\text{PV bus (k)} \\ \\
\\ \\ \\
\end{array}
\begin{bmatrix}
H_{11} & N_{11} & \cdots & 0 & N_{1s} & \cdots & H_{1k} & N_{1k} & \cdots & H_{1n} & N_{1n} \\
J_{11} & L_{11} & \cdots & 0 & L_{1s} & \cdots & J_{1k} & L_{1k} & \cdots & J_{1n} & L_{1n} \\
\vdots & \vdots & & \vdots & \vdots & & & & & & \\
0 & 0 & & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\
0 & 0 & & 0 & 1 & 0 & 0 & 0 & & 0 & 0 \\
\vdots & \vdots & & \vdots & & & & & & & \\
H_{k1} & N_{k1} & \cdots & 0 & N_{ks} & \cdots & H_{kk} & N_{kk} & \cdots & H_{kn} & N_{kn} \\
0 & 0 \cdots & 0 & 0 & 0 & 0 \cdots 0 & 0 & 1 & 0 \cdots & 0 & 0 \\
\vdots & \vdots & & \cdot & \cdot & & \cdot & \cdot & & \cdot & \cdot \\
H_{n1} & N_{n1} & \cdots & 0 & N_{ns} & \cdots & H_{nk} & N_{nk} & \cdots & H_{nn} & N_{nn} \\
J_{n1} & L_{n1} & \cdots & 0 & L_{ns} & \cdots & J_{nk} & L_{nk} & \cdots & J_{nn} & L_{nn}
\end{bmatrix}
\begin{bmatrix}
\Delta e_1 \\ \Delta V_1/V_1 \\ \vdots \\ \vdots \\ \vdots \\ \Delta e_i \\ \Delta V_i/V_i \\ \vdots \\ \vdots \\ \Delta e_n \\ \Delta V_n/V_n
\end{bmatrix}
=
\begin{bmatrix}
\Delta P_1 \\ \Delta Q_1 \\ \vdots \\ 0 \\ 0 \\ \vdots \\ \Delta P_k \\ \Delta V_k/V_k \\ \vdots \\ \Delta P_n \\ \Delta Q_n
\end{bmatrix}
\qquad (6.1)
$$

$$\qquad\qquad\qquad\text{Slack (bus)} \qquad\qquad \text{PV bus}$$

(This equation is never stored)

(i) Bus bar with assigned reactive power provided the bus voltage is within the prescribed upper and lower limits.

(ii) Bus bar with assigned voltage provided the reactive power is within the prescribed upper and lower limits.

This has been done as follows for kth bus in each iteration:

Obtain L_{kj} for $j \neq k$ from equations 4.15, 4.17, 4.18 and 4.20.

$$Q_k = -V_k^2 b_{kk} + \sum_{j \neq k} L_{kj} \qquad (6.2)$$

$$L_{kk} = Q_k - V_k^2 b_{kk} \qquad (6.3)$$

If the bus is with assigned voltage magnitude, desired change in voltage ΔV_k is obtained.

With this change in voltage, expected change in Q_k is approximately as follows:

$$Q_k \cong (\Delta V_k L_{kk})/V_k \qquad (6.4)$$

$(Q_k + \Delta Q_k)$ is now checked if this is within the prescribed bounds, the bus is treated as pv bus; if not it is treated as p q bus with prescribed reactive power equal to the upper or lower limit, depending upon which limit is violated for that iteration.

If kth bus is with prescribed reactive power, ΔQ_k is obtained from the prescribed reactive power and the calculated reactive power.

Approximate value of ΔV_k is obtained as follows:

$$\Delta V_k = (V_k \; \Delta Q_k)/L_{kk} \tag{6.5}$$

$(V_k + \Delta V_k)$ is now checked for the prescribed limits of bus bar voltage. If the limit is violated, the bus is treated as p v bus with prescribed voltage equal to the upper or lower limit, depending upon which limit is violated, otherwise it is a p q bus for that iteration.

Obtaining L_{kj} does not mean additional computation time for p v bus since H_{kj} will have to be computed and $L_{kj} = H_{kj}$; $j \neq k$.

Convergence with the above method has been found very satisfactory.

6.2 Formation of working row

In order to reduce the dimensionality of the problem, following notation has been used.

If the new number of the bus is i and its external number is i' ; for ith node, if it a p q ;

$$\Delta g_{2i-1} = \Delta P_{i'} \tag{6.6}$$

$$\Delta s_{2i} = \Delta Q_{i'} \qquad (6.7)$$

(before triangularisation)

and

$$\Delta s_{2i-1} = \Delta \theta_{i'} \qquad (6.8)$$

$$\Delta s_{2i} = \Delta V_{i'}/V_{i'} \qquad (6.9)$$

(after triangularisation)

$$a^1_{2j-1} = H_{i'j'} = sg_{i'j'} - cb_{i'j'} \qquad (6.10)$$

$$a^1_{2j} = N_{i'j'} = cg_{i'j'} + sb_{i'j'} \qquad (6.11)$$

where $s = V_{i'} V_{j'} \sin(\theta_{i'} - \theta_{j'})$

$$= f_{i'} e_{j'} - e_{i'} f_{j'} \qquad (6.12)$$

$c = V_{i'} V_{j'} \cos(\theta_{i'} - \theta_{j'})$

$$= e_{i'} e_{j'} + f_{i'} f_{j'} \qquad (6.13)$$

$$a^2_{2j-1} = J_{i'j'} = -a^1_{2j} \qquad (6.14)$$

$$a^2_{2j} = L_{i'j'} = a^1_{2j-1} \qquad (6.15)$$

Equations 6.10 to 6.15 are valid for all j except $j \neq i$.

$$a^1_{2i-1} = -\sum_{j \neq i} a^1_{2j-1} = H_{i'i'} \qquad (6.16)$$

$$a^2_{2i-1} = \sum_{j \neq i} a^1_{2j} = J_{i'i'} \qquad (6.17)$$

$$P_{i'} = V_{i'}^2\, g_{i'i'} + a^2_{2i-1} \qquad (6.18)$$

$$Q_{i'} = -V_{i'}^2\, b_{i'i'} - a^1_{2i-1} \qquad (6.19)$$

$$a^1_{2i} = V_{i'}^2\, g_{i'i'} + P_{i'} \qquad (6.20)$$

$$a^2_{2i} = -V_{i'}^2\, b_{i'i'} + Q_{i'} \qquad (6.21)$$

These elements are obtained without making use of trigometrical functions.

$H_{i's}$ and $J_{i's}$ are now replaced by zero.

New numbers are used only in forming and triangularising equation 6.1, while admittance matrix, voltages and powers are recognised by the external numbers. However a little time per iteration can be saved by reforming the admittance matrix with the new numbers and recognising all the bus bar quantities (voltages, powers, etc.) with the new numbers.

For a p v bus; whether predecided or decided during iteration; equation 6.7 is replaced by the following equation:

$$a_{2i} = V_{i'}/V_{i'} \qquad (6.22)$$

the second row for that node is as follows:

$$a^2_{2i} = 1 \qquad (6.23)$$

$$a^2_j \; ; \; j = 1, \ldots, 2n; \; j \neq 2i = 0 \qquad (6.24)$$

This working now represented by equations 6.10 to 6.17; 6.20 and 6.21 is not represented explicitly. The total number of non-zero elements, column location of each element and the numerical value of these elements are stored instead. This can be explained as follows.

Suppose the node 4 under consideration has non-zero element in the admittance matrix at column 2, 5 and 8.

The working row formed are as follows.

Table 6.1

S.No.	1	2	3	4	5	6	7	8
Non-zero element of first working row	a_3^1	a_4^1	a_7^1	a_8^1	a_9^1	a_{10}^1	a_{15}^1	a_{16}^1
Column location	3	4	7	8	9	10	15	16

Table 6.2

S.No.	1	2	3	4	5	6	7	8
Non-zero element of second working row	a_3^2	a_4^2	a_7^2	a_8^2	a_9^2	a_{10}^2	a_{15}^2	a_{16}^2
Column location	3	4	7	8	9	10	15	16

Non-zero elements in the first working row $m_1 = 8$

Non-zero elements in the second working row $m_2 = 8$

Table 6.2 need not be prepared if it is a pv node. If it is a slack node none of these rows need be prepared.

6.3 Triangularization

After the working rows are formed; the first working row is triangularised with the help of table of factors. For this fourth node under consideration; the first column of non-zero element is 3. Assuming that the non-zero elements in the third row of table of factors are in column 5, 6, 9 and 10; the working is now modified as follows.

S.No.	1	2	3	4	5	6	7	8	9	10
Non-zero element	a'_3	a'_4	a'_5	a'_6	a'_7	a'_8	a'_9	a'_{10}	a'_{15}	a'_{16}
Column	3	4	5	6	7	8	9	10	15	16

Non-zero elements $m_1 = 10$

Further modifications will be made to this working row while this is triangularised with row 4, 5, 6, etc. For this example no new non-zero elements will be added during triangularisation with the 4th row.

After the triangularisation is complete; contents of this array are transferred to the table of factors. Contents of second working row (Table 6.2) are now transferred to the first working row, and the triangularisation is repeated for the 8th row corresponding to the node 4.

This process is repeated for every node. Corrections in voltage magnitude and angles are obtained through backward substitution. The iteration is repeated till the maximum mismatch is less than the prescribed error. If maximum mismatch of one iteration is found more than that of previous iteration, divergence in the solution has been assumed. The iteration may then be repeated with modified generation and tap settings. In the programme maximum mismatch is obtained only after calculation of jacobian elements and forward triangularisation is complete. Since jacobian elements are obtained while computing the calculated powers; it was considered wasteful to check for mismatch before the triangularisation starts.

Non-zero left subdiagonal elements need not be stored if only load flow solution is needed, without skipping the formation of jacobian during any iteration. However as will be seen later; the effect of $(A^t)^{-1}$ is also to be obtained, for which the left subdiagonal elements are also needed. Compact storage scheme of table 2.2.3 has been used for the table of factors.

The load flow solution can be made even faster if it does not have a bus with indefinite mode of operation as explained in section 6.1, by adopting the storage scheme of table 2.2.4 and exploiting the symmetrical pattern of non-zero elements of the jacobian, which are obtained as follows:

$$H_{ij} = sg_{ij} - cb_{ij}$$

$$H_{ji} = -sg_{ij} - cb_{ij}$$

$$N_{ij} = cg_{ij} + sb_{ij}$$

$$N_{ji} = cg_{ij} - sb_{ij}$$

This way computation time of obtaining jacobian elements can be reduced practically to half, but triangularization time remains the same. However this scheme needs storage for the elements on the left of diagonal even if only load flow solution is needed.

-x-

CHAPTER VII

NONLINEAR PROGRAMMING

In this chapter description of the nonlinear programming problem, criteria and techniques of identifying the optimal point are discussed. Initially the discussion is restricted to the unconstrained problem of minimising the scaler f(x). Effect of constraint is included later on.

7.1 Description of the problem

f(x) is a scaler function of n-dimensional vector x. Subject to the following constraints;

$$h(x) = 0 \qquad (7.1)$$

and

$$g(x) \leq 0 \qquad (7.2)$$

a minimum of f(x) is to be obtained.

Where g(x) and h(x) are vector functions of x.

7.2 The unconstrained solution

If f(x) is a twice continuously differential function on E^n; then

$$\nabla f(x^*) = 0 \qquad (7.3)$$

if x^* minimises f over E^n and the quadratic form

$$\sum_i \sum_j \frac{\partial^2 f}{\partial x_i \partial x_j} a_i a_j \geq 0 \qquad (7.4)$$

for every point x in the neighbourhood of x^*.

The techniques for finding the minimum (if it exists) may be based upon solving the set of simultaneous equations 7.3. Solution may be obtained through iterative techniques such as Newton's, Gauss or Gauss-Seidal methods if the equations are nonlinear. In order that such techniques work, it is necessary that the guess is such that f(x) is convex and remains in convex domain from iteration to iteration.

7.3 Second order gradient method

The second order gradient method is based upon obtaining the solution of equation 7.3 through Newton's method. Fixed point formulation is as follows:

$$x = x - (f_{xx}(x))^{-1} \nabla f(x) \qquad (7.5)$$

In order that the algorithm converges to the minimum, Hessian matrix $f_{xx}(x)$ must be positive definite and norm of $(I - f_{xx}^{-1}(x^*)\, f_{xx}(x)\ )$ should be less than 1 in the $\mathcal{E}$ neighbourhood of x^* in which the initial point lies.

7.4 First order gradient methods

Assuming we are at a point x in E^n and we wish to move along some direction vector d also in E^n a certain distance, r. The objective is to successfully approach the minimum. In order to achieve rapid convergence the direction d and distance r must be properly decided.

$$f(x + rd) = f(x) + r \sum_i \frac{\partial f}{\partial x_i} d_i + \frac{r^2}{2} \sum_i \sum_j \frac{\partial^2 f}{\partial x_i \partial x_j} d_i d_j \qquad (7.6)$$

If x is the previous iteration point and we wish to compute the new iteration point x+rd which is nearer to the minimum, we must then have

$$f(x+rd) - f(x) < 0 \qquad (7.7)$$

Since the coefficient of r^2 in equation 7.6 is positive semidefinite if f(x) is convex; coefficient of r should be negative or d is such that

$$\sum_i \frac{\partial f}{\partial x_i} d_i < 0 \qquad (7.8)$$

Optimal value of r may be obtained by equating the partial derivative of f(x+rd) with respect to r to zero or

$$r^\circ = - \frac{\nabla f^t d}{d^t f_{xx} d} \qquad (7.9)$$

7.5 Steepest descent method[41]

In this method

$$d = - \nabla f(x)$$

Thus equation 7.8 is satisfied. r is obtained such that $f(x+rd)$ is minimum along the direction d. If $f(x)$ is quadratic, it may be obtained from equation 7.9.

For a quadratic convex function,

$$f(x) = a + b^t x + \frac{1}{2} x^t C\, x \tag{7.10}$$

It is proved in reference (44) that

$$(s^k)^t\,(s^k) \leq \frac{2}{m}\,(1 - m/M)^k\,(f(x^0) - f(x^*)\,) \tag{7.11}$$

where x^0 is the initial point and

$$s^k = x^k - x^0$$

m and M are respectively the minimum and maximum of non-zero eigen values of C. Since the function is convex, C is positive semidefinite, which implies that each eigen value is also positive semidefinite.

Thus,

$$\lim_{k \to \infty} x^k = x^* \tag{7.12}$$

Another important conclusion drawn by author is that a function with larger m/M will have a better convergence. One method to improve this ratio is scaling. This may be illustrated with the help of following example,

$$f(x) = x_1^2 + 4\,x_2^2$$

Hessian matrix for this example is

$$\begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix}$$

for which $m/M = .25$.

If the initial guess is $(1,1)^t$, the direction vector is $(-2, -8)^t$ for which

$$r^0 = .1308$$

$$x^1 = (.7384, -.0464)^t$$

$$r^1 = .426$$

$$x^2 = (.110, .1116)^t$$

If the variables are scaled as follows

$$y_1 = x_1$$

$$y_2 = 2x_2$$

$$f(x) = f(y) = y_1^2 + y_2^2$$

for which the corresponding initial point is $(1, 2)^t$, the direction vector $(-2, -4)^t$, Hessian matrix is

$$\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

for which $m/M = 1$

$$r^k = .5$$

$$x^1 = (0, 0)^t$$

which is obviously the optimal point.

7.6 Conjugate gradient method[43]

While convergence of steepest descent method depends upon m/M; the conjugate gradient method guarantees convergence to the exact solution, disregarding the round off errors, in n iterations for a quadratic function of equation 7.10. In this method, a set of linearly independent directors d^o, d^1, d^2, ..., d^{n-1} are generated such that these are C conjugate ($d_i^t\ C\ d_j = 0$ if $i \neq j$) and the iteration is given by

$$x^{k+1} = x^k + r^k\ d^k \tag{7.13}$$

where r^k is for minimum of $f(x^k + rd^k)$.

The algorithm as given in reference (43) is as follows:

$$d^k = -\nabla f(x^k) + \alpha^k\ d^{k-1} \tag{7.14}$$

where
$$\alpha^k = \frac{\nabla f^t(x^k)\ \nabla f(x^k)}{\nabla f^t(x^{k-1})\ \nabla f(x^{k-1})} \tag{7.15}$$

$$d^o = -\nabla f(x^o) \tag{7.16}$$

x^{k+1} is obtained from equation 7.13.

This may be explained for the example of (7.5) as follows:

$$f(x) = x_1^2 + 4x_2^2$$

$$x^0 = (1, 1)^t$$

$$\nabla f(x^0) = (2, 8)^t$$

$$d^0 = (-2, -8)^t$$

$$r^0 = .1308$$

$$x^1 = (.7384, -.0464)^t$$

$$\nabla f(x^1) = (1.4768, -.3712)^t$$

$$\alpha = .0343$$

$$d^1 = (-1.5454, +.0968)^t$$

$$r^1 = .478$$

$$x^2 = (0, 0)^t$$

7.7 Fletcher and Powell's method[44]

This gradient method has been found more powerful than other first order gradient methods, but need additional computer time and memory for the non-sparse inverse of Hessian, which is not evaluated directly as is done in the second order gradient method. Instead it is built iteratively. For a quadratic function of equation 7.10, it is shown to converge in n iterations as is the case with conjugate gradient method.

The algorithm is as follows:

$$d^k = -H^k \nabla f(x^k) \tag{7.17}$$

where H^k is the positive semidefinite symmetrical matrix, H^o may be chosen as identity matrix.

$$x^{k+1} = x^k + r^k d^k \tag{7.18}$$

where the scaler r^k is chosen to minimise $f(x^k+rd^k)$.

$$H^{k+1} = H^k + A^k + B^k \tag{7.19}$$

where A^k and B^k are square symmetrical matrices as follows:

$$A^k = \frac{r(d^k)\,(d^k)^t}{(d^k)^t(y^k)} \tag{7.20}$$

$$B^k = -\frac{(H^k y^k)\,(H^k y^k)^t}{(y^k)^t H^k y^k} \tag{7.21}$$

$$y^k = \nabla f(x^{k+1}) - \nabla f(x^k) \tag{7.22}$$

7.8 Single dimensional minimisation[43,44]

The first order gradient methods mentioned above need the step length r along the direction d^k which minimises

$f(x^k + rd^k)$. Techniques used are either an interpolative procedure or a symmetrical search such as Fibonacci method[10].

7.8.1 Parabolic extrapolation

$$y(r) = f(x^k + rd^k) \tag{7.23}$$

$$y'(r) = (d^k)^t \nabla f(x^k + rd^k) \tag{7.24}$$

assuming $y(r)$ as a quadratic function of r

$$y(r) = a_1 r^2 + a_2 r + a_3 \tag{7.25}$$

$$y'(r) = 2a_1 r + a_2$$

$$r^* = -a_2/2a_1 \tag{7.26}$$

a_1, a_2 and a_3 of equation 7.25 may be estimated by obtaining $y(0)$, $y'(0)$ and $y(a)$ for some $r = a$ as follows:

$$y(0) = a_3$$

$$y'(0) = a_2$$

$$y(a) = a_1 a^2 + a_2 a + a_3$$

$$a_1 = \frac{y(a) - y(0) - ay'(0)}{a^2}$$

$$r_e = \frac{-y'(0)\,a^2}{2(y(a) - y(0) - ay'(0))} \tag{7.27}$$

where r_e is the estimated value of r^*.

Lommel and Tinney[21] have used this method.

7.8.2 Davindon's cubical interpolation[43,44]

In this method $y'(r)$ obtained from (7.24) is examined for $r = 0, h, 2h, 4h, \ldots, a, b$ where r is doubled each time and b is the first of these values at which either y' is non negative or y has not decreased. It then follows $a \leq r^* \leq b$.

Defining

$$z = 3\,\frac{y(a) - y(b)}{b - a} + y'(a) + y'(b) \tag{7.28}$$

and

$$w = \sqrt{z^2 - y'(a)\,y'(b)} \tag{7.29}$$

$$r_e = b - \left(\frac{y'(b) + w - z}{y'(b) - y'(a) + 2w}\right)(b-a) \tag{7.30}$$

If neither $y(a)$ nor $y(b)$ is less than $y(r_e)$, r_e is an estimate of r^*, otherwise the interpolation needs be repeated over the subinterval (a, r_e) or (r_e, b) according to as $y'(r_e)$ is +ve or −ve.

7.9 Optimization under equality constraints

If some of the elements of x could be considered as independent variables u, the problem may be restated as follows:

$$\min \quad f(x, u) = 0 \tag{7.31}$$

$$\text{subject to} \quad h(x, u) = 0 \tag{7.32}$$

A set of dependent variables x may be obtained from equation 7.32 for a set of independent variables u.

$$\frac{\partial f(x,u)}{\partial u_k} = \frac{\partial f(x,u)}{\partial u_k} + \sum_j \frac{\partial f(x,u)}{\partial x_j} \frac{\partial x_j}{\partial u_k} \tag{7.33}$$

$$\frac{\partial h_i(x,u)}{\partial u_k} = \frac{\partial h_i(x,u)}{\partial u_k} + \sum_j \frac{\partial h_i(x,u)}{\partial x_j} \frac{\partial x_j}{\partial u_k} = 0 \tag{7.34}$$

From equation 7.34 following matrix equation results

$$\left(\frac{\partial x_j}{\partial u_k}\right) = -\left(\frac{\partial h_i(x,u)}{\partial x_j}\right)^{-1} \left(\frac{\partial h_i(x,u)}{\partial u_k}\right) \tag{7.35}$$

where

$\left(\frac{\partial x_j}{\partial u_k}\right)$ is a column vector whose

$$\text{jth element} = \frac{\partial x_j}{\partial u_k} \tag{7.36}$$

$\left(\frac{\partial h_i(x,u)}{\partial x_j}\right)$ is a square matrix whose

$$\text{ijth element} = \frac{\partial h_i(x,u)}{\partial x_j} \tag{7.37}$$

$\left(\frac{\partial h_i(x,u)}{\partial u_k}\right)$ is a column vector whose

$$\text{ith element} = \frac{\partial h_i(x,u)}{\partial u_k} \tag{7.38}$$

$\left(\frac{\partial f(x,u)}{\partial x_i}\right)$ is a column vector whose

$$\text{ith element} = \frac{\partial f(x,u)}{\partial x_i} \qquad (7.39)$$

From (7.33) to (7.39) following equations result

$$\frac{\partial f(x,u)}{\partial u_k} = \frac{\partial f(x,u)}{\partial u_k} - \left(\frac{\partial f(x,u)}{\partial x_i}\right)^t \left(\frac{\partial h_i(x,u)}{\partial x_j}\right)^{-1} \left(\frac{\partial h_j(x,u)}{\partial u_k}\right) \qquad (7.40)$$

Defining the dual variables λ ,

$$(\lambda) = -\left(\frac{\partial h_i(x,u)}{\partial x_j}\right)^{t^{-1}} \left(\frac{\partial f(x,u)}{\partial x_i}\right) \qquad (7.41)$$

Equation 7.40 is as follows

$$\frac{\partial f(x,u)}{\partial u_k} = \frac{\partial f(x,u)}{\partial u_k} + \sum_j \lambda_j \frac{\partial h_j(x,u)}{\partial u_k}$$

$$= \frac{\partial}{\partial u_k}\left(f(x,u) + \sum_j \lambda_j h_j(x,u)\right) \qquad (7.42)$$

Defining the Lagrangian

$$L(x,u) = f(x,u) + \sum_j \lambda_j h_j \qquad (7.43)$$

where λ may be obtained from equation 7.41 or from equation 7.44 in which case the resulting expression will be similar

$$\left(\frac{\partial L(x,u)}{\partial x_j}\right) = 0 \qquad (7.44)$$

Thus if the dual variables are obtained from equation

7.44, the partial derivative of Lagrangian with respect to u_k is in fact the partial derivative of the objective function with respect to u_k under the constraint domain of 7.32. After obtaining the partial derivatives in this way, any of the gradient techniques described in sections 7.5, 7.6 and 7.7 may be used.

Bryson and Denham[48], and Dommel and Tinney[21] have used this technique.

Sufficiency condition for the minima may be stated as follows

$$\begin{bmatrix} \Delta x^t & \Delta u^t \end{bmatrix} \begin{bmatrix} \dfrac{\partial^2 L}{\partial x^2} & \dfrac{\partial^2 L}{\partial x \, \partial u} \\[2ex] \dfrac{\partial^2 L}{\partial u \, \partial x} & \dfrac{\partial^2 L}{\partial u^2} \end{bmatrix} \begin{bmatrix} \Delta x \\[2ex] \Delta u \end{bmatrix} \geq 0 \qquad (7.45)$$

Under the constraint space of 7.32, or

$$(\Delta x) = -\left(\frac{\partial h_i(x,u)}{\partial x_j}\right)^{-1} \left(\frac{\partial h_i(x,u)}{\partial u_j}\right) (\Delta u) \qquad (7.46)$$

Where $\left(\dfrac{\partial h_i(x,u)}{\partial u_j}\right)$ has its

$$ij\text{th element} = \frac{\partial h_i(x,u)}{\partial u_j} \qquad (7.47)$$

7.10 Kuhn Tucker's theorem

For the nonlinear programming problem stated in 7.1, let

$$L = f(x) + \sum_i \alpha_i \, g_i(x) + \sum_i \lambda_i \, h_i(x) \qquad (7.48)$$

The optimal solution satisfies following equations:

$$\frac{\partial L}{\partial x_i} = 0 \qquad (7.49)$$

$$\alpha_i \geq 0 \qquad (7.50)$$

$$\alpha_i \, g_i(x) = 0 \qquad (7.51)$$

$$h(x) = 0 \qquad (7.52)$$

$$g_i(x) \leq 0 \qquad (7.53)$$

α_i and λ_i are the dual variables associated with the inequality constraints and equality constraints respectively. Equations 7.50 and 7.51 imply that $\alpha_i = 0$ if the constraint 7.53 is not reached and $\alpha_i > 0$ in the alternative.

7.11 Penalty and Barrier methods[18]

These methods seek the solution of following nonlinear programming problem

$$\min f(x) \qquad (7.54)$$

$$\text{subject to} \quad g(x) \leq 0 \qquad (7.55)$$

7.11.1 Penalty method

Defining a continuous function P(x) termed as the penalty function, such that

$$P(x) = 0 \quad \text{if} \quad x \in F \tag{7.56}$$

$$P(x) > 0 \quad \text{if} \quad x \notin F \tag{7.57}$$

where F represents the feasible domain of 7.55.

One commonly used penalty function is as follows:

$$P(x) = \sum_i \max\,(0,\ g_i(x))^2 \tag{7.58}$$

Let

$$C(x,\ R) = f(x) + \frac{1}{R}\,P(x) \tag{7.59}$$

The penalty method uses a sequence R^k, $k = 1, \ldots$ such that $R^k > 0$; $R^k > R^{k+1}$ and

$$\lim_{k \to \infty} R^k = 0$$

Defining x^k as an x that minimise

$$C(x,\ R^k) = f(x) + \frac{1}{R^k}\,P(x) \tag{7.60}$$

The optimal point via the penalty method is obtained as follows. For a certain $R^1 > 0$ solve for x^1 using the unconstrained minimisation technique. x^2 is obtained for another $R^2 < R^1$, and the process is repeated.

Following results can be easily obtained -

$$C(x^k,\ R^k) \leq C(x^{k+1},\ R^{k+1})$$

$$P(x^k) \geq P(x^{k+1})$$

$$f(x^k) \leq f(x^{k+1})$$

$$x^* = \lim_{k \to \infty} x^k$$

7.11.2 Powell's penalty method[16]

With the penalty function approach mentioned earlier, there is a tendency to involve very large numbers as $R_k \rightarrow 0$, making the optimisation difficult.

Powell's modification is as follows:

$$W(x, s) = 0 \quad \text{if } x \in F$$

$$W(x, s) = \sum_{i \mid g_i(x) > 0} \frac{(g_i(x) + s_i)^2}{R_i} \quad \text{if } x \notin F \tag{7.61}$$

$$C(x, s) = f(x) + W(x, s) \tag{7.62}$$

s_i is varied from iteration to iteration as follows

$$s_i^{k+1} = s_i^k + g_i^k(x^k) \tag{7.63}$$

Thus $s_i^{k+1} > s_i^k$

Following results can be easily obtained

$$C(x^k, s^k) \leq C(x^{k+1}, s^{k+1}) \tag{7.64}$$

$$g_i(x^k) \geq g(x^{k+1})$$

$$f(x^k) \leq f(x^{k+1})$$

$$x^o = \lim_{k \to \infty} x^k$$

7.11.3 Barrier method

A function $B(x)$ is introduced such that it becomes infinity if $g(x) = 0$. This method needs that the initial guess be strictly in the constraint region which may be very difficult to obtain. The method forces the solution in the constraint region from iteration to iteration.

Fiacco Mccormic[16] has suggested the following function

$$B(x) = -\sum_i \frac{1}{g_i(x)} \tag{7.65}$$

Lootma's[16] function is as follows:

$$B(x) = -\sum_i \text{Log}\,(-g_i(x)) \tag{7.66}$$

The transformation is as follows:

$$D(x, k) = f(x) + R\,B(x) \tag{7.67}$$

The method uses a sequence R^k, $k = 1, \ldots$ such that

$R^k > 0$ for all k and

$R^k > R^{k+1}$ and

$$\lim_{k \to \infty} R^k = 0$$

following results can be easily obtained:

$$D(x^k, R^k) \geq D(x^{k+1}, R^{k+1})$$

$$D(x^k) \leq D(x^{k+1})$$

$$f(x^k) \geq f(x^{k+1})$$

$$x^* = \lim_{k \to \infty} x^k$$

-x-

CHAPTER VIII

A REVIEW OF OPTIMAL POWER FLOW TECHNIQUES

This chapter critically reviews the published work on optimal power flow studies. The convergence of some of the techniques is compared through fixed point analysis.

8.1 Objective function

The cost f in rupees per unit time of operating the power system depends upon the active powers P_i^G, the reactive powers Q_i^G being free of cost once the equipment for their production e.g. capacitors etc. has been installed. Thus

$$f = \sum_{i \in G} f_i(P_i^G) \qquad (8.1)$$

Optimisation of equation 8.1 is subject to the constraints described later.

8.2 Classification

The techniques for finding the optimal point can be broadly classified as follows:

(1) Loss formula formulation

(2) Formulation with full a.c. simulation of network.

8.3 Loss formula method

In this method optimization of equation 8.1 is achieved through the following constraint:

$$P_L - \sum_i P_i = 0 \qquad (8.2)$$

System loss P_L is expressed as a quadratic function of power generations. This method though most widely in use because of simplicity in application is based on the following assumptions, none of which really hold good in practice.

(a) Bus voltages remain constant in magnitudes and angles.

(b) Individual loads remain a constant complex fraction of the total load.

(c) Ratios of reactive to real power generations remain constant.

This method optimizes for real powers only.

Because of the limitations mentioned above a need for a more accurate method was felt.

8.4 Dopazo, Klitin, Stagg and Watsons[32] method (1967)

This method, essentially a modification of loss formula method, expresses losses as a quadratic function of active and reactive bus bar powers as follows:

$$P_L + j\,Q_L = I^t\, Z_{bus}\, I \tag{8.3}$$

$$P_L = \sum_j \sum_k (P_j P_k \alpha_{jk} + Q_j Q_k \alpha_{jk}$$
$$+ P_j Q_k \beta_{jk} - Q_j P_k \beta_{jk}) \tag{8.4}$$

where,

$$\alpha_{jk} = \frac{r_{jk}}{V_j V_k} \cos \theta_{jk} \tag{8.5}$$

$$\beta_{jk} = \frac{-r_{jk}}{V_j V_k} \sin \theta_{jk} \tag{8.6}$$

$$\frac{\partial P_L}{\partial P_j} = 2 \sum_{k=1}^{n} (\alpha_{jk} P_k + \beta_{jk} Q_k) \tag{8.7}$$

$$\frac{\partial P_L}{\partial Q_j} = 2 \sum_{k=1}^{n} (\alpha_{jk} Q_k - \beta_{jk} P_k) \tag{8.8}$$

Augmenting the objective function of equation 8.1 with the constraint equation of 8.2 through dual variable the Lagrangian is as follows:

$$L = \sum_{i \in G} f_i(P_i^G) + \lambda(P_L - \sum_j P_j) \tag{8.9}$$

Resulting optimization equations are -

$$\frac{\partial f_i}{\partial P_i^G} + \lambda \frac{\partial P_L}{\partial P_i} = \lambda \qquad \text{for } i \in G \tag{8.10}$$

$$\frac{\partial P_L}{\partial Q_i} = 0 \qquad \text{for } i \in R \tag{8.11}$$

The suggested algorithm is as follows.

(a) Estimate a generation schedule for real and reactive powers and obtain α_{jk} and β_{jk} through load flow solution.

(b) With the active generations unchanged, obtain new reactive power schedule, making use of steepest descent method (section 7.5) with the gradients obtained from equation 8.6. This step needs repeated load flow solution and computation of α_{jk} and β_{jk}.

(c) With the reactive power schedule computed above; obtain α_{jk}, β_{jk} and system losses.

(d) Estimate λ and obtain the new power generations from equation 8.10 and the incremental Loss equation.

(e) Check if equation 8.2 is satisfied, if not repeat step (d) with another λ.

(f) Repeat from step (b) till solution converges.

8.4.1 <u>Author's comments</u>

The algorithm is based on equations 8.7 and 8.8. Evidently though not admitted in the paper these equations are obtained on the assumption that α_{jk} and β_{jk} remain unchanged for the incremental change in generation, which amounts to assuming the bus bar voltage remains constant in magnitude and angle.

The reactive power optimization is achieved through the steepest descent method, with the gradients obtained from equation 8.8. This does not appear reasonable since the slack bus reactive power is a dependent variable. If the incremental change in the reactive power losses is ignored;

$$\frac{\partial P_L}{\partial Q_i} = \frac{\partial P_L}{\partial Q_i} - \frac{\partial P_L}{\partial Q_s}$$

$$= 2 \sum_{k=1}^{n} (\alpha_{ik} - \alpha_{sk}) Q_k - (\beta_{ik} - \beta_{sk}) P_k \qquad (8.12)$$

Gradient method based on equation 8.12 should be more reasonable. A more accurate representation taking incremental change in the reactive power loss into account will however need additional coefficients $(x_{ij} \cos \theta_{ij}/V_i V_j)$ and $(x_{ij} \sin \theta_{ij}/V_i V_j)$.

Fixed point analysis

Fixed point analysis to test the convergence for real power optimization has been made as follows:

Assuming the incremental cost of generation a straight line;

$$\frac{\partial f_i}{\partial P_i^G} = a_i^1 + a_i^2 P_i^G \qquad ; \quad i \in G \qquad (8.13)$$

In the suggested algorithm, this is made equal to

$$\lambda \left(1 - \frac{\partial P_L}{\partial P_i}\right) \qquad (8.14)$$

Fixed point representation is as follows:

$$P_i = \frac{\lambda\left(1 - \frac{\partial P_L}{\partial P_i}\right) - a_i^1}{a_i^2} \tag{8.15}$$

$$= \phi_i(P^G) \tag{8.16}$$

λ obtained from equations 8.[illegible] and 8.15 is as follows:

$$\lambda = \frac{P_L - \sum_{k \notin G} P_k + \sum_{k \in G} \frac{a_k^1}{a_k^2}}{\sum_{k \in G} \frac{(1 - \partial P_L / \partial P_k)}{a_k^2}} \tag{8.17}$$

$$\frac{\partial \lambda}{\partial P_i} = \frac{2\lambda \sum_{k \in G} \frac{\alpha_{ki}}{a_k^2} \cdot \frac{\partial P_L}{\partial P_i}}{\sum_{k \in G} \frac{(1 - \partial P_L / \partial P_k)}{a_i^2}} \tag{8.18}$$

$$\frac{\partial \phi_i(P^G)}{\partial P_i} = \frac{1}{a_i^2}\left[-2\,\alpha_{ki}\lambda + \frac{\partial \lambda}{\partial P_i}\left(1 - \frac{\partial P_L}{\partial P_i}\right)\right] \tag{8.19}$$

The sufficiency condition for the Gauss type iteration to converge (section 5.4) is

$$\sum_i \left| \frac{\partial f_i(P_i^G)}{\partial P_i} \right| < 1 \qquad (8.20)$$

This condition will not be satisfied if any of the generation units has a near flat incremental cost characteristics. This establishes the unreliable convergence of the method.

A better method will be to obtain the real powers also through the gradient technique of section (7.9) with the following Lagrangian.

$$L = \sum_i f_i(P_i^G) + \lambda_1(P_L - \sum_i P_i) + \lambda_2(Q_L - \sum_i Q_i) \qquad (8.21)$$

Ignoring the incremental change of P_L; λ_1 and λ_2 are as follows:

$$\lambda_1 = \frac{\partial f_s/\partial P_s^G}{\left(1 - \frac{\partial P_L}{\partial P_s^G}\right)} \qquad (8.22)$$

$$\lambda_2 = \lambda_1 \frac{\partial P_L}{\partial Q_s} \qquad (8.23)$$

$$\frac{\partial L}{\partial P_i^G} = \frac{\partial f_i}{\partial P_i^G} + \lambda_1\left(\frac{\partial P_L}{\partial P_i^G} - 1\right) \qquad (8.24)$$

$$\frac{\partial L}{\partial Q_i^G} = \lambda_1 \frac{\partial P_L}{\partial Q_i^G} - \lambda_2 \qquad (8.25)$$

While the convergence with Gauss iterative method was found doubtful, a first order gradient method is stable since this ensures the objective function reduce from step to step.

Additional drawback of the method is large storage requirement for non-sparse Z-matrix and α_{jk}, β_{jk} coefficients.

8.5 Formulation with full a.c. simulation of network[19,36,39,46,47]

These methods optimize equation 8.1 under the equality constraints of equations 4.1 and 4.2 which are restated as follows:

$$P_i = P_i^G - P_i^L = p_i(V, \theta) = V_i \sum_j V_j y_{ij} \cos(\theta_i - \theta_j - \Psi_{ij}) \tag{8.26-i}$$

$$Q_i = Q_i^G - Q_i^L = q_i(V, \theta) = V_i \sum_j V_j y_{ij} \sin(\theta - \theta_j - \Psi_{ij}) \tag{8.27-i}$$

Inequality constraints on generation nodes are as follows:

$$(P_i^G)^2 + (Q_i^G)^2 - (\bar{S}_i)^2 \leq 0 \tag{8.28}$$

$$\underline{P}_i - P_i^G \leq 0 \tag{8.29}$$

$$P_i^G - \bar{P}_i \leq 0 \tag{8.30}$$

$$Q^G - \bar{Q}_i \leq 0 \tag{8.31}$$

$$\underline{Q}_i - Q_i^G \leq 0 \tag{8.32}$$

Constraints 8.28 to 8.32 are determined by current magnitude, boiler instability, turbine output, generator excitation limits and generator instability respectively.

Additional constraints determined primarily by insulation and transformer tap ranges for consumption as well as generation nodes are as follows:

$$V_i - \overline{V}_i \leq 0 \qquad (8.33)$$

$$\underline{V}_i - V_i \leq 0 \qquad (8.34)$$

In addition to these, upper limit of the line currents also need be fixed.

$$I_{ij} = \left| \frac{V_i \angle \theta_i - V_j \angle \theta_j}{Z_{ij}} \right|$$

where I_{ij} is the current and Z_{ij} the impedence of line connected between nodes i and j.

$$= \left| \frac{2\,V_i \sin\left(\frac{\theta_i - \theta_j}{2}\right) \angle \frac{\pi - \theta_i - \theta_j}{2} + (V_i - V_j) \angle \theta_j}{Z_{ij}} \right|$$

$$\simeq \frac{2\,V_i \sin\left(\frac{\theta_i - \theta_j}{2}\right)}{Z_{ij}}$$

for normal differences in line voltages and phase angles.

Thus prescribing the upper limit of line current almost amounts to prescribing the upper limit of $|\theta_i - \theta_j|$. This constraint can be described as follows:

$$|\theta_i - \theta_\alpha| - T_{i\alpha} = 0 \qquad (8.35)$$

where α is the adjacent node.

8.6 Subclassification

Methods making use of a.c. representation of network can be further classified as follows:

(1) Variational equations and linear programming formulation[28]

(2) Kuhn Tuckers formulation[11,13,19,46]

(3) Penalty function formulation[4,15,16]

(4) Mixed formulation, making use of both Kuhn Tucker's theorem or equivalent formulation and penalty functions[17,21].

8.7 Dauphin, Feingold and Spohn's method[28] (1967)

This method makes use of variational equations and linear programming technique, for which the objective function is as follows:

$$f = \sum_{i \in G} (a_i + b_i P_i^G) \qquad (8.36)$$

$$\Delta f = \sum_{i \in G} b_i \Delta P_i^G \qquad (8.37)$$

The general compatibility relation of equation 4.32 is as follows:

$$\sum_i \lambda_i \Delta P_i + \sum_k \mu_k \Delta Q_k + \sum_l u_l \Delta V_l = 0 \qquad (8.33)$$

λ, μ, u, k and l are as defined in section 4.6.

The method seeks the minimisation of equation 8.37 under constraints of equations 8.36 and 8.29 to 8.34.

Since the effect of second order terms is ignored, this in general causes ΔP_i^G, ΔQ_i^G and ΔV_l to fail to satisfy the power flow equations. Therefore the distribution is recalculated after obtaining the new coefficients of compatibility equations. Possibility of new values of the objective function greater than the previous value has also been admitted, in which case recalculations on the basis of half the correction has been recommended.

8.7.1 <u>Author's comments</u>

From equations 8.37 and 8.38,

$$\Delta f = \sum_{\substack{i \in G \\ i \neq s}} \left(b_i - \frac{b_s \lambda_i}{\lambda_s}\right) \Delta P_i - \sum_{k \in R} \frac{\mu_k b_s}{\lambda_s} \Delta Q_k - \sum_{l \in R} \frac{u_l b_s}{\lambda_s} \Delta V_l \qquad (8.39)$$

Assuming the inequality constraints of slack bus not violated for minimum Δf; following equations hold good:

$$\Delta P_i = \bar{P}_i^G - P_i^G \quad \text{if} \quad b_i - \frac{b_s \lambda_i}{\lambda_s} < 0$$

$$\Delta P_i = \underline{P}_i^G - P_i^G \quad \text{if} \quad b_i - \frac{b_s \lambda_i}{\lambda_s} > 0$$

$$\Delta Q_k = \bar{Q}_i^k - Q_i^k \quad \text{if} \quad \frac{b_s \mu_k}{\lambda_s} > 0$$

$$\Delta Q_k = \underline{Q}_i^k - Q_i^k \quad \text{if} \quad \frac{b_s \mu_k}{\lambda_s} < 0$$

$$\Delta V_i = \bar{V}_i - V_i \quad \text{if} \quad \frac{u_i b_s}{\lambda_s} > 0$$

$$\Delta V_i = V_i - \underline{V}_i \quad \text{if} \quad \frac{u_i b_s}{\lambda_s} < 0$$

This amounts to substituting for either maximum value or minimum value of independent variables for most buses in the next iteration. This indicates that jamming of the algorithm is quite likely to occur.

6.6 Carpentier's method[36,37,46,47] (1962-1963)

This method is based upon the following Lagrangian:

$$L = \sum_{i \in G} f_i(P_i^G) + \sum_i \lambda_i (P_i(V,\theta) - P_i)$$

$$+ \sum_i \mu_i (q_i(V,\theta) - Q_i) + \sum_{i \in R+G} \nu_i \left((P_i^G)^2 \right.$$

$$\left. + (Q_i^G)^2 - S_i^2 \right) + \sum_{i \in G} m_i (\underline{P}_i - P_i^G)$$

$$+ \sum_{i \in R} e_i (Q_i^G - \bar{Q}_i) + \sum e'_i (\underline{Q}_i - Q_i^G)$$

$$+ \sum_i \sum_\alpha r_{i\alpha} (\theta_i - \psi_\alpha - T_{i\alpha})$$

$$+ \sum \pi_i (V_i - \bar{V}_i) + \sum \pi'_i (\underline{V}_i - V_i) \qquad (8.40)$$

Necessary conditions for f to be minimum are given in section (7.10).

$$\frac{\partial L}{\partial P_i^G} = 0 \quad \text{gives}$$

$$\lambda_i = \frac{\partial F_i}{\partial P_i^G} + 2 K_i P_i^G - m_i \quad \text{for } i \in G. \qquad (8.41\text{-}1)$$

$$\frac{\partial L}{\partial Q_i^G} = 0 \quad \text{gives}$$

$$\mu_i = 2 \lambda_i Q_i^G + e_i - e'_i \quad \text{for } i \in R \qquad (8.42\text{-}1)$$

$$\frac{\partial L}{\partial \theta_i} = 0 \quad \text{gives}$$

$$\sum_j \lambda_j \frac{\partial P_j}{\partial \theta_i} + \sum_j \mu_j \frac{\partial Q_j}{\partial \theta_i} + \sum_\alpha (r_{i\alpha} - r_{\alpha i}) = 0 \qquad (8.43\text{-}1)$$

$$\frac{\partial L}{\partial V_i} = 0 \quad \text{gives}$$

$$\sum_j \lambda_j \frac{\partial P_j}{\partial V_i} + \sum_j \mu_j \frac{\partial Q_j}{\partial V_i} + \pi_i - \pi'_i = 0 \qquad (8.44\text{-}1)$$

Since slack bus voltage and angle is fixed; equations 8.43 and 8.44 do not apply for this bus.

Apart from these equations; power flow equations of 8.26 and 8.27 must be satisfied.

8.8.1 Verification of number of equations and unknowns

(i) Consumption nodes: The unknowns are θ_i, V_i, λ_i, μ_i and the non-zero dual variables corresponding to the inequality constraints of equations 8.28 to 8.35.

The equations are 8.26, 8.27, 8.43 and 8.44 plus one equation each arising from the active inequality constraints.

(ii) Production nodes (excluding slack node): The unknowns are P_i^G , Q_i^G , θ_i , V_i , λ_i , μ_i and the non-zero dual variables of inequality constraints. The equations are 8.41, 8.42, 8.43, 8.44, 8.26 and 8.27 plus one equation each arising from the corresponding active inequality constraints.

(iii) Slack node: The unknowns are P_i^G , Q_i^G , λ_i and μ_i and the non-zero dual variables of the inequality constraints. The equations are 8.41, 8.42, 8.26 and 8.27 plus one equation each from the active inequality constraint.

8.8.2 Algorithm

(i) Assume a set of unknown P, Q, V, θ, μ and λ. These may not satisfy the power flow equations.

(2) Assuming $\Delta\theta_\alpha = 0$ and $\Delta V_\alpha = 0$; obtain $\Delta\theta_i$ and ΔV_i from the variation equations of 8.26-i and 8.27-i respectively as follows:

$$\Delta\theta_i = \frac{P_i^G - P_i^L - p_i(V, \theta)}{\partial p_i(V, \theta)/\partial\theta_i} \tag{8.45-i}$$

$$\Delta V_i = \frac{Q_i^G - Q_i^L - q_i(V, \theta)}{\partial q_i(V, \theta)/\partial\theta_i} \tag{8.46-i}$$

Check if $\theta_i + \Delta\theta_i$ and $V_i + \Delta V_i$ does not violate the inequality constraints. If these constraints are violated, fix θ_i and V_i to the constraint value.

(3) If constraints in (2) are not active, $r_{i\alpha}$, $r_{\alpha i}$, π_i and π_i' are equal to zero, obtain λ_i and μ_i respectively from equations 8.43-i and 8.44-i, if i is not a slack node.

(4) If constraints in (2) are not active and it is not a slack node, obtain P_i^G and Q_i^G from equations 8.41-i and 8.42-i respectively, if it is a production node. If these powers are within limits, h_i, m_i, e_i and e_i' are zero. If power constraints are violated, these are equated to the constraint value and corresponding dual variables obtained from equations 8.41-i and 8-42-i respectively.

(5) If constraint in (2) are active and this is a production node, generations are obtained from equations 8.26-i and 8.27-i. λ_i and μ_i are obtained from equations 8.41-i and 8.42-i respectively. If it is not a production node, λ_i or $r_{i\alpha}$ and λ_j or π_i are obtained from equations 8.43-i and 8.44-i respectively.

The iteration is repeated from (2) till the solution converges.

8.3.3 <u>Author's comments</u>

The method turns out to be the Gauss Seidal iterative method applied to each node in succession. The convergence of this iterative technique is not reliable. A fixed point analysis of the method also appears difficult. The method assumes that a solution in the feasible domain satisfying all equality and inequality constraints exist which may not always be so.

The solution does not remain in the feasible domain from iteration to iteration. Hence final values are of little use if solution fails to converge. Investigations on this method by Peschon[19] et al show divergence in solutions where the initial guess is not close enough to the optimal solution. For systems with unusually high or low reactive powers, the method failed to give the solution even if the initial guess is very close to the optimal solution.

This paper can be regarded as a historical break through. Later on number of other papers making use of Carpentier's formulation have appeared. Numerical technique adopted to solve system of equations are based on either Gauss Seidal or gradient method. In many cases not all the inequality constraints are considered.

8.9 El-abiad and Jaimes's method[13] (1969)

In this paper inequality constraints are excluded. Partial derivative of the Lagrangian with respect to the slack bus voltage is also equated to zero, which evidently means that this voltage is also adjustable. Equations 8.41 to 8.44 thus reduce to the following:

$$\lambda_i = \frac{\partial f_i}{\partial P_i^G} , \quad i \in G \tag{8.47-1}$$

$$\mu_i = 0 ; \qquad i \in R \tag{8.48-1}$$

$$\sum_j \lambda_j \frac{\partial P_j}{\partial \theta_i} + \sum_j \mu_j \frac{\partial Q_j}{\partial \theta_i} = 0; \quad i \neq s \tag{8.49-1}$$

$$\sum_j \lambda_j \frac{\partial P_j}{\partial V_i} + \sum_j \mu_j \frac{\partial Q_j}{\partial V_i} = 0 \tag{8.50-1}$$

From equations 8.48, 8.49 and equations of 8.50 for nodes with no reactive power generation resulting equations is 8.51.

$$\left[\begin{array}{c|c} m^1_{ij} = \dfrac{\partial P_j}{\partial e_i} & m^2_{ij} = \dfrac{\partial Q_j}{\partial e_i} \\ i \neq s;\ j \neq s & i \neq s;\ j \notin R \\ \hline m^3_{ij} = \dfrac{\partial P_j}{\partial V_i} & m^4_{ij} = \dfrac{\partial Q_j}{\partial V_i} \\ i \neq s;\ j \notin R & i \notin R,\ j \notin R \end{array}\right] \left[\begin{array}{c} \lambda_j \\ j \neq s \\ \hline \mu_j \\ j \notin R \end{array}\right] = -\lambda_s \left[\begin{array}{c} b^1_j = \dfrac{\partial P_s}{\partial e_j} \\ j \neq s \\ \hline b^2_j = \dfrac{\partial P_s}{\partial V_j} \\ j \notin R \end{array}\right] \qquad (8.51)$$

Jacobian on the right hand side of equation 8.51 is the transfer of jacobian needed for load flow solution by Newton's method with voltage as independent variable for reactive power production nodes.

The algorithm is as follows:

(1) Assuming a set of feasible generations and voltages and a value of λ_s; λ and μ are obtained from equations 8.51 and 8.48. Real generations are obtained from equation 8.47 in terms of a multiplier λ' included to assure a proper level of generation. For the incremental costs given by equation 8.13 the value of λ' such that generations are equal to the pre-estimated losses and demand is given by 8.52

$$\lambda' = \frac{P_L + P_D + \sum_{i \in G} a^1_i / a^2_i}{\sum_{i \in G} \left\{ \dfrac{\lambda_i}{a^2_i} \right\}} \qquad (8.52)$$

and $$\frac{\partial f_i}{\partial P^G_i} = \lambda' \lambda_i \qquad (8.53)$$

(2) Obtain voltage of the nodes with reactive power generations from the equations 8.50-i, $i \notin R$, which after rearranging is as follows:

$$AE = b \tag{8.54}$$

where A is a square matrix whose ijth element is as follows:

$$a_{ij} = \lambda_j\, y_{ij} \cos(\theta_i - \theta_j - \psi_{ij})$$
$$+ \mu_i\, y_{ij} \sin(\theta_i - \theta_j - \psi_{ij}) \text{ for } i \neq j;\ i \in R,\ j \in R$$

$$a_{ii} = \lambda_i\, g_{ii}$$

$$b_i = - \sum_{j \notin R} \lambda_j V_j\, y_{ij} \cos(\theta_i - \theta_j - \psi_{ij})$$
$$- \sum_{j \notin R} \mu_j V_j\, y_{ij} \sin(\theta_i - \theta_j - \psi_{ij}) \quad \text{for } i \in R$$

E is a column vector of voltages of nodes with reactive power generation.

8.9.1 Author's comments

Though the exact fixed point analysis appears difficult, an analysis for real power dispatch has been made as follows.

Premultiplying both sides of equation 8.51 by a row vector $(\Delta\theta_i,\ i \neq s;\ \Delta V_i,\ i \notin R)$; following relation is obtained

$$\sum_{i} \lambda_i \, \Delta P_i + \sum_{i \notin R} \mu_i \, \Delta Q_i = 0 \qquad (8.55)$$

Thus the dual variables obtained in this method are in fact coefficients of the compatibility relation of equation 4.36. These coefficients are related to the loss formula expression of equation 8.4 as follows:

$$\lambda_i = \lambda'(1 - \frac{\partial P_L}{\partial P_i}) \qquad (8.56)$$

$$\mu_i = - \lambda' \frac{\partial P_L}{\partial Q_i} \qquad (8.57)$$

Step 1 of the algorithm thus amounts to

$$\frac{\partial F_i}{\partial P^G_i} = \lambda_i = \lambda'(1 - \frac{\partial P_L}{\partial P_i}) \qquad (8.58)$$

This step is similar to that of equation 8.13 and 8.14. Hence the fixed point analysis of 8.4.1 is valid for this method too. Thus convergence of the method is doubtful if any of the generation unit has a near flat incremental cost characteristics. Poor convergence and need of acceleration technique has been admitted for the five node problem given in the paper.

8.10 Shen and Laughton's method[11] (1969)

This method also utilises the Carpentier's formulation and uses Gauss Seidel type iterative technique. Optimal adjustment for phase shifting transformer and a tap setting is also included. Apart from the constant load demand, a variable component of load demand proportionate to the voltage magnitude has been included, however objective function is cost of real power generation. Excluding some of these additional features, the system Lagrangian is as follows:

$$L_1 = L + \sum_h \sigma_h (t_h - \bar{t}_h) + \sum_h \sigma'_h (\underline{t}_h - t_h)$$
$$+ \sum_h \beta_h (\phi_h - \bar{\phi}_h) + \sum_h \beta'_h (\underline{\phi}_h - \phi_h) \tag{8.59}$$

where L is given by equation 8.40.

Equations obtained are 8.41, 8.42, 8.43 and 8.44. Equations 8.43 and 8.44 are initially written for all the nodes including the slack node. Additional equations for the transformers are as follows:

$$\frac{\partial L_1}{\partial t_h} = \lambda_i \frac{\partial p_i}{\partial t_h} + \lambda_j \frac{\partial p_j}{\partial t_h} + \mu_i \frac{\partial q_i}{\partial t_h}$$
$$+ \mu_j \frac{\partial q_j}{\partial t_h} + \sigma_h - \sigma'_h = 0 \tag{8.60}$$

$$\frac{\partial L_1}{\partial \phi_h} = \lambda_i \frac{\partial p_i}{\partial \phi_h} + \lambda_j \frac{\partial p_j}{\partial \phi_h} + \mu_i \frac{\partial q_i}{\partial \phi_h}$$
$$+ \mu_j \frac{\partial q_j}{\partial \phi_h} + \beta_h - \beta'_h = 0 \tag{8.61}$$

where i and j are two nodes to which the hth transformer is connected.

In each iteration the computation is divided in four steps involving in turn computation of

(1) active generation and voltage phase angles

(2) phase shifting transformer tap settings

(3) reactive generation and voltage magnitudes

(4) non-phase shifting transformer tap settings

8.10.1 Determination of active generation and voltage phase angles

In this step, an estimate of power generations, voltage phase angles and dual variables λ is made, assuming other variables as fixed.

Let

$$e_{pi} = P_i(V, \theta) - P_i^G + P_i^L \qquad (8.62)$$

$$e_{\theta i} = \frac{\partial L}{\partial \theta_i} = \text{l.h.s. of equation 8.43-i} \qquad (8.63)$$

$$\Delta e_{pi} + \Delta P_i^G = \sum_j \frac{\partial P_i(V,\theta)}{\partial \theta_j} \Delta \theta_j \qquad (8.64)$$

$$\Delta e_{Gi} = \sum_j \frac{\partial P_j}{\partial \theta_i} \Delta \lambda_j + \sum_j \frac{\partial}{\partial \theta_j} \left(\frac{\partial \lambda_j}{\partial \theta_i}\right) \Delta \theta_j \qquad (8.65)$$

Without any explanation, the paper indicates that $\frac{\partial}{\partial \theta_j}\left(\frac{\partial \lambda_i}{\partial \theta_i}\right)$ elements are small; therefore equation 8.65 modified is as follows:

$$\Delta e_{Gi} = \sum_j \frac{\partial P_j}{\partial \theta_i} \Delta \lambda_j \qquad (8.66)$$

The method is as follows:

(1) Since the matrix associated with 8.66 is singular; $\Delta \lambda_i$ are obtained from this equation in terms of $\Delta\lambda_n$ which is to be adjusted for proper level of generations obtained from equation 8.41.

(2) Phase angles are obtained from 8.64.

8.10.2 Author's comments

This step is similar to the real power optimization of 8.9 in the following respect. Both obtain the dual variables as functions of λ_s which is chosen for proper level of generations. Thus fixed point analysis of 8.9.1 is valid for this method too for real power optimization.

This indicates that convergence is doubtful if any of the generation unit has a near flat incremental cost characterition. λ and ε in this method have been obtained by assuming μ and e respectively as constants. Convergence of a partitioned iterative method has been found poorer than convergence without partitioning (well established for Z-matrix load flow).

8.10.3 Obtaining phase shifting transformer settings

Equation 8.61 written explicitly is as follows:

$$
\begin{aligned}
\frac{\partial L}{\partial \phi_h} = & -\lambda_i V_i V_j y_{ij} \sin(\theta_i - \theta_j - \psi_{ij} + \phi_h) \\
& + \lambda_j V_i V_j y_{ij} \sin(\theta_j - \theta_i - \psi_{ij} - \phi_h) \\
& + \mu_i V_i V_j y_{ij} \cos(\theta_i - \theta_j - \psi_{ij} + \phi_h) \\
& - \mu_j V_i V_j y_{ij} \cos(\theta_j - \theta_i - \psi_{ij} - \phi_h) \\
& + \beta_h - \beta_h'
\end{aligned}
\qquad (8.67)
$$

Thus ϕ_h is at a limit $\overline{\phi}_h$ or $\underline{\phi_h}$ with β_h or β_h' positive or

$$
(\theta_i - \theta_j + \phi_h) = \tan^{-1} \frac{(\lambda_i - \lambda_j) \sin \psi_{ij} + (\mu_i - \mu_j) \cos \psi_{ij}}{(\lambda_i + \lambda_j) \cos \psi_{ij} - (\mu_i + \mu_j) \sin \psi_{ij}} \qquad (8.68)
$$

New values of voltage phase angles and phase shifts are obtained from equation 8.68 and following variational equation

$$
\Delta P_i = 0 = \sum_{\substack{k \\ k \neq n}} \frac{\partial P_i}{\partial \theta_k} \Delta \theta_k + \sum_h \frac{\partial P_i}{\partial \phi_h} \Delta \phi_h \qquad (8.69)
$$

8.10.4 Obtaining reactive power generation and voltage magnitudes

Let

$$e_{qi} = q_i(V, t) - Q_i^G + Q_i^L \qquad (8.70)$$

$$e_{vi} = \frac{\partial L}{\partial V_i} = \text{left hand side of equation 8.44} \qquad (8.71)$$

$$\Delta e_{qi} + \Delta Q_i^G = \sum_j \frac{\partial q_i}{\partial V_j} \Delta V_j \qquad (8.72)$$

$$\Delta e_{vi} = \sum_j \frac{\partial q_j}{\partial V_i} \Delta \mu_j + \sum_j \frac{\partial}{\partial V_j} (e_{vi}) \Delta V_j + \Delta \pi_i - \Delta \pi_i' \qquad (8.73)$$

Neglecting $\frac{\partial}{\partial V_j}(e_{vi})$ for $i \neq j$, ΔQ^G, $\Delta \mu$ and ΔV are obtained from equations 8.72, 8.73 and 8.42, remembering

$\Delta Q_j^G = 0$ for load buses

$\mu_j = \Delta \mu_j = 0$ for buses with reactive power generation not violating V or Q constraints.

8.10.5 Obtaining optimal transformer settings

The transformer setting t_h is either at $\bar{t}_h$ or $\underline{t}_h$ with σ_h or σ_h' positive or following relation obtained from equation 8.60 hold good

$$G_{h1} t_h - G_{h2} = 0 \qquad (8.74)$$

where
$$G_{h1} = 2\lambda_i V_i^2 y_{h1} \cos \psi_{h1} - 2 \mu_i y_{h1} V_i^2 \sin \psi_{h1} \qquad (8.75)$$

$$G_{h2} = \lambda_i V_i V_j y_{h2} \cos(\theta_i - \theta_j - \psi_{h2})$$

$$+ \lambda_j V_i V_j y_{h2} \cos(\theta_j - \theta_i - \psi_{h2})$$

$$+ \mu_i V_i V_j y_{h2} \sin(\theta_i - \theta_j - \psi_{h2})$$

$$+ \mu_j V_i V_j y_{h2} \sin(\theta_i - \theta_j - \psi_{h2}) \qquad (8.76)$$

$$y_{ii} \underline{/\psi_{ii}} = y^o_{ii} \underline{/\psi^o_{ii}} + t_h^2 \, y_{h1} \underline{/\psi_{h1}} \qquad (8.77)$$

$$y_{ij} \underline{/\psi_{ij}} = y^o_{ij} \underline{/\psi^o_{ij}} - t_h \, y_{h2} \underline{/\psi_{h2}} \qquad (8.78)$$

If e_t is the value of left hand side of equation 8.74;

$$\Delta e_t = t_h \frac{\partial G_{h1}}{\partial V_i} \Delta V_i + G_{h1} \Delta t_h$$

$$- \frac{\partial G_{h2}}{\partial V_i} \Delta V_i - \frac{\partial G_{h2}}{\partial V_j} \Delta V_j \qquad (8.79)$$

$$\Delta q_i = \frac{\partial q_i}{\partial V_i} \Delta V_i + \frac{\partial q_i}{\partial V_j} \Delta V_j + \frac{\partial q_i}{\partial t_h} \Delta t_h \qquad (8.80)$$

$$\Delta q_j = \frac{\partial q_j}{\partial V_i} \Delta V_i + \frac{\partial q_j}{\partial V_j} \Delta V_j + \frac{\partial q_j}{\partial t_h} \Delta t_h \qquad (8.81)$$

Δt_h, ΔV_i and ΔV_j are obtained from equations 8.79, 8.80 and 8.81 with $\Delta q_i = \Delta q_j = 0$ if voltage constraints are not violated.

8.10.6 <u>Author's comments</u>

The author has very extensively tested the method of tap adjustment described in 8.10.5 on AEP 14 bus and 30 bus system and finds that this does not work. In most cases Δt_h is found of the same sign as the gradient obtained from equation 8.60. This indicates that the objective function will in fact increase if this adjustment is made, and optimal solution will not be obtained even with damping or acceleration.

Similar to Carpentier's method, the method also turns out to be the Gauss Seidal iterative method, with the iteration not remaining in feasible domain. The method also assumes that a feasible solution exists. Certain cases where the solution fails to converge for real power scheduling and transformer settings have been pointed out. For active and reactive power scheduling the method turns out a partitioned version of 8.9 in which case the authors themselves have pointed out poor convergence.

As pointed out earlier, the load includes a variable component proportionate to the voltage magnitude while the objective function is cost of real power generation. To the author this does not sound very reasonable. With change in voltage, revenue also changes because of this component of demand. Hence a more reasonable objective function is cost of generation minus revenue for the variable component of load demand.

8.11 Peschon, Piercy, Tinney, Tveit and Cuenod's method[19,36] (1968)

This method is in fact a historical development of the research carried out for the BPA system. The system has 100% hydro-generation. Thus objective function is system loss. Method provides for reactive power optimization. Use has been made of the Lagrangian of equation 8.40 with objective function as slack bus power and constraints on active power, apparent power and angles ignored. Resulting equations are as follows:

$$\lambda_s = 1 \tag{8.82}$$

$$\mu_i = e_i - e_i' \tag{8.83}$$

$$\sum_j \lambda_j \frac{\partial p_j}{\partial \theta_i} + \sum_j \mu_j \frac{\partial q_j}{\partial \theta_i} = 0; \; i \neq s \tag{8.84}$$

$$\sum_j \lambda_j \frac{\partial p_j}{\partial V_i} + \sum_j \mu_j \frac{\partial q_j}{\partial V_i} + \pi_i - \pi_i' = 0 \tag{8.85}$$

In view of the obvious fact that efficient transmission of power implies high voltage levels throughout the grid, it is initially guessed that at the optimal point nodes with reactive power generation, except where the upper limit of generation is violated, will be at the maximum voltage. The algorithm is as follows:

(1) With $V = \overline{V}$ at the reactive production nodes, all remaining primary variables P, Q, θ and V are obtained from power flow equations.

(2) For nodes where $Q > \bar{Q}$; $Q = \bar{Q}$ is imposed and primary variables recomputed.

(3) Obtain dual variables from equations 8.82 to 8.85 and check for the values. If $Q_i = \bar{Q}_i$; $e_i > 0$. If $V_i = \bar{V}_i$; $\pi_i > 0$. For nodes where e_i or π_i are negative; corresponding Q_i or V_i is decreased to make these dual variables as zero. A linear relationship is assumed between V_i and π_i or Q_i and e_i as follows:

$$V_i(k) = V_i(k-1) - \frac{V_i(k-1) - V_i(k-2)}{\pi_i(k-1) - \pi_i(k-2)} \pi_i(k-1) \tag{8.86}$$

$$Q_i(k) = Q_i(k-1) - \frac{Q_i(k-1) - Q_i(k-2)}{e_i(k-1) - e_i(k-2)} e_i(k-1) \tag{8.87}$$

8.11.1 Author's comments

This method turns out to be a first order gradient method with a set of dependent variables adjusted through linear extra-polation. Convergence depends essentially upon the choice of $V_i(2) - V_i(1)$ and $Q_i(2) - Q_i(1)$. The iteration is in the feasible domain of power flow equations. It is admitted that convergence has been good where there are few reactive production nodes with the optimal voltage not at the maximum, which essentially means that the initial guess is sufficiently close to the optimal solution.

Two separate load flow solutions have been suggested in step (1) and (2). With the authors load flow programme described in Chapter 6, this can be achieved through a single solution.

7.12 Sasson's method[4,15,16]

This method[16] makes use of Powell's penalty function formulation of section (7.11.2) to represent power flow equations and other inequality constraints, all expressed in rectangular form. Optimisation is achieved by Fletcher Powell's gradient method[44] (sec. 7.7) in terms of real and quadrature components of line voltages, and active and reactive power generations. The approach needs less computer logic compared to Carpentier-type methods. Because of very large number of variables, optimisation may be very difficult to achieve in presence of discontinuous penalty functions. Fletcher powell's gradient method need a non-sparse though symmetrical Hessian matrix. For the 30 node problem 230 Fletcher Powell's steps were needed[16]. With 4 nodes for real power generation and 6 nodes for reactive power generation, a 68 x 68 matrix need be recalculated for every Fletcher Powell's step. Iteration is not in the feasible domain.

In a later paper[4], it is admitted that the method becomes very inefficient in terms of storage requirement and time for large systems. For such systems use of decomposition technique has been suggested. The paper demonstrates that this way the computer time is cut down to 10 for a system of 57 buses and

5 for 110 buses. Efficiency of decomposition depends upon how the system is partitioned, which still is an open question.

8.13 Dommel and Tinney's method[21]

This method makes use of mixed formulation. The power flow equations in polar form have been handled through the dual variables and other inequality constraints by augmenting the objective function through penalty function approach, of 7.11.1. The independent variables (called control variables) are the active power generations and line voltage magnitude or reactive power of the generation nodes except the powers for the slack node. Slack bus active power generation is expressed as a function of voltage magnitudes and angles. Thus the objective function of equation 8.1 is rewritten as follows:

$$f = \sum_{\substack{i \in G \\ i \neq s}} f_i(P_i^G) + f_s(p_s(V,\theta) + P_s^L) \qquad (8.88)$$

Sensitivity of the objective function with respect to the decision variables is obtained by the method of section (7.9) making use of the following Lagrangian:

$$L = \sum_{\substack{i \in G \\ i \neq s}} f_i(P_i^G) + f_s(p_s^G(V,\theta) + P_s^L) + \text{penalty terms}$$

$$+ \sum_{\substack{i \\ i \neq s}} \lambda_i(p_i(V,\theta) - P_i^G + P_i^L)$$

$$+ \sum_{\substack{i \\ i \neq s}} \mu_i(q_i(V,\theta) - Q_i^G + Q_i^L) \qquad (8.89)$$

It may be noted that this Lagrangian is different from that of Carpentier's. Use has been made of the steepest descent technique of section (7.5) with the optimal step length obtained through parabolic extrapolation (section 7.8-1). Whenever the gradient changes the sign from iteration to iteration, the control variable is re-obtained through linear intrapolation. This modification of the gradient method has been called as mixed method.

C.13.1 Author's comments

This technique though not the latest the author finds as the most promising of all the discussed techniques for the following reasons:

(1) The iteration is in the feasible domain of power flow equations.

(2) Gradient technique has been used instead of Gauss Seidel iterative technique. While convergence of the later technique is doubtful, convergence of the former is more reliable since ensures that the objective function reduce from step to step, even if the initial guess is not in the convex domain.

(3) Dependent variables (called state variables) have been obtained through load flow solution and not through gradient technique. Convergence of the gradient technique becomes more difficult as the number of

variables to be adjusted by this technique increase. Application of Newton's load flow assures that the power flow equations are satisfied to a very high degree of accuracy.

(4) Algorithm makes the application of sparsity techniques possible.

The author's experience over the method while trying [illegible] 14 bus and 30 bus systems is as follows:

(1) Application of parabolic intrapolation in the presence of penalty functions did not work satisfactorily.

(2) Though the paper suggests the use of either voltage magnitude or reactive power for nodes with such generations as independent variable, the author finds that the step length obtained with voltage magnitude as independent variable is much smaller (about 50%) compared to one obtained with reactive power as such variable. This makes the change in tap settings and real power generation smaller, resulting in the need for more iterations with the former as independent variable.

(3) Application of mixed gradient method did not work satisfactorily in the presence of adjustable reactive power and tap settings possibly because the off diagonal elements of the Hessian $\partial^2 f/\partial Q \partial t$ could not be ignored.

8.14 Ramamoorthy and Gopala Rao's method[17]

In this paper voltage magnitudes and angles of generation buses are adjusted through gradient method. Inequality constraints on other variables are represented by square type penalty functions. In the analysis, application of dual variables has been avoided. It is as follows:

$$\begin{bmatrix} \Delta P^G \\ \Delta Q^G \\ \hline \Delta P^L \\ \Delta Q^L \end{bmatrix} = \left[\begin{array}{c|c} C_1 & C_2 \\ \hline C_3 & C_4 \end{array}\right] \begin{bmatrix} \Delta \theta^G \\ \Delta V^G \\ \hline \Delta \theta^L \\ \Delta V^L \end{bmatrix} \qquad (8.90)$$

With $\Delta P^L = 0$ and $\Delta Q^L = 0$,

$$\begin{bmatrix} \Delta \theta^L \\ \Delta V^L \end{bmatrix} = - C_4^{-1} C_3 \begin{bmatrix} \Delta \theta^G \\ \Delta V^G \end{bmatrix} \qquad (8.91)$$

$$\begin{bmatrix} \Delta P^G \\ \Delta Q^G \end{bmatrix} = \left[C_1 - C_2 C_4^{-1} C_3 \right] \begin{bmatrix} \Delta \theta^G \\ \Delta V^G \end{bmatrix} \qquad (8.92)$$

Elements of jacobian in equation 8.92 are the partial derivatives of the generation with respect to the voltage phase angles and magnitudes of generation buses in the domain of power flow equations.

$$\frac{\partial f}{\partial \theta_i^G} = \sum_j \frac{\partial f_j}{\partial P_j^G} \frac{\partial P_j^G}{\partial \theta_i^G} + \sum_j \frac{\partial f}{\partial Q_j^G} \frac{\partial Q_j^G}{\partial \theta_i^G} \qquad (8.93)$$

$$\frac{\partial f}{\partial V_i} = \sum_j \frac{\partial f}{\partial P_j} \frac{\partial P_j}{\partial V_i} + \sum_j \frac{\partial f}{\partial Q_j} \frac{\partial Q_j}{\partial V_i} \qquad (8.94)$$

.14.1 <u>Author's comments</u>

(1) In the choice of independent variables, depend the minimum to maximum ratio of eigen values of the Hessian matrix which determines convergence of the gradient technique (section 7.5). Though Hessian is too difficult to obtain, the convergence as judged from the results supplied is not good enough compared to that of 8.13 as found by the author.

(2) Though the method has the advantages of 8.13 from 1 to 3 the algorithm needs C_4^{-1} explicity in each iteration. This matrix for a 30 bus system is 52 x 52 with four generation nodes.

-x-

[illegible]

[illegible]

[illegible]

As discussed in section 8.13.1, the gradient method of Dommel and Tinney[21] based on the technique of section 7.9 is the most promising. The author's computer programme is based on this method modified to take care of the experiences mentioned in this section (8.13.1). While the method of reference 21 need the cost of slack bus expressed in terms of V and θ, in this programme it is taken as function of power.

As discussed in section 8.1..1, it is more advantageous to treat the reactive power generation as independent variable, rather than voltage magnitudes. Treating the inequality constraints on voltage magnitudes for nodes with reactive power generation through penalty method has been found to result in considerable convergence difficulties since the solution oscillates between inside and outside of the feasible domain from iteration to iteration. Such constraints are therefore treated differently.

9.1 Effect of inequality constraints

Inequality constraints on the independent variable u_i can be considered by realising for the optimal solution;

$$\frac{\partial L}{\partial u_i} = 0 \qquad \text{if } \underline{u}_i \leq u_i \leq \bar{u}_i \tag{9.1}$$

$$\frac{\partial L}{\partial u_i} \geq 0 \qquad \text{if } u = \underline{u}_i \tag{9.2}$$

$$\frac{\partial L}{\partial u_i} \leq 0 \qquad \text{if } u = \bar{u}_i \tag{9.3}$$

Inequality constraints on the function $g_i(x, u)$ can be considered by restating the problem of section 7.1 as follows:

$$\min \; f(x, u) \tag{9.4}$$

subject to

$$h(x, u) = 0 \tag{9.5}$$

$$g(x, u) - v = 0 \tag{9.6}$$

$$v \leq 0 \tag{9.7}$$

where v is the vector of additional variables introduced. If the jacobian $\left[\left(\frac{\partial g}{\partial x}\right)^t, \left(\frac{\partial h}{\partial x}\right)^t \right]^t$ is non-singular, v can be considered as independent variable.

Defining the Lagrangian as follows

$$L = f(x,u) + \sum \lambda_i \, h_i(x,u) + \sum \alpha_i (g_i(x,u) - v_i) \tag{9.8}$$

Then the analysis of section 7.9; if the dual variables are obtained from equation 9.9 as follows,

$$
\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_k \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_l \end{bmatrix}
= -
\begin{bmatrix}
\frac{\partial g_1}{\partial x_1} & \cdots\cdots & \frac{\partial g_k}{\partial x_1} & \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_l}{\partial x_1} \\
\vdots & & \vdots & & & \\
\frac{\partial g_1}{\partial x_{k+l}} & \cdots & \frac{\partial g_k}{\partial x_{k+l}} & \frac{\partial F_1}{\partial x_{k+l}} & \cdots & \frac{\partial F_l}{\partial x_{k+l}}
\end{bmatrix}^{-1}
\begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \vdots \\ \frac{\partial f}{\partial x_{k+l}} \end{bmatrix}
\qquad (9.9)
$$

then

$$\frac{\partial}{\partial v_i} = -\alpha_i = \frac{\partial f(x,u)}{\partial F_i}\bigg|_{x,u} \quad \text{in the constraint domain} \qquad (9.10)$$

Hence for the optimal solution the following relations are satisfied:

$$(9.11) \qquad \text{if} \quad v_i < 0 \; ; \qquad \alpha_i = 0$$

$$(9.12) \qquad \text{if} \quad v_i = 0 \; ; \qquad \alpha_i \geq 0$$

9.2 Variables

The unknown variables have been classified as follows:

(a) Independent variables: These variables include active and reactive powers of all generation nodes treated as p q nodes, slack bus voltage magnitude and adjustable transformer tap settings.

(b) Dependent variables: These variables are slack bus active and reactive powers, voltage magnitudes and angles of p q nodes, reactive powers and phase angles of p v nodes..

9.3 Lagrangian

The computer programme is based upon the following Lagrangian :

$$
\begin{aligned}
L = & \sum_{i \in G} f_i(P_i^G) + \text{penalty terms} \\
& + \sum_i \lambda_i (p_i(V, \theta) - P_i^G + P_i^L) \\
& + \sum_i \mu_i (q_i(V, \theta) - Q_i^G + Q_i^L) \\
& + \sum_{i \in P} \pi_i (V_i - \bar{V}_i) + \sum_{i \in R} \pi_i' (V_i - \underline{V_i}) \\
& + \alpha_s (V_s - V_s') + \beta_r (\theta_r - \theta_r') \qquad (9.13)
\end{aligned}
$$

Penalty terms include voltage inequality constraints on nodes other than with reactive power generation, inequality constraints on the slack bus powers and inequality constraints on voltage angle differences. After some experimentation, Powell's penalty functions of equation 7.61 have been adopted. These are restated as follows:

$$
\begin{aligned}
P(x, s) &= 0 \qquad \text{if } \underline{x} \leq x \leq \overline{x} \\
&= \frac{(x - \overline{x} + s_x^h)^2}{s_x} \qquad \text{if } x > \overline{x} \\
&= \frac{(\underline{x} - x + s_x^l)^2}{s_x} \qquad \text{if } x < \underline{x}
\end{aligned}
\tag{9.14}
$$

Dual variables α_r and β_r have been introduced with the slack bus voltage magnitude and angle to make the Lagrangian compatible with the load flow programme of Chapter VI.

If the objective function is different from the cost of real power generation, the same is substituted in place of $\sum_{i \in G} f_i(P_i^G)$.

9.4 Dual Variables

The variables have been obtained by making the partial derivatives of the Lagrangian with respect to the dependent variables equal to zero as described in section 7.9.

$$\frac{\partial L}{\partial P_s^G} = 0 = \frac{\partial F(P^G)}{\partial P_s^G} - \lambda_s \qquad (9.15)$$

$$\frac{\partial L}{\partial Q_s^G} = 0 = -\mu_s \qquad (9.16)$$

The reactive power generation nodes have been considered as p-q nodes except if the voltage constraint is violated for that node, in which case these are considered as p v nodes with the assigned voltages equal to $\bar{V}$ or $\underline{V}$ depending upon which constraints are violated. For such nodes Q^G is a dependent variable

$$\frac{\partial L}{\partial Q_i^G} = 0 = \mu_i \text{ for p v node} \qquad (9.17)$$

The dual variables π_i and π_i' ; $i \in R$ are zero if the voltage constraint is not reached.

Thus for every node except the slack node, there are two constraint equations and two non-zero dual variables as follows :

(a) p q nodes (set L)

The constraint equations are as follows:

$$h_{2i-1} = p_i(V, \theta) - P_i^G + P_i^L = 0 \qquad (9.18)$$

$$h_{2i} = q_i(V, \theta) - Q_i^G + Q_i^L = 0 \qquad (9.19)$$

Associated non-zero dual variables are λ_i and μ_i respectively.

(b) p v nodes (Set ..)

The constraint equations are as follows:

$$h_{2i-1} = P_i(V, \theta) - P_i^0 + P_i^1 = 0 \tag{9.20}$$

$$h_{2i} = V_i - \bar{V}_i \text{ or } V_i - \underline{V}_i = 0 \tag{9.21}$$

The non-zero dual variables are λ_i and π_i or π_i'.

(c) Slack node (Set C)

The constraint equations included for the reasons mentioned earlier are as follows:

$$h_{2s-1} = \theta_s - \theta_s' = 0 \tag{9.22}$$

$$h_{2s} = V_s - V_s' = 0 \tag{9.23}$$

Associated dual variables are β_s and α_s respectively.

In addition the slack bus has a non-zero dual variable λ_s which is obtained from equation 9.15.

By equating the partial derivative of L with respect to V_j to zero and pre-multiplying the resulting equation by V_j,

$$-V_j \frac{\partial F}{\partial V_j} - \lambda_s V_j \frac{\partial P_s}{\partial V_j}$$

$$= \sum_{i \in A} \left(\lambda_i V_j \frac{\partial h_{2i-1}}{\partial V_j} + \mu_i V_j \frac{\partial h_{2i}}{\partial V_j} \right)$$

$$+ \sum_{i \in B} \left(\lambda_i V_j \frac{\partial h_{2i-1}}{\partial V_j} + \pi_i' V_j \frac{h_{2i}}{V_j} \right)$$

$$+ \sum_{i \in C} \left(\beta_i V_j \frac{\partial h_{2i-1}}{\partial V_j} + \alpha_i V_j \frac{\partial h_{2i}}{\partial V_j} \right) \qquad (9.24\text{-}j)$$

where F is the objective function augmented with the penalty terms.

$\pi_i' = \pi_i$ or π_i' depending upon which constraint is active.

Similarly by equating the partial derivative of Lagrangian with respect to θ_j to zero, resulting equation is as follows:

$$-\frac{\partial F}{\partial \theta_j} - \lambda_s \frac{\partial P_s}{\partial \theta_j} = \sum_{i \in A} \left(\lambda_i \frac{\partial h_{2i-1}}{\partial \theta_j} + \mu_i \frac{\partial h_{2i}}{\partial \theta_j} \right)$$

$$+ \sum_{i \in B} \left(\lambda_i \frac{\partial h_{2i-1}}{\partial \theta_j} + \pi_i' \frac{\partial h_{2i}}{\partial \theta_j} \right)$$

$$+ \sum_{i \in C} \left(\beta_i \frac{\partial h_{2i-1}}{\partial \theta_j} + \alpha_i \frac{\partial h_{2i}}{\partial \theta_j} \right) \qquad (9.25\text{-}j)$$

Thus the 2n non-zero dual variables can be obtained from equations 9.24-j and 9.25-j, j = 1, ..., n. The associated matrix is the transpose of the jacobian used in load flow solution. These non-zero dual variables have been obtained by making use of the table of factor obtained during the load flow solution and the computer logic described in section 2.17.

9.5 Sensitivity of the augmented objective function

The gradient vector of the augmented objective function with respect to the independent variables, in the constraint domain of power flow equations is as follows:

$$\frac{\partial F}{\partial P_i} = \frac{\partial C_i^G}{\partial P_i} - \lambda_i \quad \text{for } i \in G;\ i \neq s \tag{9.26}$$

$$\frac{\partial F}{\partial Q_i} = -\mu_i \quad \text{for } i \in N \tag{9.27}$$

$$\frac{\partial F}{\partial V_i} = -\pi_i' \quad \text{for } i \in N \tag{9.28}$$

from equation 9.10

$$\frac{\partial F}{\partial t_h} = \lambda_i \frac{\partial P_i}{\partial t_h} + \lambda_j \frac{\partial P_j}{\partial t_h} + \mu_i \frac{\partial Q_i}{\partial t_h} + \mu_j \frac{\partial Q_j}{\partial t_h}$$

$$= G_{h1}\, t_h - G_{h2} \tag{9.29}$$

where G_{h1} and G_{h2} are respectively given by equations 8.75 and 8.76 respectively.

$$\frac{\partial F}{\partial \psi_h} = -\lambda_i V_i V_j y_{ij} \sin(\theta_i - \theta_j - \psi_{ij} + \psi_h)$$
$$+ \lambda_j V_i V_j y_{ij} \sin(\theta_j - \theta_i - \psi_{ij} - \psi_h)$$
$$+ \mu_i V_i V_j y_{ij} \cos(\theta_i - \theta_j - \psi_{ij} + \psi_h)$$

$$- \mu_j V_i V_j y_{ij} \cos(\theta_j - \theta_i - \psi_{ij} - \delta_n) \qquad (9.3[illegible])$$

For the optimal solution equations 9.1 to 9.[illegible] hold good. For the bus bars with reactive power generation, reaching the voltage constraint; equation 9.31 holds

$$\pi_i^1 \geq 0 \quad \text{if} \quad V_i = \bar{V}_i$$

$$\pi_i^1 \leq 0 \quad \text{if} \quad V_i = \underline{V}_i \qquad (9.31)$$

$$\frac{\partial C}{\partial P_i} = \lambda_i \qquad (9.32)$$

With the increase in load demand, operating cost increases. Thus $\lambda_i > 0$ for all i if the objective function is the operating cost.

9.6 Direction vector

The direction vector can be found out by any of the methods discussed in sections 7.5, 7.6 and 7.7. The conjugate gradient and Fletcher-Powell's gradient methods guarantee the minimum can be located exactly within certain finite number of iterations for the quadratic function. Even in regions remote from minimum, these methods by taking account of the curvature of the function are expected to deal with complex situations such as presence of a long curving valley[43,44]. As discussed earlier conjugate gradient method needs only very little additional memory in space compared to the steepest descent method, this method has therefore been adopted. The new direction vector has been obtained from equations 7.14 and 7.15.

... constraints on the decision variable; following modification has been made.

If the decision variable u_i^k at kth iteration is at the constraint limit and sign of gradient satisfies equations 9.2 and 9.3; that variable is kept fixed. For such variable $d_i^k = 0$. Corresponding $\left(\frac{\partial f}{\partial u_i}\right)$ term is not added while obtaining ∇f^1 ... $\nabla f(u^k)$. If sign of the gradient for the variable at constraint limit do not satisfy equations 9.2 and 9.3; the following value of d_i^{k-1} is substituted

$$d_i^{k-1} = \frac{u_i^k - u_i^{k-1}}{r_e} \qquad (9.13)$$

For bus bars with controllable reactive powers with constraints on Q and V, Q has normally been taken as an independent variable, since this has been found to result in better convergence. For such bus bars a provision has been made in the load flow solution to check if voltage constraint is violated in which case the bus bar is considered as p v node with the bus voltage magnitude equal to the constraint value, provided Q is within the constraint limits (section 6.1).

For each node the gradient has been approximately estimated as follows:

$$\frac{\partial}{[illegible]} \cdot \frac{[illegible]}{[illegible]_1} \frac{\partial V_1}{[illegible]_1} \approx - [illegible] \frac{V_1}{1 - [illegible]_1^2 [illegible]_{11}} \quad \ldots [illegible]$$

9.7 <u>Optimal step length</u> (cf section 7.[illegible])

In order to find the optimal step length through parabolic extrapolation as done in reference 21 (section 7.[illegible].1), y(0), y'(0) and y(r) need be obtained and r_e is obtained from equation (7.[illegible]).

In order to obtain y(r) a load flow solution is needed. y'(r) can be obtained by just one additional solution of the simultaneous equations for which table of factor has already been obtained in compact form. With these four values, cubical extrapolation is now possible to obtain a more accurate value of r_e. Even this was not found working satisfactorily, hence Davidon's cubical interpolation described in section 7.[illegible].2 had to be used. This method needs y(r) and y'(r) for r = 0, h, 2h, 4h, ..., a, b where r is doubled each time and b is the first of these values at which y' is non-negative or y has not decreased. For every such value of r, a load flow solution and one additional triangularization is needed. Hence it is desirable that y'(r) is non-negative in as few trials as possible. After obtaining the direction vector, h is obtained on the basis of maximum change in the independent variable. After obtaining the value of r , maximum change in the decision variable is reobtained. For the next trial the maximum change in the independent variable is assumed 'a' (a < 1) times the maximum change of this trial.

$$h_p = \frac{\Delta [illegible]}{\max (d_i^{[illegible]})}$$

$$h_q = \frac{\Delta}{\max (d_i^q)}$$

$$h_t = \frac{\Delta t}{\max (d_i^t)} \qquad (9.35)$$

where Δ , Δ and Δt are the estimated maximum changes in real power, reactive power and tap settings. The value of h is the minimum of h_p, h_q and h_t. For the next iteration ΔP, Δ and Δt are estimated as follows:

$$\Delta P_i^{new} = (a \;\; \Delta P^{old} \;\; r[illegible])/h_p$$

$$\Delta \;{}_i^{new} = (a \;\; \Delta [illegible]^{old} \;\; r[illegible])/h_q$$

$$\Delta t_i^{new} = (a \; \Delta t^{new} \;\; r[illegible])/h_t \qquad (9.36)$$

Typical initial values taken are Δ = .5 pu, Δ = .1 pu, Δt = .1 and a = .8 .

y'(r) is obtained from equation 7.24 except in case when some of independent variables violate the constraint, the following modification is made:

$$y'(r) = ([illegible]) \nabla (u + rd) \qquad ([illegible].37)$$

$$= \sum_i g_i d_i \qquad (9.38)$$

where g_i is the gradient obtained as in section 9.[illegible] if

$$v_i = u_i + r d_i$$

and

$$\underline{u}_i \leq v_i \leq \bar{u}_i .$$

If $v_i < \underline{u}_i$ or $v_i > \bar{u}_i$; the new value of the independent variable is taken equal to the bounded value and $g_i = 0$ if $y'(r) < 0$.

Thus in order to obtain $y'(r)$; $\sum_{i \in F} g_i d_i$ is obtained for the independent variables within the constraint limit. $\sum_{i \notin F} g_i d_i$ for the variables violating the constraints is obtained separately with g_i as the gradient at the constraint limit. The two are added if $\sum_{i \in F} g_i d_i$ is non-negative.

In case load flow programme fails to converge; a solution with half the changes in the independent variables is attempted. Convergence of load flow solution is judged on the basis of maximum mismatch in powers as described in section 6.3.

9.8 System Description

The system extensively tested is [illegible]0-bus system, description of which is as follows.

Base MVA = 100

Table 9.8.1: Impedence and line charging data

Line designation		Resistance per unit	Reactance per unit	Line charging per unit
1	2	0.0192	0.0575	0.0264
1	3	0.0452	0.1652	0.0204
2	4	0.0570	0.1737	0.0184
2	5	0.0472	0.1983	0.0209
2	6	0.0581	0.1763	0.0187
3	4	0.0132	0.0379	0.0042
4	6	0.0119	0.0414	0.0045
4	12	0.0000	0.0256	0.0000
5	7	0.0460	0.1160	0.0102
6	7	0.0267	0.0820	0.0085
6	8	0.0120	0.0420	0.0045
6	9	0.0000	0.2080	0.0000
6	10	0.0000	0.5560	0.0000
6	28	0.0169	0.0599	0.0065
8	28	0.0636	0.2000	0.0214
9	10	0.0000	0.1100	0.0000
9	11	0.0000	0.2080	0.0000
10	17	0.0324	0.0845	0.0000
10	20	0.0936	0.2090	0.0000
10	21	0.0348	0.0749	0.0000
10	22	0.0727	0.1499	0.0000
12	13	0.0000	0.1400	0.0000
12	14	0.1231	0.2559	0.0000
12	15	0.0662	0.1304	0.0000
12	16	0.0945	0.1987	0.0000
14	15	0.2210	0.1997	0.0000
15	18	0.1073	0.2185	0.0000
15	23	0.1000	0.2020	0.0000
16	17	0.0824	0.1923	0.0000
18	19	0.0639	0.1292	0.0000
19	20	0.0340	0.0680	0.0000
21	22	0.0116	0.0236	0.0000
22	24	0.1150	0.1790	0.0000
23	24	0.1320	0.2700	0.0000
24	25	0.1885	0.3292	0.0000
25	26	0.2544	0.3800	0.0000

Table 9.[illegible].1 (contd..

Line designation		Resistance per unit	Reactance per unit	Line charging per unit
25	27	0.1093	0.2087	0.0000
27	28	0.0000	0.3960	0.0000
27	29	0.21[illegible]	0.4153	0.0000
27	30	0.[illegible]02	0.6027	0.0000
29	30	0.2399	0.4[illegible]3	0.0000

Table 9.[illegible].2: Static capacitor data

Bus number	Susceptance per unit
10	0.1900
24	0.4300

Table 9.[illegible].3: Transformer data (Initial values)

Transformer designation		Tap setting
4	12	0.932
6	9	0.978
6	10	0.969
28	27	0.968

Table 9. [illegible]

[illegible] obtained through the subroutine [illegible] in [illegible] .6 [illegible] matrix stored in the [illegible] obtained through the subroutine [illegible] in [illegible].

Table [illegible]

Node	Bus number	Diagonal elements of admittance matrix: Real part	Diagonal elements of admittance matrix: Imaginary part	I	J
1	[illegible]	6.46[illegible]	-2[illegible].[illegible]95	1	1
[illegible]	16	[illegible].7[illegible]2	-3[illegible].64[illegible]	[illegible]	1
[illegible]	14	9.4[illegible]9	-2[illegible].602	6	2
4	26	16.314	-[illegible].405	7	3
[illegible]	[illegible]	4.[illegible]	-12.19[illegible]	9	[illegible]
6	29	[illegible].341	-[illegible].174	10	6
7	1[illegible]	6.544	-1[illegible].4[illegible]6	1[illegible]	8
8	12	7.7[illegible]3	-[illegible]6.[illegible]27	1[illegible]	10
[illegible]	[illegible]1	0.000	-1[illegible].706	16	11
10	28	1[illegible].462	-41.[illegible]	1[illegible]	12
11	[illegible]	0.000	- [illegible].[illegible]07	2[illegible]	14
12	30	6.573	-2[illegible].424	2[illegible]	1[illegible]
13	2	0.000	- 7.142	26	16
1[illegible]	11	4.017	- [illegible].424	26	17
1[illegible]	27	0.365	-16.011	27	1[illegible]
16	2[illegible]	3.034	- [illegible].497	29	20
17	10	[illegible].638	-14.710	30	21
18	9	4.886	- 9.906	30	23
19	24	0.958	-17.983	31	24
20	[illegible]	7.667	-1[illegible].750	32	25
[illegible]1	7	21.[illegible]	-4[illegible].10[illegible]	32	27
[illegible]	20	21.[illegible]	-4[illegible].4[illegible]2	33	28
[illegible]	6	[illegible].42[illegible]	- 6.95[illegible]	34	30
[illegible]4	19	5.[illegible]11	- [illegible].[illegible]01	35	31
2[illegible]	18	4.495	- 7.06[illegible]	36	33
2[illegible]	1	1.[illegible]16	- 1.017	38	34
27	17	[illegible]	- 9.460	38	35
2[illegible]	1[illegible]	5.[illegible]06	-22.342	41	[illegible]6
[illegible]9	[illegible]	1.907	- [illegible].604	41	39
30	4	1.599	- .017	42	40

Table [illegible]

[illegible]	[illegible]	[illegible]$_2$	[illegible]	Off-diagonal element: Real part	Imaginary part
1	2	1	1	-[illegible].[illegible]24	1[illegible].646
2	3	1	2	-1.243	[illegible].096
[illegible]	4	2	[illegible]	-1.705	5.197
4	[illegible]	[illegible]	6	-1.135	4.772
5	6	2	4	-1.6[illegible]6	5.116
6	4	2	5	-[illegible].19[illegible]	2[illegible].530
7	6	4	7	-6.41[illegible]	22.[illegible]11
[illegible]	12	5	9	0.0[illegible]	[illegible].640
9	7	6	10	-2.954	7.449
10	7	6	11	-[illegible].59[illegible]	11.026
11	[illegible]	6	12	-[illegible].2[illegible]9	22.012
12	9	6	13	[illegible].000	4.701
13	10	9	1[illegible]	[illegible].000	1.742
14	28	9	17	-[illegible].[illegible]62	15.463
15	23	4	[illegible]	-1.44[illegible]	4.540
16	10	1[illegible]	22	0.000	9.090
17	11	12	23	0.000	4.007
1[illegible]	17	12	24	-[illegible].[illegible]56	10.317
1[illegible]	20	14	26	-1.7[illegible]4	[illegible].985
2[illegible]	21	1[illegible]	25	-5.101	10.98[illegible]
[illegible]1	22	10	18	-[illegible].61[illegible]	[illegible].400
22	1[illegible]	16	29	[illegible].000	7.1[illegible]
23	14	15	27	-1.[illegible]26	[illegible].173
24	1[illegible]	18	30	-[illegible].095	6.097
2[illegible]	16	10	19	-1.[illegible]51	4.104
26	15	19	31	-2.490	2.25[illegible]
27	18	10	20	-1.[illegible]10	[illegible].687
2[illegible]	2[illegible]	10	[illegible]1	-1.968	3.976
29	17	21	[illegible]	-1.882	4.393
30	1[illegible]	15	28	-3.07[illegible]	[illegible]
31	2[illegible]	22	[illegible]	-5.[illegible]82	11.764
32	2[illegible]	23	24	-1[illegible].774	34.127
[illegible]3	2[illegible]	24	3[illegible]	-2.540	[illegible].954
34	2[illegible]	2[illegible]	[illegible]6	-1.461	2.989
35	25	2[illegible]	37	-1.309	[illegible].2[illegible]7
36	26	6	1[illegible]	-1.216	1.[illegible]17
37	27	8	1[illegible]	-1.96[illegible]	[illegible].760
38	28	27	38	0.000	2.444
39	29	27	39	-0.995	1.[illegible]81
40	30	27	4[illegible]	-0.687	1.293
41	30	29	41	-0.912	1.723
[illegible]	30	[illegible]	[illegible]	0.000	0.000

where,

i_1 = index sequence of corresponding row, right off diagonal elements

i_2 = index sequence of corresponding row, left off diagonal elements

L_1 = column of corresponding right off diagonal element

L_2 = column of corresponding left off diagonal element

l = corresponding location of left off diagonal element in the array of right off diagonal elements.

Table 9.6.5: Transformer tap admittance

Transformer designation		l_t	y_t	y_{td}
4	12	8	0.0+j3.90625	0.0-j3.90625
6	9	12	0.0+j4.80769	0.0-j4.80769
6	10	13	0.0+j1.79856	0.0-j1.79856
28	27	38	0.0+j2.52525	0.0-j2.52525

where, l_t = location of right off diagonal element in Table 9.6.4-b

y_{td} = (y+ jB) of equation 3.3

y_t = (-y) of equation 3.3.

Table 9.[illegible].6: Initial operating conditions

Bus number	Initial bus voltage: Magnitude per unit	Initial bus voltage: Phase angle degrees	Generation: MW	Generation: Mvar	Load: MW	Load: Mvar
1	1.060	0.000	0.000	0.000	0.000	0.000
2	1.000	0.000	40.000	0.000	21.700	12.700
3	1.000	0.000	0.000	0.000	2.400	1.200
4	1.000	0.000	0.000	0.000	7.600	1.600
5	1.000	0.000	0.000	0.000	94.200	19.000
6	1.000	0.000	0.000	0.000	0.000	0.000
7	1.000	0.000	0.000	0.000	22.800	10.900
8	1.000	0.000	0.000	0.000	30.000	30.000
9	1.000	0.000	0.000	0.000	0.000	0.000
10	1.000	0.000	0.000	0.000	5.800	2.000
11	1.000	0.000	0.000	0.000	0.000	0.000
12	1.000	0.000	0.000	0.000	11.200	7.500
13	1.000	0.000	0.000	0.000	0.000	0.000
14	1.000	0.000	0.000	0.000	6.200	1.600
15	1.000	0.000	0.000	0.000	8.200	2.500
16	1.000	0.000	0.000	0.000	3.500	1.800
17	1.000	0.000	0.000	0.000	9.000	5.800
18	1.000	0.000	0.000	0.000	3.200	0.900
19	1.000	0.000	0.000	0.000	9.500	3.400
20	1.000	0.000	0.000	0.000	2.200	0.700
21	1.000	0.000	0.000	0.000	17.500	11.200
22	1.000	0.000	0.000	0.000	0.000	0.000
23	1.000	0.000	0.000	0.000	3.200	1.600
24	1.000	0.000	0.000	0.000	8.700	6.700
25	1.000	0.000	0.000	0.000	0.000	0.000
26	1.000	0.000	0.000	0.000	3.500	2.300
27	1.000	0.000	0.000	0.000	0.000	0.000
28	1.000	0.000	0.000	0.000	0.000	0.000
29	1.000	0.000	0.000	0.000	2.400	0.900
30	1.000	0.000	0.000	0.000	10.600	1.900

* Slack bus.

9.7 Example

Assuming active power generation at buses 1, 2 and 11 and reactive power generations at buses 1, 2, 5, [illegible], 11 and 13 and assuming following generation costs;

$$f_1(P_1^G) = .7\,P_1^G + .01(P_1^G)^2$$

$$f_2(P_2^G) = .7\,P_2^G + .02(P_2^G)^2$$

$$f_{11}(P_{11}^G) = .8\,P_{11}^G + .02(P_{11}^G)^2\ ;$$

With the reactive power generations $-.5 \leq Q_i^G \leq .5$ and the line voltage magnitudes $1.0 \leq V \leq 1.15$; results are given below. Slack bus voltage is also considered a decision variable.

Table [illegible].1: Initial operating conditions

Bus Number	V pu	θ degrees	[illegible] pu	[illegible] pu
1	1.0[illegible]	[illegible]	2.60209	[illegible].14564
2	1.0[illegible]7[illegible]	-5.[illegible]097	0.1830[illegible]	-0.12700
3	1.0[illegible]017	-[illegible].10003	-0.0239[illegible]	-0.01199
4	1.[illegible]1127	-[illegible].77967	-0.07600	-0.01600
5	1.[illegible]	-14.5[illegible]142	-0.[illegible]4200	[illegible]
6	[illegible]34[illegible]	-11.[illegible]	[illegible].00000	[illegible].0000[illegible]
7	[illegible]16	-1[illegible].[illegible]1[illegible]	-[illegible].[illegible]2[illegible]00	-[illegible].10[illegible]
8	1.0[illegible]	-12.2[illegible]53	-[illegible].30000	-0.30[illegible]20
9	[illegible].9[illegible]2[illegible]	-1[illegible].2[illegible]470	0.0000[illegible]	0.0000[illegible]
10	[illegible]	-17.1534[illegible]	-0.0[illegible]7[illegible]	-0.02000
11	1.0[illegible]	-1[illegible].[illegible]6	[illegible].00000	[illegible].[illegible]4[illegible]08
12	[illegible]7642	-16.17507	-[illegible].11200	-[illegible].07[illegible]00
13	1.0[illegible]	-1[illegible].17507	0.00000	0.16[illegible]43
14	0.96[illegible]4[illegible]	-17.27[illegible]72	-0.06199	-0.01600
15	[illegible]	-17.[illegible]49[illegible]	-0.08200	-0.02500
16	[illegible].97[illegible]37	-16.92746	-0.03500	-0.01800
17	0.9[illegible]147	-17.3[illegible]664	-0.09000	-0.05799
18	[illegible].96546	-1[illegible].[illegible]161	-0.03200	-0.00900
19	[illegible]6560	-1[illegible].[illegible]942	-0.09500	-0.0[illegible]400
20	0.97142	-1[illegible].15[illegible]04	-0.02199	-0.00700
21	0.[illegible]8795	-17.[illegible]29	-0.17500	-0.11200
22	[illegible]71[illegible]6	-17.[illegible]5118	0.00000	0.00000
23	0.98550	-1[illegible].67487	-0.03200	-0.01600
24	1.[illegible]1[illegible]14	-1[illegible].6[illegible]44	-0.08699	-0.06700
25	0.90787	-1[illegible].129[illegible]4	0.00000	0.00000
26	0.96[illegible]65	-1[illegible].57549	-0.03500	-0.02299
27	0.9[illegible]2	-16.[illegible]2239	0.00000	0.00000
28	1.00212	-12.24857	0.00000	0.00000
29	0.95951	-1[illegible].26488	-0.02399	-0.00900
30	0.94748	-19.2[illegible]135	-0.10599	-0.01900

With the above operating conditions obtained at the tap settings of Table 9.[illegible]; the objective function without penalty function = 2.1[illegible]741 pu and with penalty function = 2.20042.

Optimal operating conditions and the corresponding tap settings are given in Tables 9.9.2 and 9.9.3 respectively.

Table 9.[illegible]: [illegible]

[illegible]	[illegible]	[illegible]	[illegible]	pu
1	[illegible]	[illegible]	[illegible]	[illegible]
2	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	1.07344	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
6	[illegible]	[illegible]	0.00000	0.00000
7	[illegible]	[illegible]	-0.22800	[illegible]
[illegible]	1.07040	[illegible]	-0.30000	[illegible]
[illegible]	[illegible]	[illegible]	0.00000	0.00000
[illegible]	[illegible]	[illegible]	-0.05799	[illegible]
11	[illegible]	[illegible]	[illegible]	[illegible]
12	[illegible]	[illegible]	-0.11200	[illegible]
[illegible]	[illegible]	[illegible]	0.00000	[illegible]
14	[illegible]	[illegible]	-0.06199	[illegible]
[illegible]	[illegible]	[illegible]	-0.08200	[illegible]
[illegible]	[illegible]	[illegible]	-0.03500	[illegible]
17	[illegible]	[illegible]	-0.09000	[illegible]
[illegible]	[illegible]	[illegible]	-0.03200	[illegible]
[illegible]	[illegible]	[illegible]	-0.09500	[illegible]
[illegible]	[illegible]	[illegible]	-0.02199	[illegible]
21	[illegible]	[illegible]	[illegible]	-0.11200
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	1.07710	[illegible]	-0.03200	-0.01600
24	1.11166	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	0.00000	0.00000
[illegible]	[illegible]	[illegible]	-0.00000	[illegible]
27	1.08473	-13.06276	0.00000	0.00000
28	1.06980	[illegible]	0.00000	0.00000
29	1.06612	[illegible]	-0.02399	-0.00900
30	1.05536	[illegible]	[illegible]	[illegible]

Table [illegible]: [illegible] settings for optimal operation:

[illegible]		[illegible] settings
4	12	0.[illegible]961
[illegible]	9	0.[illegible]71
[illegible]	10	0.977[illegible]1
[illegible]	27	1.00[illegible]02

[illegible] value [illegible] objective function with and without penalty [illegible] conditions is found to be [illegible] function [illegible] 2.1590[illegible].

Table [illegible].4:

[illegible] optimal operating conditions.

Table [illegible]: [illegible] power generations

Bus number	Initial value	Optimal value
1	2.62209	1.97725
2	0.40000	0.65[illegible]94
11	0.00000	0.33684

Table [illegible]: [illegible]active power [illegible]erations:

[illegible]r	Initial value	Optimal value
1	[illegible].1456[illegible]	-[illegible]. 6772
2	0.[illegible]	[illegible].11142
[illegible]	[illegible].[illegible]52[illegible]	[illegible].50[illegible]
	[illegible].26937	0.36974
11	[illegible].[illegible]4[illegible]	-[illegible].02051
13	[illegible].1[illegible]843	0.04740

Table 9.9.4-c: Transformer tap settings

Transformer designation		Initial value	Optimal value
4	12	0.9[illegible]200	[illegible].97961
6	9	0.97800	[illegible].9[illegible]071
6	10	0.96900	0.97721
2[illegible]	27	0.96800	1.00802

Table 9.9.4-d: Slack bus voltage

Initial value = 1.[illegible]

Optimal value = 1.08521

Table 9.9.2: Objective functions

	Initial value	Final value
Objective function without penalty function	[illegible].11741	2.15905
Objective function with penalty function	[illegible].[illegible]042	2.15905

9.10 Example 2:

Another application for which the computer programme was tested is planning for additional reactive power equipment needed to maintain the minimum prescribed voltage on the buses. The objective function is as follows:

$$f = \sum (\text{additional Mvar needed}) \qquad (9.39)$$

The total additional capacity is to be made minimum by adjusting the controllable reactive power generation.

Table 9.10.1:

Resume of initial and optimal operating conditions [illegible] [illegible].1[illegible].1- : [illegible]

[illegible]de	[illegible] value	[illegible] value
1	[illegible].0902[illegible]	-[illegible].[illegible]5[illegible]
2	.[illegible]	-[illegible].1[illegible]
5	[illegible].33562	-0.23[illegible]1
[illegible]	.12[illegible]	-[illegible].[illegible]6552
11	[illegible]	-[illegible].[illegible]5272
1[illegible]	[illegible]	-[illegible].44[illegible]6

Tabl[illegible] [illegible].1[illegible].1-[illegible] : [illegible] Kvar needed ([illegible])

[illegible]ode	Initial value	[illegible]al value
[illegible]	0.06434	-
12	0.16163	-
14	[illegible].04093	-
1[illegible]	0.01040	-
16	[illegible].0[illegible]235	-
1[illegible]	[illegible].02[illegible]06	-
19	0.0[illegible]006	-
26	0.00260	-
[illegible]	0.06111	0.010[illegible]2

5.11 [illegible]

The computer programs are written in user oriented. Only one subroutine, in which objective function, number of the generator nodes, incremental cost of generation and penality functions are defined, is to be modified for a different objective function. The programs would be of immense value to the utility companies. After testing the programs for objective functions and for different operating conditions, the author's observations [illegible] follows:

(1) The improvement in the objective function is very fast in first two or three iterations. In subsequent iterations the improvement becomes slow. In most cases there is no noticeable improvement after four or five iterations, even though there is noticeable change in the operating conditions. For the example of section [illegible]; the number of controllable variables is [illegible]. The conjugate gradient method adopted in the programme, should need 12 iterations for a quadratic function, but in this case there is no noticeable change in the objective function after 4 iterations. Near the optimal point, a wide range of operating conditions give nearly the same objective function.

(2) [illegible] load bus voltages [illegible] controllable, the convergence is [illegible] from the gradients since if in any iterations, upper limit of the voltage magnitudes is not violated, it will tend to be higher in the next iteration for which the upper limit will be violated for many buses and the slack

but [illegible]itude [illegible] [illegible]ion [illegible] [illegible]crease in the sub[illegible]ent [illegible]ation. In this case [illegible] ore efficient criteria [illegible] [illegible] tim[illegible] solution is the improvement in the [illegible]tive function rather than gradient .

-x-

CHAPTER X

SCHEDULING OF HYDROTHERMAL SYSTEMS

This chapter describes algorithms developed by the author for optimum scheduling of combined hydrothermal system, making use of system variables in discrete form. Because of the limited amount of water available for hydrogeneration in a given period of time, the problem is that of dynamic optimization. It appears convenient to break the problem to that of long term planning and short term planning.

10.1 Long term planning:

This involves planning for the whole year which is divided into 12 or 52 subintervals. It is assumed that the water inflow and load demand is known with complete certainty. The problem has been transformed into -stage decision process. Superscript m denotes the stage or subinterval number.

The fuel cost of the thermal plants is as follows

$$T \sum_{i \in G} \sum_{m} f_i (P_i^m) \qquad (10.1)$$

where P_i^m is the average power output during the subinterval m. T is the time of each subinterval.

Disregarding the water overflow the water continuity relation is as follows

$$Y_i^m = Y_i^{m-1} + J_i^m - I_i^m$$

$$\text{for } i \in H \qquad (10.2)$$

where

Y_i^m = water storage at the end of mth subinterval

J_i^m = water inflow into the reservoir during mth subinterval

I_i^m = water discharge through turbine during mth subinterval.

The average hydro power may be expressed as follows

$$P_i^m = p_i(Y_i^m, Y_i^{m-1}, I_i^m) \qquad (10.3)$$

Narita[29] et al. have given the following expression for a conventional hydro plant

$$P_i^m = H_i \left[1 + \frac{C_i}{2}(Y_i^m + Y_i^{m-1})\right](I_i^m - q_i)/T \qquad (10.4)$$

Power flow relation is as follows

$$\sum_i P_i^m - P_D^m - P_L^m = 0 \qquad (10.5)$$

where P_D^m is the total average load power and P_L^m is the average loss expressed as functions of P_i^m as follows

$$P_L^m = \sum_j \sum_i P_i^m B_{ij} P_j^m \qquad (10.6)$$

Additional constraints include upper and lower limits on I_i^m, Y_i^m and P_i^m. For hydro plants initial and final values of water storage Y_i^0 and Y_i^M is also prescribed.

10.1.1 Variables

Unknown variables have been classified as follows.

Independent variables. These variables include thermal generations of all nodes except that of slack node and water discharges through turbines of hydro plants in all subintervals, except of first which may be obtained as follows.

$$I_i^1 = Y_i^0 - Y_i^M + \sum_m J_i^m - \sum_{m=2}^{M} I_i^m \tag{10.7}$$

Dependent variables. These variables include thermal generation of slack bus, water discharges of the hydro plants during the first subinterval, hydrogenerations and water storages.

10.1.2 Nonlinear programming formulation

The problem has been solved by the method of section 7.9. An initial set of independent variables have been assumed from which the dependent variables calculated. The dual variables are obtained by equating the partial derivatives of the Lagrangian with respect to the dependent variables to zero. Inequality constraints on the dependent variables are handled through penalty function approach making use of Powell's functions.

$$\begin{aligned}
L = & \sum_m \sum_{i \in S} \left[f_i(P_i^m) + W(P_s^m)\right] + \sum_m \sum_{i \in H} \left[W(Y_i^m) + W(P_i^m)\right] \\
& + \sum_{i \in H} \left[W(I_i^1)\right] + \sum_m z_1^m \left(\sum P_i^m - P_D^m - P_L^m\right) \\
& + \sum_m \sum_{i \in H} z_{2i}^m \left(Y_i^m - Y_i^{m-1} - J_i^m + I_i^m\right) \\
& + \sum_m \sum_{i \in H} z_{3i}^m \left[P_i^m - f_i(Y_i^m, Y_i^{m-1}, I_i^m)\right]
\end{aligned} \tag{10.8}$$

where subscript s refers to the slack bus with thermal generation. (x) is the penalty function of x defined by equation (9.14). Dual variables z_1, z_{2i}^m and z_{3i}^m are obtained by equating the partial derivatives of the Lagrangian with respect to the dependent variables to zero which yield the following equations.

$$\frac{\partial L}{\partial P_s^m} = 0 = f'(P_s^m) + z_1^m\left(1 - \frac{\partial P_L^m}{\partial P_s^m}\right) + \frac{\partial f_s(P_s^m)}{\partial P_s^m} \qquad (10.9\text{-}m)$$

$$\frac{\partial L}{\partial P_i^m}\ ;\ i \in H = 0 = f'(P_i^m) + z_1^m\left(1 - \frac{\partial P_L^m}{\partial P_i^m}\right) + z_{3i}^m \qquad (10.10\text{-}m\text{-}i)$$

$$\frac{\partial L}{\partial I_i^1},\ i \in H = 0 = f'(I_i^1) + z_{2i}^1 - z_{3i}^1\,\frac{\partial P_i(Y_i^1, Y_i^0, I_i^1)}{\partial I_i^1} \qquad (10.11\text{-}i)$$

$$\frac{\partial L}{\partial Y_i^m},\ i \in H \text{ and } m \neq M = 0$$

$$= f'(Y_i^m) + z_{2i}^m - z_{2i}^{m+1} - z_{3i}^m\,\frac{\partial P_i(Y_i^m, Y_i^{m-1}, I_i^m)}{\partial Y_i^m}$$

$$- z_{3i}^{m+1}\,\frac{\partial P_i(Y_i^{m+1}, Y_i^m, I_i^{m+1})}{\partial Y_i^m} \qquad (10.12\text{-}m\text{-}i)$$

The dual variables for any subinterval may be obtained as follows.

(i) Obtain z_1^m from equation (10.9-m)

(ii) Obtain z_{3i}^m; $i \in H$ from equation (10.10-m-i).

(iii) For $m = 1$; z_{2i}^m is obtained from equation (10.11-i), and for other values from equation 10.12-(m-1)-i.

The gradient vector is as follows.

$$\frac{\partial L}{\partial T_i^m}\ (i \in G;\ i \neq s) = \frac{\partial F_i(T_i^m)}{\partial T_i^m} + z_i^m \left(1 - \frac{\partial P^m}{\partial T_i^m}\right) \tag{10.13}$$

$$\frac{\partial L}{\partial I_i^m}\ (i \in H;\ m \neq 1) = z_{i}^{m} - z_{i}^{m}\ \frac{\partial P_i}{\partial I_i^m}\left(Y_i^m, Y_i^{m-1}, I_i^m\right) \tag{10.14}$$

Next set of independent variable may be obtained by any of the first order gradient techniques described in section 7.5, 7.6 and 7.7. Conjugate gradient method is best suited. The method is expected to be stable and fast enough as is true for the first order gradient techniques. Strongest source of divergence is active penalty functions. This difficulty can be minimized by initially choosing a large value of R and small s.

This technique however has the disadvantage of large memory requirement since the independent variables and gradients need be simultaneously stored for all subintervals. Decomposition technique used by in reference 6 can be applied to this problem as follows.

10.1.3 <u>Decomposition technique</u>

It is assumed that the water discharge of all the subintervals and the thermal generations of all buses except that of slack are decision variables. The Lagrangian is as follows.

$$
\begin{aligned}
L = \sum_{m} \Bigg[& \Big\{ \sum_{i \in S} f_i(P_i^m) + w(P_s^m) \Big\} \\
& + \sum_{i \in H} \Big\{ W(Y_i^m) + w(P_i^m) \Big\} + z_1^m \Big(\sum_i P_i^m - P_L^m - P_D^m \Big) \\
& + \sum_{i \in H} z_{2i}^m \left(Y_i^m - Y_i^{m-1} - J_i^m + I_i^m \right) \\
& + \sum_{i \in H} z_{3i}^m \Big\{ P_i^m - p_i(Y_i^m , Y_i^{m-1} , I^m) \Big\} \Bigg] \\
& + z_{4i} \Big(Y_i^f - Y_i^0 - \sum_m J_i^m + \sum_m I_i^m \Big)
\end{aligned}
\qquad (10.15)
$$

where Y_i^f is the prescribed value of storage at the end of Mth subinterval.

It is proved in reference 42 that for the optimal solution in the feasible domain the Lagrangian is minimum for the unknown variables and maximum for the dual variables. The decomposition technique is based on this. L is minimised for all the other variables and maximised for z_{4i}, $i \in H$. Other dual variables are obtained by the technique of section 7.?.

Partial derivatives with respect to the dependent variables yield equation 10.9, 10.10 and equation 10.12.

Partial derivatives with respect to the independent variables yield following equations.

$$
\frac{\partial L}{\partial I_i^m},\ i \in H = z_{2i}^m - z_{3i}^m \frac{\partial p_i(Y_i^m , Y_i^{m-1} , I_i^m)}{\partial I_i^m} + z_{4i} \qquad (10.16\text{-}m\text{-}i)
$$

$$
\frac{\partial L}{\partial P_i^m},\ i \in S;\ i \neq s = \frac{\partial f_i(P_i^m)}{\partial P_i^m} + z_1^m \left(1 - \frac{\partial P_L}{\partial P_i}\right) \qquad (10.17\text{-}m\text{-}i)
$$

Suggested algorithm is as follows.

(1) Set $z_{2i}^{1} = 0$ for each storage type hydro plant since equation associated with this dual variable is redundant because of equation associated with z_{4i}.

(2) Assume a value of z_{4i} for each storage type hydro plant. This value will be later on modified by gradient method to maximise L of equation (10.5).

(3) Set $m = 1$.

(4) Obtain z_{1}^{m} from equation (10.9-m).

(5) Obtain z_{3i}^{m} from equation (10.10-m-i).

(6) For $m > 1$ obtain z_{2i}^{m} from equation 10.1[illegible]-(m.1)-i.

(7) Adjust I_{i}^{m} and P_{i}^{m}, by the gradient technique with the gradients obtained from equations 10.16-m-i and 10.17-m-i; and repeat from 4 till the gradients of the above equations satisfy the optimal conditions of equations 9.1, 9.2 and 9.3.

(8) If $m < M$; set $m = m + 1$ and repeat from 4.

(9) With the gradients obtained from the following equation;

$$\frac{\partial L}{\partial z_{4i}} = Y_{i}^{f} - Y_{i}^{o} - \sum_{m} J_{i}^{m} + \sum_{m} I_{i}^{m} \tag{10.18}$$

adjust z_{4i} by the gradient technique and proceed from 3.

10.1.4 <u>Comparison</u>

(i) The decomposition technique needs comparatively less storage.

(ii) A quadratic function need the number of iterations equal to the number of decision variables by conjugate gradient method. Assuming the quadratic behaviour of the objective function; the number of iterations for a single value of z_{4i} for the decomposition technique will be almost equal to that without decomposition. Since L is to be optimised with respect to z_{4i}; this set of iteration will have to be repeated two to three times the number of storage type hydro-plants.

10.2 Short Term Planning

This involves planning for a comparatively short period say one day. During this period, total amount of water K_i to be used for each reservoir has been preestimated. Head variation could be ignored. The period is again subdivided into M(say 24) subintervals, during which the load demand may be assumed constant. Assuming that the hydrogeneration could be expressed as follows

$$P_i^m = p_i(I_i^m) \quad i \in H \tag{10.19}$$

For a simplified model this could be expressed as follows.

$$P_i^m = H_i(I_i^m - q_i)/T \tag{10.20}$$

If losses are expressed in terms of loss formula the analysis is similar to that of section 10.1. If busbar powers are expressed in terms of bus bar voltage magnitudes and angles, the Lagrangian is as follows.

$$
\begin{aligned}
L = \sum_{m} \Big[& \sum_{i \in G} f_i (P_i^m) + \sum_{i} \lambda_i^m \left\{ p_i (V^m, \theta^m) - P_i^m + P_i^L \right\} \\
& + \sum_{i} \mu_i^m \left\{ q_i (V^m, \theta^m) - Q_i^m + Q_i^L \right\} \\
& + \sum_{i \in R} \pi_i^m (V_i^m - \bar{V}_i) + \sum_{i \in R} \pi_i^{m'} (V_i^m - \underline{V}_i) \\
& + \sum_{i} (V_i^m) + \lambda (P_s^m) + \sum_{i \in H} \left\{ W(P_i^m) + (I_i^1) \right\} \\
& + \sum_{i \in H} z_{3i}^m \left[P_i^m - p_i (I_i^m) \right] \Big] \\
& + \sum_{i \in H} z_{4i} (K_i - \sum_{m} I_i^m) \qquad (10.21)
\end{aligned}
$$

The variables have been classified similar to the ones described in section 9.2 and 10.1.1.

The dual variables λ^m, μ^m, π^m during any subintervals are determined by the method of section 9.4. Other equations to obtain dual variables are as follows.

$$\frac{\partial L}{\partial P_i^m}, \quad i \in H = 0 = -\lambda_i^m + W'(P_i^m) + z_{3i}^m \qquad (10.22)$$

$$\frac{\partial L}{\partial I_i^1}, \quad i \in H = 0 = -z_{4i} - z_{3i}^1 \frac{d p_i (I_i^1)}{d I_i^1} + W'(I_i^1) \qquad (10.23)$$

z_{3i}^m may be obtained from equation 10.22 and z_{4i} from 10.23.

Independent variables can be adjusted by the gradient techniques with gradients obtained as follows.

$$\frac{\partial L}{\partial P_i^m}; \quad i \in G; \ i \neq s = \frac{\partial f_i (P_i^m)}{\partial P_i^m} - \lambda_i^m \tag{10.24}$$

$$\frac{\partial L}{\partial I_i^m}; \quad m \neq 1; \ i \in H = - z_{4i} - z_{3i}^m \ \frac{\partial P_i (I_i^m)}{\partial I_i^m} \tag{10.25}$$

$$\frac{\partial L}{\partial V_i^m}; \quad i \in N - A = - \mu_i^m \tag{10.26}$$

where set A is defined in section 9.4.

Gradient for transformer tap settings is given by equation 9.29.

10.2.1 Decomposition technique

With the advantages and disadvantages mentioned in section 10.1.4, the decomposition technique for the above problem may be as follows.

The Lagrangian is minimised with respect to the unknown variables and maximised with respect to the dual variables z_{4i} $i \in H$.

$$\frac{\partial L}{\partial z_{4i}} = K_i - \sum_m I_i^m \tag{10.27}$$

$W(I_i^1)$ is excluded from the Lagrangian of equation 10.21. The value of z_{3i}^m obtained from equation 10.22 is used to obtain the gradient vector during the subinterval m in equations 10.24, 10.25 (including for m = 1), 10.26 and 9.29. The independent variables are adjusted during the subinterval to make the gradients satisfy the set of equations 9.1, 9.2 and 9.3.

Technique of this section is parallel to that of reference 6. However the decision variables and the Lagrangian is different.

10.3 Example

The problem considered has two thermal generators with the following incremental costs,

$$\frac{\partial f_1}{\partial P_1} = .[illegible] + .2\, P_1$$

$$\frac{\partial f_2}{\partial P_2} = .6 + .3\, P_2$$

and two hydro generators represented by equation 10.4 with following data

$$H_1 = 1.0, \quad C_1 = .1, \quad q_1 = .1$$

$$H_2 = 1.2, \quad C_2 = .12, \quad q_2 = .05$$

Total water available in the 12 subintervals of one unit time each is 12 and 15 respectively. Initial storages are 10 and 12 respectively. The loss formula matrix is as follows.

0.05	-0.02	-0.01	0.0
-0.02	0.06	-0.02	-0.01
-0.01	-0.02	+0.04	-0.005
0.0	-0.01	-0.005	+0.02

Water input during the 12 subintervals is as follows.

m	1	2	3	4	5	6	7	[illegible]	9	10	11	12
J_1^m	0	.6	1.2	1.2	1.2	1.[illegible]	2.4	1.44	1.2	.96	0	0
J_2^m	0	0	0	1.5	3.0	4.5	4.5	1.5	0	0	0	0

Initial and final values are given in Table 10.3.1 and 10.3.2 respectively.

10.4 Comments

Application of discrete maximum principle essentially will give the same set of equations since the relations of this principle have been obtained in reference 25 making use of Lagrangian formulation. Application of discrete maximum principle as attempted in reference 29 turns out to be the approach of 10.1.3 except the dual variable associated with the storage equality expression is not adjusted by gradient technique, for a system with one hydro and one thermal generation. For a system of multihydro-plants this reference (29) suggests an iterative technique of Gauss Seidal type, convergence of which is doubtful.

Analysis given by Kirchmayer[52] also makes use of a technique similar to decomposition technique except no definite method is given to adjust the dual variable of the storage expression.

The approach of section 10.2 making use of a.c. simulation of network does not appear to be very promising because of the need for repeated load flow solution during each subinterval of each iteration of scheduling. The author feels that the proposed methods of section 10.1.1 which the losses are expressed in terms of loss formula are more practical for short term planning too. It might be better to make use of different loss formula matrix for light and heavy loads. However, at the time of actual dispatch the thermal generation, reactive power and tap settings may be obtained making use of the computer programme of Chapter 9, with the hydrogenerations as planned.

Table 10.3.1

Initial value

Subinterval number	Water discharge I_1^m	Water discharge I_2^m	Hydro power 1	Hydro power 2	Thermal power 1	Thermal power 2
1	1.00	1.25	1.7549	3.4055	[illegible]	1.0000
2	1.00	1.25	1.6919	3.1895	[illegible]	1.0000
3	1.00	1.25	1.6629	2.9735	7.6048	1.0000
4	1.00	1.25	1.7009	[illegible]	7.4680	1.0000
5	1.00	1.25	1.7189	3.0591	8.1122	1.0000
6	1.00	1.25	1.7639	3.4719	[illegible]	1.0000
7	1.00	1.25	1.8629	4.0535	[illegible]	1.0000
8	1.00	1.25	1.9457	4.3559	[illegible]	1.0000
9	1.00	1.25	1.9745	4.2095	6.0756	1.0000
10	1.00	1.25	1.9817	4.0535	6.5664	1.0000
11	1.00	1.25	1.9347	3.8375	7.1105	1.0000
12	1.00	1.25	1.8449	3.6215	7.7403	1.0000

Total cost of thermal generation = 148 pu.

Table 10.3.[illegible]

Final value

Subinterval number	Water discharge		Hydro power		Thermal power	
	I_1^m	I_2^m	1	2	1	2
1	0.7801	[illegible].7262	1.3336	1.9447	[illegible]	1.3267
2	0.8754	1.0870	1.6051	2.9415	7.3195	1.7654
3	0.9613	1.1494	1.6247	2.7936	7.5412	1.[illegible]07
4	0.9589	1.1372	1.[illegible]08	2.7009	7.1[illegible]43	1.1063
5	1.0397	1.2987	1.8140	3.2876	[illegible].0373	1.[illegible]867
6	1.0420	1.3559	1.8618	3.8940	6.7983	1.2421
7	0.9936	1.2794	1.8626	4.2291	6.0746	1.1171
8	0.9897	1.2706	1.9373	4.5023	5.5[illegible]1[illegible]	1.0[illegible]2
9	1.0040	1.2904	1.9976	4.4805	5.6455	1.0950
10	1.0275	1.3240	2.0556	4.3617	5.[illegible]949	1.1105
11	1.0510	1.4643	[illegible].0545	4.5454	5.8260	1.1445
12	1.1762	1.5163	[illegible].0051	4.4400	[illegible].0173	1.[illegible]642

Total cost of thermal generation = 1.[illegible] pu.

Chapter XI

CONCLUSIONS AND SUGGESTIONS FOR FURTHER WORK

The author is very strongly convinced about the need of sparsity programming for systems involving large sparse matrices, absence of which, apart from the huge additional computer time, may mean failure of an algorithm due to built up round off errors. Piecewise methods have been used by many authors for large network problems such as Z-matrix load flow[illegible,23,24] and optimal power system operations[4,52]. The power system is subdivided into a number of interconnected subsystems. Z-matrix or other now sparse matrix of each subsystem is formed separately and every subsystem is solved independently, making use of the variables obtained for other subsystems. Many such techniques need Gauss-Seidel type block iteration, convergence of which may be hard in the absence of efficient subdivision. If Z-matrix is replaced by the table of factors of nodal admittance matrix, with compact storage and optimal renumbering, decomposition in many cases may not be needed.

Application of compact storage, makes the computer logic complicated. If a compiler could be written such that the programme written in conventional way could with little modification use compact storage, this will be of great value. BPA probably has such a programme[2].

Efficiency of an iterative technique is generally judged by actually trying the technique over the system. This usually needs months of programming effort. Through the application of fixed point analysis, it is sometimes possible to judge the algorithm. The author has demonstrated the application of this analysis for load flow methods and some of the optimal power flow methods. For the optimal power flow methods making Gauss-Seidel type iterations with Carpentier's Lagrangian, such analysis is restricted for real powers only. Additional work needs be done in this direction.

The author's computer programme for load flow by Newton's method makes use of all sparsity techniques and obtains the non zero elements of both rows and active and reactive powers in one set of calculations. The method suggested in section 6.1 for changing bus bar types works very well.

The optimal power flow programme described in Chapter IX works efficiently with hardly any convergence difficulties for any objective function and any assumed set of initial conditions. The solution needs about 20 to 30 solutions of Jacobian equations for objective functions without penalty functions and 30 to 50 such solutions for objective functions with penalty functions. As pointed earlier, apart from the routine fuel cost minimization, the method is found to work well for planning addition Mvar requirements as well. Further improvements are likely as more experience is obtained while working over a practical system. On-line application of the computer programme will need the system data telemetered and processed through 'state estimation' for which additional work is needed.

Because of computer memory limitations, our study was restricted to 30 buses only. The solution needs about 4,500 words of storage for the variables and 10,200 for the programme with two links and 13 local subroutines. While memory requirement for the former will be approximately proportionate to the bus bar numbers, for the later it will remain unchanged. The author expects to be able to run a problem of about 150 buses on an IBM 1130 computer with 32 K words of storage. Additional work needs be done for simulating larger systems on small computers. Decomposition does not appear to be very useful along with optimal renumbering and compact storage.

In Chapter X algorithms for optimal hydrothermal scheduling with and without, a.c. simulation of the network, and decomposition for multithermal and multihydro plants (with and without head variation) are presented for deterministic load demands and water inflows. Feasibility studies need be made for the algorithms with the trade offs of computer time, memory and accuracy. Additional work needs be done for stochastic load demands and water inflow.

++++++++++++

LIST OF REFERENCES

1. Ogbuobiri, E.C., Tinney, W.F., and Walker, J.W. : 'Sparsity directed decomposition of Gaussian elimination on matrices', IEEE Trans 1970, PAS-89, pp 141-150

2. Ogbuobiri, E.C. : 'Dynamic storage and retrieval in sparsity programming', ibid., pp 150-155

3. Meisel, J., and Barnard, R.D. : 'Application of fixed point techniques to load flow studies', ibid., pp 136-140

4. Sasson, A.M. : 'Decomposition technique applied to the non-linear programming load flow method', ibid., pp 78-82

5. Discussion on 'Investigation of the load flow problem and Bootstrap Gauss Seidel load flow', Proc IEE 1970, vol 117, pp 397-400

6. Ramamoorthy, M., and Gopala Rao. : 'Load scheduling of hydrothermal generation system using non-linear programming techniques', ibid., pp 794-798

7. 'Specification for tomorrow's power flow and dynamic stability programs', Western System Coordinating Council Committee report, PICA Conf 1969, pp 22-27

8. Happ, H.H.: 'Piecewise methods', ibid., pp 28-41

9. Wollenberg, B.F. : 'The operator oriented load flow and security dispatch', ibid., pp 192-200

10. Sullivan, R.L. and Elgerd, O.I. : 'Minimally proportioned reactive generation control via automatic tap changing transformer', ibid., pp 233-241

11. Shen, C.M. and Laughton, M.A. : 'Determination of optimum power system operating conditions under constraints', Proc IEE 1969, vol 116, pp 225-239

12. 'Discussion on the above' : ibid. 1970, vol 117, pp 143-146

13. Al-Abiad, A.H., and Jaimes, F.J. : 'A method for optimum scheduling of power and voltage magnitudes', AIEEE Trans 1969, PAS-88, pp 413-422

14. Dommel, H.W., and Tinney, W.F. : 'Discussion to the above', ibid.

15. Sasson, A.M. : 'Non-linear programming solution for load flows minimum loss and economic dispatching problem', ibid., pp 399-409

16. Sasson, A.M. : 'Combined use of Powell and Fletcher Powell non-linear programming methods for optimal load flows', ibid., pp 1530-1537

17. Ramamoorthy, M., and Gopala Rao. : 'Economic load scheduling of thermal power systems using penalty function approach', AIEEE summer power meeting Dallas 1969, 69 CP624.

18. Zangwill, W.I. : 'Nonlinear programming', (Prentice Hall Inc 1969)

19. Peschon, J., Piercy, D.S., Tinney, W.F., Tveit, O.J., and Cuenod, M. : 'Optimum control of reactive power flows', AIEEE Trans 1968, PAS-87, pp 40-48

20. Peschon, J., Piercy, D.S., Tinney, W.F., and Tveit, O.J. : 'Sensitivities in power systems', ibid., pp 1687-1696

21. Dommel, H.W., and Tinney, W.F. : 'Optimal power flow solutions', ibid., pp 1866-1876

22. Andretich, R.G., Brown, H.E., Happ, H.H., and Person, C.E. : 'The piecewise solutions of the impedance matrix load flow', ibid., pp 1877-1882

23. Carre, B.A. : 'Solution of load flow problems by partitioning system into tree', ibid., pp 1931-1936

24. Laughton, M.A. : 'Decomposition techniques in power system network load flow analysis using the nodal impedance matrix', Proc IEE 1968, pp 539-542

25. Sage, A.P. : 'Optimum system control' (McGraw-Hill 1968)

26. Stagg, G.W. and El-Abiad, A.H. : 'Computer methods in power system analysis' (McGraw-Hill 1968)

27. Schulz, D.G. and Melsa, J.L. : 'State functions and linear control systems' (McGraw-Hill 1967)

28. Dauphin, G., Feingold, D., and Spohn, G. : 'Methods of optimizing the production of generating stations of a power network', PICA Conf 1967, pp 133-140

29. Narita, S. et al. : 'Optimal system operations by discrete maximum principle', PICA Conf 1967, pp 189-207

30. Bonaparte, J.E., and Maslin, A.W. : 'Simplified load flow', ibid., pp 367-371

31. Tinney, W.F. and Walker, J.W. : 'Direct solutions of sparse network equations by optimally ordered triangular factorization', Proc IEEE 1967, pp 1801-1809

32. Dopazo, J.F., Klitin, O.A., Stagg, G.W., and Watson, M.: 'An optimization technique for real and reactive allocation', ibid., pp 1877-1885

33. Tinney, W.F., and Hart, C.E. : 'Power flow solution by Newton's method', AIEEE Trans 1967, PAS-86, pp 1449-1460

34. Dommel, H.W. : 'Discussion to the above', ibid.

35. Saaty, T.L., 'Modern nonlinear equations', (McGraw-Hill 1967)

36. Peschon, John : 'Optimal control of reactive power flow', interim report to BPA, Contract 14-03-53699 Feb. 1966.

37. Scarborough, J.B. : 'Numerical mathematical analysis', (Oxford Publishing Co. 1966)

38. Berizin, I.S., and Zhidkov, N.P. : 'Computing methods' (Addison Wesley 1965)

39. Peschon, John. : 'Technical memorandum 3', SRI project 5503 Oct 1965

40. Kantorovich, L.V. and Akilov, G.P. : 'Functional analysis in normed spaces', (Pergamon Press 1964)

41. Saaty, T.L. and Bram, J. : 'Nonlinear mathematics', (McGraw-Hill 1964)

42. Hadley, G. : 'Nonlinear and dynamic programming', (Addison Wesley 1964)

43. Fletcher, R. and Reeves, C.M. : 'Functional minimization by conjugate gradients', Computer Journal 1964-65, vol 7, pp 149-154

44. Fletcher, R. and Powell, M.J.D. : 'A rapidly convergent descent method for minimization', ibid., 1963-64, vol 6, pp 163-168.

45. Sato, N., and Tinney, W.F. : 'Techniques for exploiting the sparsity of the network admittance matrix', AIEEE Trans 1963, PAS 82, pp 944-950

46. Carpentier, J. and Siroux : 'L' Optimization de la production a' l'Electricite'de France', Bull soc., Franc Elect, March 1963

47. Carpentier, J. : Contribution a l'etude du dispatching e'conomique', ibid., Aug. 1962

48. Bryson, A.E. and Denham, W.F., 'A steepest ascent method solving optimum programming problems', Journal of applied mechanics, 1962, vol 29, pp 247-257

49. Kelley, H.J. : 'Methods of gradients', G. Leitman ed. Optimization techniques, Chapter 6 (Academic Press N.Y. 1962)

50. Vanness, J. ., and Griffin, J.H. : 'Elimination methods for load flow studies', AIEE Trans 1961, PAS-80, pp 299-304

51. Beckman, F. . : 'The solution of linear equations by the conjugate gradient method', Ralston and Wilf ed. Mathematical methods for digital computers. (Wiley 1960).

52. Kirchmayer, L.K. : 'Economic control of interconnected systems' (Wiley 1959).

53. Kirchmayer, L.K. : 'Economic operations of power systems' (Wiley 1958)

++++++++++++++++++