

Spring 5-31-1988

On conventional and harmonic load flow analysis

Ranjit B. Pradhan
New Jersey Institute of Technology

Follow this and additional works at: <https://digitalcommons.njit.edu/theses>



Part of the [Electrical and Electronics Commons](#)

Recommended Citation

Pradhan, Ranjit B., "On conventional and harmonic load flow analysis" (1988). *Theses*. 1402.
<https://digitalcommons.njit.edu/theses/1402>

This Thesis is brought to you for free and open access by the Electronic Theses and Dissertations at Digital Commons @ NJIT. It has been accepted for inclusion in Theses by an authorized administrator of Digital Commons @ NJIT. For more information, please contact digitalcommons@njit.edu.

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

ABSTRACT

Title of thesis: On Conventional And Harmonic Load Flow
Analysis

Ranjit Pradhan, Master of Science in Electrical
Engineering, 1988

Thesis directed by: Dr Walid Hubbi, Assistant
Professor, Electrical Engineering department

In conventional power flow analysis, simultaneous linear equations numbering in thousands must be solved repeatedly and in many instances on-line. Luckily the matrix of coefficients of these equations is sparse with a few elements in each row. Therefore for a faster solution and to save computer memory, sparsity must be exploited and an optimal ordering of the equations must be sought. When considering harmonics, the number of the simultaneous equations to be solved is multiplied and therefore, considering sparsity and optimal ordering is doubly important.

In this thesis a software package for optimizing computation time and memory requirements through exploitation of sparse matrices for Power System Bus Equations was developed. Exploitation of sparsity was accomplished by utilizing a packed storage scheme and special programming.

HARMFLO package was obtained from E.P.R.I. and was used to compare the results obtained using the developed package with those obtained from HARMFLO. The equations that must be solved for a harmonic load flow analysis are detailed and an example is given. A subroutine to modify the admittance matrix in accordance with the order of harmonics was developed.

ON CONVENTIONAL AND HARMONIC LOAD FLOW ANALYSIS

by

Ranjit B. Pradhan

Thesis submitted to the Faculty of the Graduate School
of the New Jersey Institute of Technology in partial
fulfillment of the requirements for the degree of
Master of Science in Electrical Engineering
1988

APPROVAL SHEET

Title of Thesis: On Conventional And Harmonic Load
Flow Analysis

Name of Candidate: Ranjit B. Pradhan
Master of Science
in Electrical Engineering, 1988

Thesis and Abstract Approved :

Dr. Walid Hubbi
Assistant Professor
Department of Electrical Engineering

Date

Department of Electrical Engineering

Date

VITA

Name: Ranjit Bhalchandra Pradhan

Permanent Address:

Degree and date to be conferred: M. S. E. E., 1988

Date of Birth:

Place of birth:

Secondary Education: Jaihind College, India, 1984

Collegiate institutions attended	Dates	Degree
R. V. College Of Eng.	1979-84	B.S.E.E.
N. J. Institute Of Tech.	1985-88	M.S.E.E.

Major: Electrical Engineering

Minor: Computer Sciences

Positions held: 1986 Research Assistant for EE department at NJIT
1987 Graduate Assistant for the Media Center at NJIT

ACKNOWLEDGEMENT

I wish to place on record my sincere and deep gratitude to Dr. Walid Hubbi, Assistant Professor in the Electrical Engineering department at N.J.I.T. for his constant and invaluable assistance, encouragement and support in the accomplishment of this thesis. Without his expert and skillful guidance this thesis would not have taken this final form.

I would also like to express my appreciation to my wife, Medini for her timely help in getting the thesis together and also to my parents without whose help studying in the U.S. would have just been a dream.

CONTENTS

Chapter	Page
ACKNOWLEDGEMENT.....	i
LIST OF FIGURES.....	vi
LIST OF TABLES.....	vii
1. INTRODUCTION.....	1
1.1 Overview.....	2
1.2 Load Flow Equations.....	3
1.3 Iterative Solutions To the LF Equations....	6
1.4 Exploiting Sparsity.....	11
1.5 Thesis Outline.....	13
2. CONVENTIONAL LOAD FLOW ALGORITHM.....	16
2.1 Algorithm Design.....	16
2.2 Conceptual Model.....	18
2.2.1 Implemented Storage Scheme.....	20
2.2.2 Programming Techniques.....	21
A. Simulation And Ordering.....	21
B. Reduction.....	22
C. Direct Solution.....	22
3. CONVENTIONAL LOAD FLOW PROGRAM.....	24
3.1 Exploiting sparsity and optimal ordering (Program JEET).....	25
3.1.1 Main Routine - initializing and reading network data.....	25
A. Description.....	25
B. List of the main routine.....	28

3.1.2	Subroutine PQCAL - calculation of P and Q.....	33
	A. Description.....	33
	B. List of subroutine PQCAL.....	35
3.1.3	Subroutine DELTA - calculating the mismatches.....	36
	A. Description.....	36
	B. List of subroutine DELTA.....	38
3.1.4	Subroutine TESTPO - testing for convergence or divergence.....	39
	A. Description.....	39
	B. List of subroutine TESTPO.....	42
3.1.5	Subroutine JACOB - calculation of the jacobian matrix.....	44
	A. Description.....	44
	B. List of subroutine JACOB.....	46
3.1.6	Subroutine DATAGEN - formation of packed density scheme.....	48
	A. Description.....	48
	B. List of subroutine DATAGEN.....	51
3.1.7	Subroutine RADO - optimal ordering scheme.....	55
	A. Description.....	55
	B. List of subroutine RADO.....	58
3.1.8	Subroutine FACTO - factorisation of the optimally ordered scheme.....	61
	A. Description.....	61
	B. List of subroutine FACTO.....	64
3.1.9	Subroutine SLAD - Adjustment for slack bus.....	66
	A. Description.....	66
	B. List of subroutine SLAD.....	68

3.1.10	Subroutine UTION - final solution vector.....	69
	A. Description.....	69
	B. List of subroutine UTION.....	71
3.2	Exploiting sparsity and not optimal ordering (program NOOPTIMAL).....	72
3.2.1	Modified Subroutine DATAGEN - bypassing optimal ordering.....	72
	A. Description.....	72
	B. List of modified subroutine DATAGEN.....	75
3.3	Testing under different loading conditions (program DIFFLOAD).....	77
3.3.1	Description.....	77
3.3.2	List in part of program DIFFLOAD....	79
4.	HARMONIC LOAD FLOW ANALYSIS.....	82
4.1	AC System Harmonics.....	82
4.2	Summary Of Harmonic Studies.....	83
4.2.1	Reformulation of NR method to allow for harmonics.....	86
4.2.2	Effect of harmonics on power studies.....	102
4.3	Recalculating line elements at different harmonic frequencies (program HARMONIC).....	104
4.3.1	Description.....	104
4.3.2	List of the program HARMONIC.....	106
4.4	Evaluation of HARMFLO.....	109
4.4.1	Special environment instructions...	109
4.4.2	System size.....	110

4.4.3	Time and memory requirement.....	110
4.4.4	Data preparation.....	110
5.	RESULTS AND OBSERVATIONS.....	121
5.1	The test systems.....	122
5.1.1	14 bus system.....	122
5.1.2	118 bus system.....	128
5.1.3	15 bus system.....	128
5.2	Interpretation of output.....	129
5.3	Observations and discussion.....	135
5.3.1	JEET.....	135
5.3.2	HARMFLO.....	137
5.3.3	NOOPTIMAL.....	138
5.3.4	DIFFLOAD.....	138
5.3.5	HARMONIC.....	139
APPENDIX A	Example of implemented storage scheme..	142
APPENDIX B	Description of parameters.....	144
APPENDIX C	Nomenclature.....	145
APPENDIX D	Example to calculate TG & H matrices.....	147
SELECTED BIBILOGRAPHY.....		154

LIST OF FIGURES

Figure	Page
1.1	Flow diagram of the basic Newton-Raphson load flow algorithm.....10
3.1	Flow chart of the Main routine.....27
3.2	Flow chart of the subroutine PQCAL.....34
3.3	Flow chart of the subroutine DELTA.....37
3.4	Flow chart of the subroutine TESTPO.....41
3.5	Flow chart of the subroutine JACOB.....45
3.6	Flow chart of the subroutine DATAGEN.....50
3.7	Flow chart of the subroutine RADO.....57
3.8	Flow chart of the subroutine FACTO.....63
3.9	Flow chart of the subroutine SLAD.....67
3.10	Flow chart of the subroutine UTION.....70
3.11	Flow chart of the modified subroutine DATAGEN.....74
3.12	Flow chart of the program DIFFLOAD.....78
4.1	Flow chart of the program HARMONIC.....105
5.1	The 14 node system network.....124

LIST OF TABLES

Table	Page
4.1 Harmonic power flow Jacobian related matrix dimensions.....	98
4.2 System size for HARMFLO version 4.0.....	114
4.3 HARMFLO control code 3 information.....	115
4.4 Allowed bus types (HARMFLO).....	115
4.5 Allowed bus sub-types (HARMFLO).....	116
4.6 Allowed combination of bus types and sub-types (HARMFLO).....	116
4.7 HARMFLO control code 4 information.....	117
4.8 HARMFLO control code 11 and 12 information.....	118
4.9 Specified and estimated parameters according to device types.....	119
4.10 HARMONIC control code 13 information.....	120
5.1 Input classification.....	123
5.2 Sample input file format (JEET, DIFFLOAD, NOOPTIMAL - 14 bus system).....	125
5.3 Sample input file format (HARMFLO - 14 bus system).....	127
5.4 Sample input file format (HARMFLO - 15 bus system).....	130
5.5 Sample output (JEET, DIFFLOAD, NOOPTIMAL)....	131
5.6 Sample output (HARMFLO).....	132
5.7 Sample output (HARMONIC).....	136
5.8 Comparison of results (HARMFLO & JEET).....	140
5.9 Comparison of results (JEET & NOOPTIMAL 14 bus system).....	141
5.10 Comparison of results (JEET & NOOPTIMAL 118 bus system).....	141

CHAPTER 1

INTRODUCTION

Power system harmonics have existed on power systems for a long time. The increasing use of nonlinear loads has been contributing to their increase. The resulting distorted voltage and current waveforms have been a source of major concern since they could have a damaging effect on both consumer load and power network. Hence harmonic load flow analysis has to be studied in a similar fashion to conventional load flow analysis.

Conventional load flow analysis deals with linear buses, while harmonic load flow analysis deals with linear and nonlinear buses. Nonlinear buses are those having nonlinear loads, exhibiting non-sinusoidal voltage and current characteristics.

This thesis work begins with a basic view of conventional load flow analysis and then goes on to discuss the modifications to accommodate for harmonic studies.

1.1 Overview

In a power system, power flows from the generating centers to the load centers. Now to study this flow, the complex power systems have to be simulated. This simulation and subsequent assesment of power flow in the lines subject to the boundry conditions of power supply and demand is commonly known as load flow analysis. Load flow study thus aims at arriving at a steady state nodal voltage solution of complete power networks. Once nodal voltages are known subject to certain boundry conditions, line flow can be easily calculated.

A power system is an ac network handling electrical energy. Every node or bus of the system is characterised with :

- P - Active Power
- Q - Reactive Power
- V - Voltage Magnitude
- θ - Voltage Phase Angle

Even though every bus of the system is associated with four variables P,Q,V and θ . Primarily only two of these variables are known at each bus. Based on the known variables the buses are classified as follows:

1. Slack Bus : V and θ known, P and Q unknown.

This bus supplies any additional generation needed beyond that specified for other generating buses to match the system load plus the transmission losses. This is virtually a generation bus with a specified magnitude of voltage but without a specified generation. This bus derives its existence from the fact that power losses are not known until the load flow study is completed [1].

2. Generator Bus : P and V known, Q and θ unknown.

This is the bus where the generation occurs. Thus the active power P at this bus is specified and the voltage magnitude V can be regulated using voltage regulators.

3. Load Bus : P and Q known, V and θ unknown.

A bus with no generation but from where the consumer demand is met is known as the load bus. Hence at such buses V and θ vary according to consumer demand.

1.2 Load Flow (LF) Equations

It has been stated that LF study virtually determines the complex voltages of the system buses for a particular system loading conditions. To do this with

the help of a digital computer, a set of equations expressing the active and reactive loadings of the buses in terms of complex voltages is needed. For a network having n nodes excluding ground, a set of following equations, one for each node, can be written

$$\bar{I}_1 = \bar{Y}_{11} \bar{V}_1 + \bar{Y}_{12} \bar{V}_2 + \dots + \bar{Y}_{1n} \bar{V}_n \quad (1.2-1)$$

$$\bar{I}_2 = \bar{Y}_{21} \bar{V}_1 + \bar{Y}_{22} \bar{V}_2 + \dots + \bar{Y}_{2n} \bar{V}_n \quad (1.2-2)$$

$$\begin{array}{ccccccc} \vdots & & \vdots & & \vdots & & \vdots \\ \vdots & & \vdots & & \vdots & & \vdots \\ \vdots & & \vdots & & \vdots & & \vdots \end{array}$$

$$\bar{I}_n = \bar{Y}_{n1} \bar{V}_1 + \bar{Y}_{n2} \bar{V}_2 + \dots + \bar{Y}_{nn} \bar{V}_n \quad (1.2-3)$$

The above equations can be written in the following compact form

$$\bar{I}_i = \sum_{m=1}^n \bar{Y}_{im} \bar{V}_m, \quad i = 1, 2, 3, \dots, n \quad (1.2-4)$$

where

\bar{I}_i = complex current entering in the i th bus

\bar{V}_m = complex voltage to ground of the bus m

\bar{Y}_{im} = complex admittance between buses i and m ; when

$i = m$ it is the driving point admittance,

otherwise it is the transfer admittance.

An over-bar indicates a complex or a phasor quantity. Matrices are denoted by boldface symbols. Superscripts, subscripts and Σ indices are not

explained after every equation where obvious.

In matrix form, equation (1.2-1) becomes

$$I = Y V \quad (1.2-5)$$

where I is a column matrix containing $\bar{I}_1, \bar{I}_2, \dots, \bar{I}_n$ as its elements. Similarly, V is also a column matrix with $\bar{V}_1, \bar{V}_2, \dots, \bar{V}_n$ as its elements. Admittances, both driving-point and transfer are contained within the admittance matrix Y .

In a power system load forecast, the power, active and reactive, at load buses is estimated. This makes the LF problem nonlinear. Had we been able to forecast nodal currents, the nodal voltages would be easily calculated using (1.2-5). The complex power input into a bus can be expressed as

$$P_i + jQ_i = \bar{V}_i \bar{I}_i^* \quad (1.2-6)$$

Substituting equation (1.2-4) into equation (1.2-6)

$$P_i + jQ_i = \bar{V}_i \sum_{m=1}^n \bar{Y}_{im}^* \bar{V}_m^*, \quad i=1, 2, \dots, n \quad (1.2-7)$$

where n is the number of buses and \bar{V}_i is the complex voltage to ground at bus i .

Separating the real and imaginary parts of (1.2-7)

gives

$$P_i = \text{Re} [\bar{V}_i \sum_{m=1}^n \bar{Y}_{im}^* \bar{V}_m^*] \quad (1.2-8)$$

$$Q_i = \text{Im} [\bar{V}_i \sum_{m=1}^n \bar{Y}_{im}^* \bar{V}_m^*] \quad (1.2-9)$$

1.3 Iterative Solutions To The LF Equations

Equation (1.2-7) represents a set of nonlinear simultaneous algebraic equations for which there is no closed form. These equations are to be solved iteratively by some numerical method. The following two methods have been extensively used for solving such equations:

1. Gauss-Seidel (GS) method.
2. Newton-Raphson (NR) method.

NR method was developed initially to solve nonlinear equations. GS method was developed initially to solve linear equations iteratively, and then the application of the method was extended to nonlinear equations. The method for the linear case will be explained here. It is required to solve

$$0 = \mathbf{A}\bar{\mathbf{X}} - \bar{\mathbf{b}} \quad (1.2-10)$$

where \mathbf{A} is $n \times n$ matrix.

If the unknown vector \mathbf{X} is added to both sides of the equation with \mathbf{u} representing the identity matrix then

$$\bar{\mathbf{X}} = (\mathbf{A} + \mathbf{u}) \bar{\mathbf{X}} - \bar{\mathbf{b}} \quad (1.2-11)$$

Thus the equation used for an elementary iterative scheme to solve for \mathbf{X} is

$$\bar{\mathbf{X}}^{k+1} = (\mathbf{A} + \mathbf{u}) \bar{\mathbf{X}}^k - \bar{\mathbf{b}} \quad (1.2-12)$$

where k is the iteration number.

Extending the GS method to the equation (1.2-7) we have

$$\sum_{m=1}^{NB} \bar{Y}_{im} \bar{V}_m = (\bar{S}_i / \bar{V}_i)^* \quad (1.2-13)$$

then

$$\bar{Y}_{ii} \bar{V}_i + \sum_{\substack{m=1 \\ m \neq i}}^{NB} \bar{Y}_{im} \bar{V}_m = (\bar{S}_i / \bar{V}_i)^* \quad (1.2-14)$$

$$\bar{V}_i = 1 / \bar{Y}_{ii} \left[(\bar{S}_i / \bar{V}_i)^* - \sum_{\substack{m=1 \\ m \neq i}}^{NB} \bar{Y}_{im} \bar{V}_m \right] \quad (1.2-15)$$

thus

$$\bar{V}_i^{k+1} = 1 / \bar{Y}_{ii} \left[(\bar{S}_i / \bar{V}_i^k)^* - \sum_{\substack{m=1 \\ m \neq i}}^{NB} \bar{Y}_{im} \bar{V}_m^k \right] \quad (1.2-16)$$

where k is the number of iteration.

Now applying the NR method to equation (1.2-7), we can represent this by a generalized equation $f(X) = b$. A Taylor series expansion of this equation about the true solution vector, T, is as follows

$$f(X-T) = f(T) + \delta f / \delta X (X-T) + \frac{1}{2} (X-T)^t \delta^2 f / \delta X^2 (X-T) \dots (1.2-17)$$

$$= f(T) + J(T) (X-T) + \frac{1}{2} (X-T)^t H(T) (X-T) \dots (1.2-18)$$

Because they appear in many applications, the first derivative, J is called the Jacobian and the second derivative, H, is called the Hessian. When $f(X)$ is an analytic function such as in power flow, both the Jacobian and the Hessian may easily be computed.

The Newton-Raphson iterative scheme attempts to find the true solution, T , to the generalized equation using successive evaluations of the Jacobian based on the last estimate of X :

$$g(T) = 0 = f(T) - b \quad (1.2-19)$$

This is almost equal to

$$f(X^k) - b + \delta f / \delta X (X^{k+1} - X^k) = 0 \quad (1.2-20)$$

so that $X^k \rightarrow T$ as $k \rightarrow \infty$. Usually, k is a small number of iterations.

When the two successive estimates for X are within a small tolerance limit, $(X^{k+1} - X^k)$ goes to zero and $f(X^k) - b$ is also zero at the solution. The elementary iterative scheme solves for the next value of X based on previous quantities:

$$X^{k+1} = X^k + [\delta f / \delta X (X^k)]^{-1} [b - f(X^k)] \quad (1.2-21)$$

$$= X^k + [J(X^k)]^{-1} [b - f(X^k)] \quad (1.2.22)$$

The Jacobian is updated with the latest X^k and its inverse computed for each iteration. The GS method is simple and takes less computation time per iteration but requires a large number of iterations to converge to a solution.

On the other hand, the NR method involves more calculation per iteration but only a few iterations are usually required for convergence. Also the increasing storage capability and computational speed together with sparse matrix methods to invert large matrices has resulted in the predominance of the NR method. The flowchart (fig. 1.1) illustrates the NR algorithm.

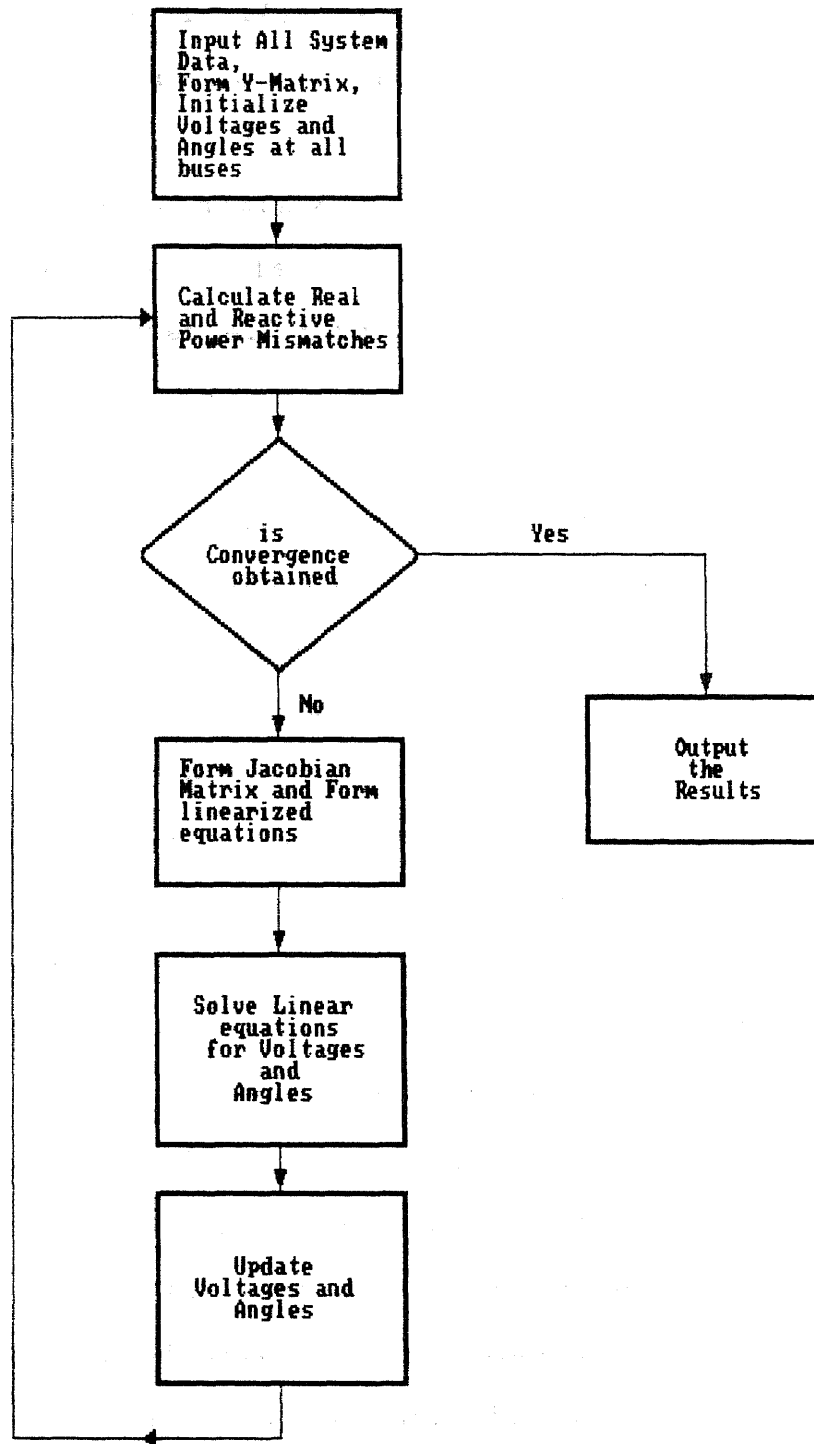


FIGURE 1.1 FLOW DIAGRAM OF THE BASIC NEWTON-RAPHSON LOAD FLOW ALGORITHM

1.4 Exploiting Sparsity

Sparsity is a significant property of the admittance bus matrix. A matrix with less than 15% non-zero elements is considered to be sparse[2]. Hence the Y bus matrix containing admittances is said to be a sparse matrix. There are plenty of benefits that can be realized from special methods of solving for sparse matrices. They are summarized below.

1. The table of factors can be obtained in a small fraction of the time required for the inverse.
2. The storage requirement is small permitting much larger systems to be solved.
3. Direct solutions can be obtained much faster.
4. Round off error is reduced.
5. Modification due to changes in matrix can be made much faster[3].

For a sparse matrix, when it is triangularized by forward substitution, the non-zero elements that are generated at every step are known as the fill-ins. The order in which the rows are processed when the matrix is triangularized affects the number of fill-in terms. Optimal ordering is to reorder the matrix to generate least number of such terms.

The optimal ordering for the buses can be done so that at each step of the reduction process minimum number of fill-in's are introduced. Following is the description of three schemes to achieve near optimal ordering.

They are listed in increasing order of programming complexity, execution time and optimality.

1. Number the rows according to the number of non-zero off-diagonal terms before elimination. In this scheme the rows with only one off-diagonal term are numbered first. Those with two terms second and then those with the most terms. This scheme does not take into account any of the subsequent effects of the elimination process. The only information needed is a list of number of non-zero terms in each row of original matrix.

2. Number the rows so that at each step of the process the next row to be operated upon is the one with the fewest non-zero terms. If more than one row meets this criterion then select any one. This scheme requires a simulation of the effects on the accumulation of non-zero terms of the elimination process. Input information is a list of rows of the column numbers of the non-zero off-diagonal terms.

3. Number the rows so that each step of the process the next row to be operated upon is the one that will introduce the fewest new non-zero terms. If more than one row meets this criterion, select any one. This involves a trial simulation of every feasible alternatives of the elimination process at each step. Input information is the same as for scheme two.

The comparative advantages of these schemes are influenced by network topology and size and the number of direct solutions wanted. Scheme 1 stands for simplicity and speed. For nodal equations of power networks scheme 2 is believed to be the best, better than scheme 1 and scheme 3[3].

1.5 Thesis Outline

As the title of the thesis suggests this work is done on conventional and harmonic load flow analysis. Chapter 1 is an introductory chapter. It discusses the overview of the power systems, the LF equations, the iterative schemes for solving the LF equations and it introduces the concept of exploiting sparsity while dealing with the LF equations.

The focus of chapter 2 is on the conventional load flow algorithm. It discusses the conceptual model used for designing the software to carry out the conventional LF. It also discusses the factors that were considered while designing the software for conventional LF.

Chapter 3 gives the listings and flowcharts of conventional LF analysis programs. Comments have been used liberally to clarify the program listing.

Chapter 4 discusses the harmonic LF analysis. The focus of this chapter is on established, new and future possible sources of power harmonics and it also summarizes the topic of harmonic studies and discusses reformulation of NR method to allow for harmonics.

It discusses the use of software HARMFLO in carrying out the harmonic LF analysis. Lastly in this chapter the listing and flow chart of the program that was written to modify circuit elements in accordance with the order of harmonics has been included.

Chapter 5 is devoted entirely to the results and observations noted down during the execution of conventional and harmonic LF analysis.

Appendices A and B are included to simplify the notations and the structures of the arrays generated during exploiting sparsity while solving for LF analysis. They denote a packed storage scheme. Appendix C gives the nomenclature used for items in chapter 4 and Appendix D gives an example to form TG and H matrices while dealing with harmonic load flow equations.

Finally the selected bibliography provides a reference to a wide range of literature from introductory material to specific technical papers describing the conventional and harmonic LF analysis.

CHAPTER TWO

CONVENTIONAL LOAD FLOW ALGORITHM

2.1 Algorithm design

The conventional LF programs have been designed so that the following options are available.

1. With Optimal ordering
2. Without optimal ordering
3. Testing systems under different loading conditions

The brief explanation of these options follows.

1. With Optimal Ordering

This thesis uses the scheme 2 discussed in section 1.4 for optimal ordering. The background material for this has been collected from the work introduced by K. Zollenkopf[4]. The original triangular decomposition method was modified by Tinney [5-7] where the reduction of coefficient matrix is the same as in triangularization. The direct solution vector however is found by successive factor-matrix-by-vector multiplication. The method given by Tinney was also derived by K. Zollenkopf and was introduced by the term bi-factorisation.

The method described in this thesis to solve the conventional load flow analysis should be used for sparse coefficient matrices that have non-zero diagonal terms and are asymmetric in element value but with a symmetric sparsity structure. Furthermore it is assumed that the matrix is diagonally dominant.

This program has been listed under the name JEET.

2. Without Optimal Ordering

To record the advantages of optimally ordering the columns of the matrix elements, we have to first find what would happen if we do not order the columns optimally. Hence this is an effort to avoid optimal ordering. This is done by making changes to the subroutines of JEET. They have been listed in chapter three in the program NOOPTIMAL.

Significant variations are noticed in execution time and memory with increasing the number of buses. The tests were carried on 14 bus and 118 bus systems.

3. Testing systems under different loading conditions

This program exploits sparsity and optimal ordering but at the same time could be used to study the same system under different loading conditions.

By this we mean that the system could be analyzed by using randomly generated values for voltages and phase angles at different buses. By changing the value of the variable from which these values are randomly generated, values for voltages and phase angles could be changed though system specified values for V and θ are kept unchanged. On reading V and θ the program calculates P and Q which in turn are interpreted as specified values and then the execution of the program continues like JEET. This program utilises VAX/VMS system function RAND. The program is listed under the name DIFFLOAD.

2.2 Conceptual model for conventional load flow

Now that the deciding factors for the algorithm have been discussed, the next entity to be reviewed is a scheme and programming technique to solve the load flow equations. Since our jacobian matrix is sparse we can achieve the saving of storage and computation time if we implement in our programming effort, a scheme which stores and processes only non-zero terms. Since we are interested in maintaining sparsity as far as possible the sparsity directed pivotal selection helps in our objective, which is nothing but optimal ordering.

This in turn reduces the fill-in terms to a great extent but at the same time requires additional programming and computation time which could take away a great deal of the advantages obtained by exploiting sparsity. Hence in our programming we use a strategy where a near optimal ordering is achieved by little additional computation.

The principle of this strategy is to select at each step of the reduction process that column as pivot which contains the fewest number of non-zero terms. If more than one column meets this criterion, any one is selected. This scheme requires a current bookkeeping of the number of non-zero terms in each column or row.

For exploiting sparsity we need a packed storage scheme where along with the elements of the matrix, the indexing and addressing information of the elements also has to be stored.

This would be comparatively simple if the number of elements did not vary in the course of computation. But since the number of elements do change due to fill-in's and due to the reduction process a flexible storage mode is necessary.

2.2.1 Implemented storage scheme

The non-zero matrix elements are stored columnwise in array CE and rowwise in array RE. RE and CE do not include the diagonal elements which are stored separately in DE. Since the matrix has a symmetric structure of non-zero elements, the row and column indices of the elements in RE and CE are stored in array ITAG. The accompanying table LNXT contains the location of the next element in CE in ascending order. The entry 0 in LNXT indicates the last term of a column. The starting row positions of the individual columns in CE are stored in table LCOL and the table NOZE contains the number of non-zero elements in each column. The vacant position of arrays RE and CE should be set equal to 0 as also the last position in array LNXT. The other vacant positions in array LNXT must be numbered consecutively.

Other than this the dimension of the coefficient matrix is stored in DFM and the first vacant location in tables CE, RE, ITAG and LNXT is stored in LF. In case a row or column has no off diagonal terms the respective position in table LCOL is to be set to zero. The detailed example for this storage scheme is given in Appendix A and the parameter explanation has been given in Appendix B.

2.2.2 Programming techniques

The programming part is as important as the method itself. Optimal ordering can be determined during the course of computation, but it is more efficient to determine it by simulating the reduction process beforehand. Thus the program can be split up into three parts[4].

- A. Simulation and ordering
- B. Reduction
- C. Direct solution

A. Simulation and ordering

The optimal ordering requires an additional array NSEQ. This array initially contains integers 1 to n where n is the number of columns or the rows. At the end of simulation process array NSEQ contains the pivotal sequence as it results from the applied strategy. After processing the simulation and ordering subprogram, the tables LCOL, NOZE, NSEQ, ITAG and LNXT no longer contain the information on the structure of the original coefficient matrix, but contain instead the structure of the factor matrices.

B. Reduction

The reduction subprogram operates upon the storage image resulting from the simulation and ordering subprogram. The actual reduction of the coefficient matrix is guided by the pivotal sequence contained in the table NSEQ. At each stage of the reduction process only those terms of the reduced residual matrix with subscripts corresponding to row indices of the pivotal column have to be recalculated. Thus the corresponding columns are compared term by term with the pivotal column in much same way as the simulation subprogram. Every derived term of the factor matrices is left in the position of the corresponding term of the coefficient matrix.

C. Direct solution

The right hand side vector of the equation is stepwise transformed to the solution vector by successive factor matrix by vector multiplications. After having processed the direct solution subroutine the right hand side vector contains the solution. The total number of arithmetical operations (multiplications and additions) for computing the direct solution in the bi-factorisation method is the same as in the triangular decomposition method.

An important advantage of the bi-factorisation method, however, is realised in programming, because the symmetric structure of the coefficient matrix can be completely exploited. Also in the bi-factorisation method the factor matrices are formed from the rows of the upper triangle and from the columns of the lower triangle which are identical in pattern of non-zero elements. Hence the bi-factorisation method requires only half as much indexing information as the triangular decomposition method.

CHAPTER THREE

CONVENTIONAL LOAD FLOW PROGRAM

This chapter explains the step by step listing of the program to solve for conventional LF analysis. The program is listed under the name JEET. It is written in FORTRAN 77 and runs on VAX 11/750 machine. The program utilizes the techniques of sparsity and optimal ordering discussed in the earlier part of the thesis. Perfect optimal ordering would involve a lot of computation which could get complicated and thus exploiting sparsity would loose its significance. Hence a near perfect optimal ordering is achieved in the program. JEET has been initialized with data from a 14 bus system. It could be used to test any system by changing the initial data statements. It has a main routine and nine subroutines. The length of the arrays LCOL, NOZE, NSEQ, DE AND CON is governed by the number of unknowns and is declared in DFM while the length of the arrays ITAG, LNXT, RE, CE declared by SWN cannot be determined in advance. Thus it is recommended to have SWN sufficiently large. The subroutines are used to read the input data, form the jacobian matrix, simulate reduction, factorize the matrix and obtain the final solution. In conclusion the program has been found to give satisfactory results on testing.

3.1 Exploiting sparsity and optimal ordering

(Program JEET)

3.1.1 MAIN ROUTINE : Initializing and reading network data.

A. Description:

The control of the program lies with the main routine. It is responsible for calling subroutines to achieve the necessary results. The flowchart is shown in figure 3.1.

The function of the main routine is

1. Initialize the data statements.
2. Open the input data files.
3. Read the line data.
4. Calculate active and reactive power from line data by calling subroutine PQCAL.
5. Calculate the array of power mismatches CON by calling subroutine DELTA.
6. Test for convergence or divergence of mismatches by calling subroutine TESTPO.

7. Calculate the Jacobian matrix by calling subroutine JACOB.
8. Form the special arrays to store non-zero elements by calling subroutine DATAGEN.
9. Simulate the reduction process by calling subroutine RADO.
10. Carry out the actual reduction by calling subroutine FACTO.
11. Adjust for slack bus in array CON by calling subroutine SLAD.
12. Find the final solution vector by calling subroutine UTION.
13. Update the unknowns V and θ .
14. Return to step 4.

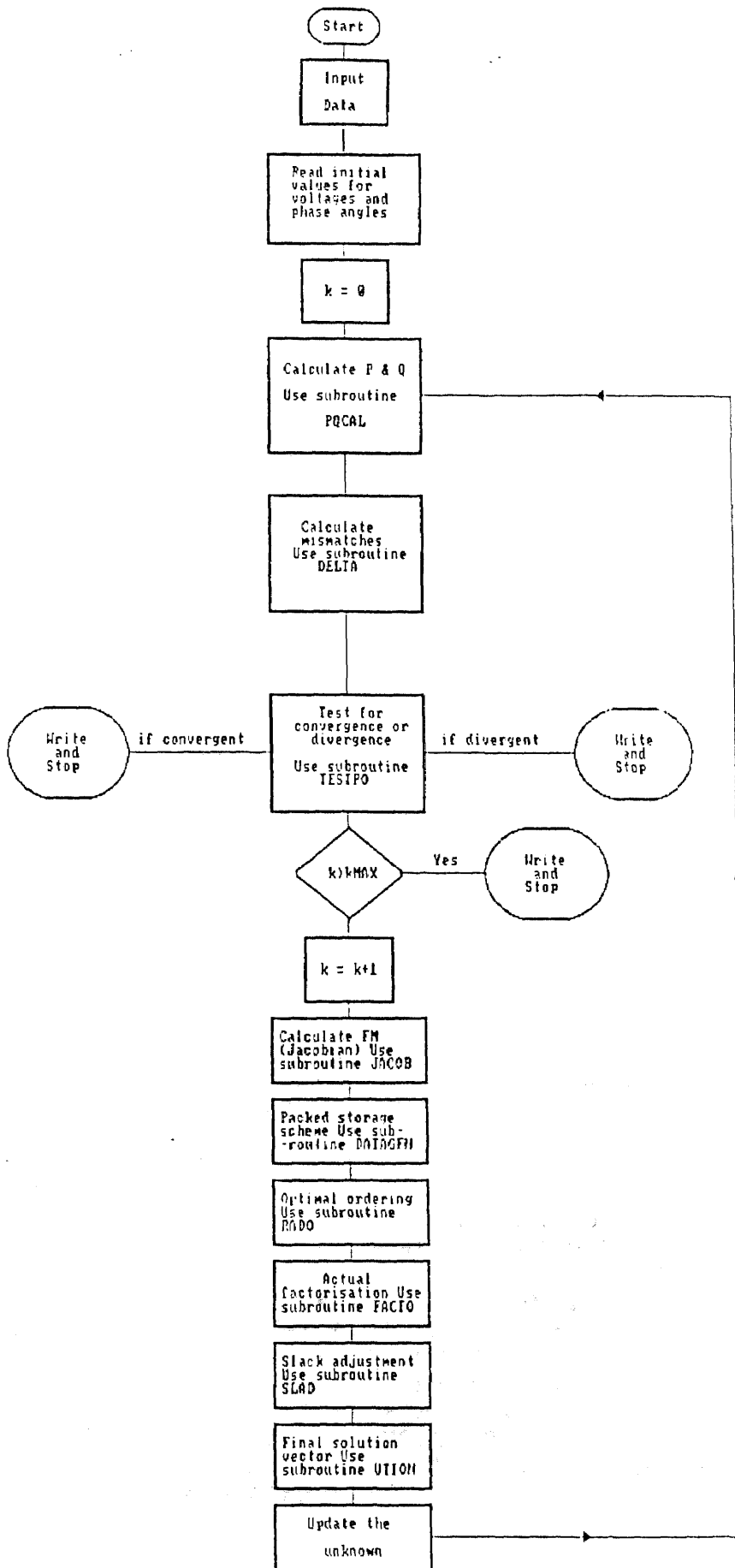


FIGURE 3.1 MAIN ROUTINE

B. List of the Main Routine

```

C *****
C *                PROGRAM JEET                *
C *                FOR                          *
C *                CONVENTIONAL LOAD FLOW ANALYSIS *
C *****

```

```

C THIS PROGRAM IS USED TO SOLVE A FOURTEEN BUS
C POWER SYSTEM EXPLOITING SPARSITY AND USING
C OPTIMAL ORDERING.
C     TERMS USED ARE
C NB          NUMBER OF BUSES
C NPN         TWICE THE NUMBER OF BUSES
C NL          NUMBER OF LOAD NOADS
C N LINES     NUMBER OF LINES
C DFM         DIMENSION OF JACOBIAN
C FM          JACOBIAN MATRIX
C SWN         DIMENSION OF ARRAYS IN SUBROUTINE
C             DATAGEN TO ACCOMODATE FOR THE FILL IN
C             TERMS.
C PSP        SPECIFIED REAL POWER
C QSP        SPECIFIED IMAGINARY POWER
C SP         SPECIFIED COMPLEX POWER
C VSP        SPECIFIED VOLTAGE AT GENERATOR BUSES
C V          ASSUMED INITIAL VOLTAGES AT LOAD BUSES
C D          ASSUMED PHASE ANGLES AT LOAD AND
C             GENERATOR BUSES
C CON        ARRAY CONTAINING POWER MISMATCHES;
C             WILL CONTAIN SOLUTION VECTOR IN THE
C             END.
C G          LINE CONDUCTANCES
C B          LINE SUSEPTANCE; TOGETHER WITH G FORMS
C             LINE ADMITANCES
C CRIT       CRITERIA FOR CONVERGENCE OR DIVERGENCE
C KMAX       MAXIMUM NUMBER OF ITERATIONS.
C THIS PROGRAM READS THE INPUT FROM FILE NAMED
C Z14.DAT. Z14.DAT IS THE DATA FILE CONTAINING LINE
C ADMITANCES, SPECIFIED POWER AT LOAD BUSES AND
C SPECIFIED VOLTAGES AT GENERATOR BUSES IN THAT
C ORDER. THE OUTPUT OF THIS PROGRAM IS DIRECTED TO
C APPEAR ON THE TERMINAL SCREEN. THE OUTPUT
C CONSISTS OF THE SOLUTION VECTOR IN TERMS OF POWER
C MISMATCHES WHICH IS THE ARRAY CON. THOUGH THIS
C PROGRAM IS INITIALISED FOR A 14 BUS SYSTEM, IT
C COULD BE USED FOR ANY SYSTEM BY CHANGING THE
C DIMENSIONS IN VARIABLE DECLARATIONS AND THE
C INITIAL DATA STATEMENTS.

```

```

INTEGER DFM,SWN
COMMON DFM,SWN
COMPLEX SP
REAL PSP(14),QSP(14),V(14),D(14), CON(28),
1G(14,14),B(14,14),VSP(14),FM(22,22),P(14),Q(14),
2DE(22),RE(125),CE(125)
REAL JK11(14,14),JK12(14,14)
REAL JK21(14,14),JK22(14,14)
INTEGER FJM,MFJM,LF,NR,ITEST
INTEGER LCOL(22),NOZE(22),NSEQ(22)
INTEGER ITAG(125),LNXT(125),DOZE(22)
DATA NB,NLINES,NL,K1,K2,NPN/14,20,9,10,11,28 /
DATA DFM,SWN/22,125/
DATA CRIT,KMAX/0.0005,40/
DATA NR,FJM,MFJM/1,0,0/
DATA V(1),D(1)/1.06,0.0/
COMMON LF
OPEN(UNIT=15,FILE='JEET.IN',STATUS='OLD')
C*****
C K1 IS THE NO. OF LOAD NODES + 1
C K2 IS THE NO. OF LOAD NODES + 2, NEEDED TO READ
C VSP
C NPN = 2*NB
C DFM = NB + NL -1 AND IS THE DIMENSION OF FM
C*****
DO 5 I=1,NB
DO 5 J=1,NB
B(I,J)=0.0
5 G(I,J)=0.0
NBL=NB+NLINES
100 FORMAT(2I3,2F10.5)

C*****READING THE LINE ADMITANCES*****

DO 10 L=1,NBL
READ(15,100)I,J,G(I,J),B(I,J)
G(J,I)=G(I,J)
B(J,I)=B(I,J)
10 CONTINUE

C*****READING THE SPECIFIED POWER*****

DO 15 I=1,NB
READ(15,*)J,SP
PSP(I)=-REAL(SP)
QSP(I)=-AIMAG(SP)
15 CONTINUE

```

```

C*****
C   NOTE THAT PSP(1) AND QSP(1) ARE BOTH ZERO (SLACK
C   NODE
C   ALSO Q FOR THE GENERATOR NODES IS ZERO
C*****

      IF(K1.EQ.NB)GO TO 20

C***READING THE SPECIFIED VOLTAGES FOR GEN. BUSES.***

      READ(15,*) (VSP(I),I=K2,NB)
20    CONTINUE
22    K=0

C*****INITIAL VALUES*****

      DO 25 I = 2,NB
      V(I) = 1.0
25    D(I) = 0.0
      IF(K1.EQ.NB)GO TO 40
      DO 30 I=K2,NB
      V(I) = VSP(I)
30    CONTINUE
40    CONTINUE

C*****
C   PQCAL CALCULATES P AND Q AT THE BUSES FROM V, D
C   AND G, B.
C*****

      CALL PQCAL(P,Q,V,D,B,G,NB)

C*****
C   DELTA FINDS THE POWER MISMATCHES AND FORMS THE
C   ARRAY CON.
C*****

      CALL DELTA(PSP,QSP,P,Q,CON,NB,NPN)

C*****
C   TESTPO CHECKS IF THE SOLUTION HAS CONVERGED OR
C   DIVERGED
C   CONVERGENCE CRITERIA IS DECIDED BY CRIT.
C*****

      CALL TESTPO(NB,NPN,K1,ITEST,CRIT,CON,PSP,
                QSP,K,FJM,MFJM,NR)
      IF(ITEST.EQ.1.OR.ITEST.EQ.2) GO TO 60
      IF(K.GE.KMAX) THEN
      WRITE(6,666)K

```

```

666  FORMAT(10X,'ELEMENTS OF CON FAILED TO CONVERGE OR
1DIVERGE IN',I3,3X,'ITERATIONS')
      WRITE(6,292)(CON(I),I=1,NB)
      WRITE(6,292)(CON(I+NB),I=1,NB)
292  FORMAT(/1X,10(F9.4,1X))
      GO TO 444
      ENDIF

```

```

C*****INCREMENT THE ITERATION COUNT*****

```

```

      K=K+1

```

```

C*****
C      JACOB CALCULATES THE JACOBIAN. THE OUTPUT OF
C      JACOB IS FM.
C*****

```

```

      CALL JACOB(JK11,JK12,JK21,JK22,FM,
                P,Q,V,D,NB,K1,B,G)

```

```

C*****
C      DATAGEN READS FM AND FORMS ARRAYS TO EXPLOIT THE
C      SPARSITY OF FM. THIS ENSURES OPTIMUM COMPUTATION
C      TIME AND A PACKED MATRIX STORAGE SCHEME.
C*****

```

```

      CALL DATAGEN(FM,LCOL,NOZE,DE,ITAG,
                  LNXT,CE,RE,NSEQ)

```

```

C*****
C      RADO ENSURES OPTIMAL ORDERING. THIS SIMULATES THE
C      REDUCTION PROCESS STEP BY STEP AND PROVIDES
C      POSITIONS TO FILL IN TERMS.
C*****

```

```

      CALL RADO(LCOL,NOZE,DE,ITAG,LNXT,CE,RE,NSEQ)

```

```

C*****
C      FACTO ACTS ON THE IMAGE PRODUCED BY RADO AND
C      CARRIES ON THE ACTUAL REDUCTION.
C*****

```

```

      CALL FACTO(LCOL,NOZE,DE,ITAG,LNXT,CE,RE,NSEQ)

```

```

C*****SLAD DOES THE SLACK BUS ADJUSTMENT ON CON*****

```

```

      CALL SLAD(CON,NPN,NB)

```

```
C*****
C   UTION IS THE ACTUAL SOLUTION SUBROUTINE. IT
C   OPERATES ON CON TO GIVE FINAL SOLUTION VECTOR.
C*****
```

```
      CALL UTION(LCOL,NOZE,DE,ITAG,LNXT,
                CE,RE,NSEQ,CON,NPN)
```

```
      DO 50 I=2,NB
50     D(I) = D(I) + CON(I - 1)
      DO 55 I = 2,K1
      V(I) = V(I) *(1.0 + CON(I + NB - 2))
55     CONTINUE
      GO TO 40
60     WRITE(6,292) (CON(I),I=1,NB)
      WRITE(6,292) (CON(I+NB),I=1,NB)
444    STOP
      END
```

```
C*****
```

3.1.2 SUBROUTINE PQCAL : Calculation of P and Q.

A. Description:

Subroutine PQCAL is used to calculate injected active and reactive power at the buses from complex voltages and line admittances. The flowchart is shown in figure 3.2.

The steps involved are

1. Initialize P and Q arrays to zero.
2. Find the sin and cos values of the phase angle differences between the involved buses.
3. Using the equations (1.2-8) and (1.2-9) calculate P and Q.
4. Return to the main routine.

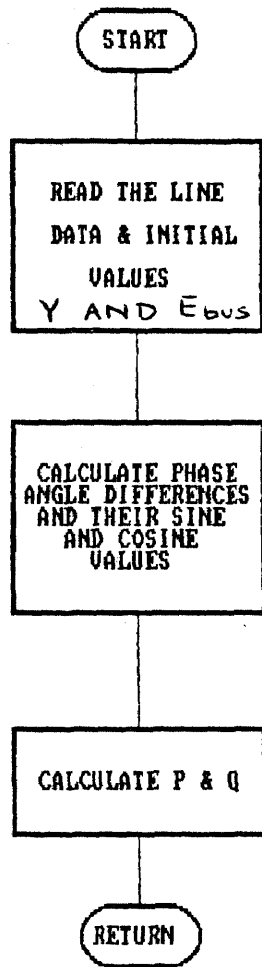


FIGURE 3.2 SUBROUTINE PQCAL

B. List of the subroutine PQCAL

```

*****
*                               SUBROUTINE PQCAL                               *
*****

      SUBROUTINE PQCAL(P,Q,V,D,B,G,NB)

*****
C THE OUTPUT IS P(CAL) AND Q(CAL) AT ALL THE NODES
*****

      REAL P(NB),Q(NB),V(NB),D(NB),G(NB,NB),B(NB,NB)
      DO 30 I=1,NB
      P(I)=0.0
      Q(I)=0.0
      DO 25 J=1,NB
      IF(G(I,J).EQ.0.0.AND.B(I,J).EQ.0.0)GO TO 25
      DIJ=D(I)-D(J)
      SINDIJ = SIN(DIJ)
      COSDIJ = COS(DIJ)

C*****CALCULATING P AND Q*****
      P(I) = P(I) + V(J)*(G(I,J)*COSDIJ -
                               B(I,J)*SINDIJ)
      Q(I) = Q(I) + V(J)*(B(I,J)*COSDIJ +
                               G(I,J)*SINDIJ)
25      CONTINUE
      P(I) =P(I)*V(I)
      Q(I) = Q(I)*V(I)
30      CONTINUE
      RETURN
      END

```

3.1.3 SUBROUTINE DELTA : Calculating the mismatches.

A. Description:

Subroutine DELTA is used to calculate the power mismatches at every node. The flowchart is shown in figure 3.3.

The steps involved are

1. Read in the specified power and the power calculated by subroutine PQCAL.
2. Find the mismatch of power.
3. Form the array CON of mismatches, this involves combining ΔP and ΔQ in one array.
4. Return to the main routine.

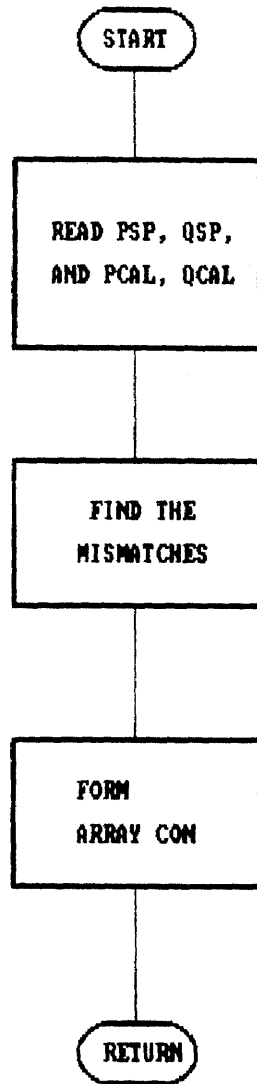


FIGURE 3.3 SUBROUTINE DELTA

B. List of the subroutine DELTA

```
*****
*                SUBROUTINE DELTA                *
*****

      SUBROUTINE DELTA(PSP,QSP,P,Q,CON,NB,NPN)
      REAL PSP(NB),QSP(NB),P(NB),Q(NB),CON(NPN)

C*****CON = SPECIFIED - CALCULATED FOR ALL THE NODES.
      DO 10 I=1,NB
      CON(I) = PSP(I) - P(I)
      CON(I + NB) = QSP(I) - Q(I)
10    CONTINUE
      RETURN
      END
```

3.1.4 SUBROUTINE TESTPO : Testing for convergence or divergence.

A. Description:

Subroutine TESTPO checks if the solution has converged or diverged. The convergence criteria is decided by the variable CRIT in the data statement. This value can be changed to any other value for specific usage. Convergence criterion is that all the mismatches are either less than CRIT or less than 1 percent of the specified P or Q. If any of the mismatches is greater than 10 the method is considered to be divergent. The output of TESTPO is variable ITEST which is set to different values for different conditions namely convergence, divergence or neither of the two. The flow chart is as shown in figure 3.4.

The steps involved are

1. Set ITEST to 0 (neither convergent or divergent).
2. Check for convergence according to the convergence criterion discussed above.
3. If convergence is obtained set ITEST to 1 and return to main routine.

4. If convergence is not obtained check if the method has diverged.
5. If divergence is confirmed set ITEST to 2 and return to main routine.
6. If neither convergence nor divergence is obtained value of ITEST remains zero.
7. Return to the main routine.

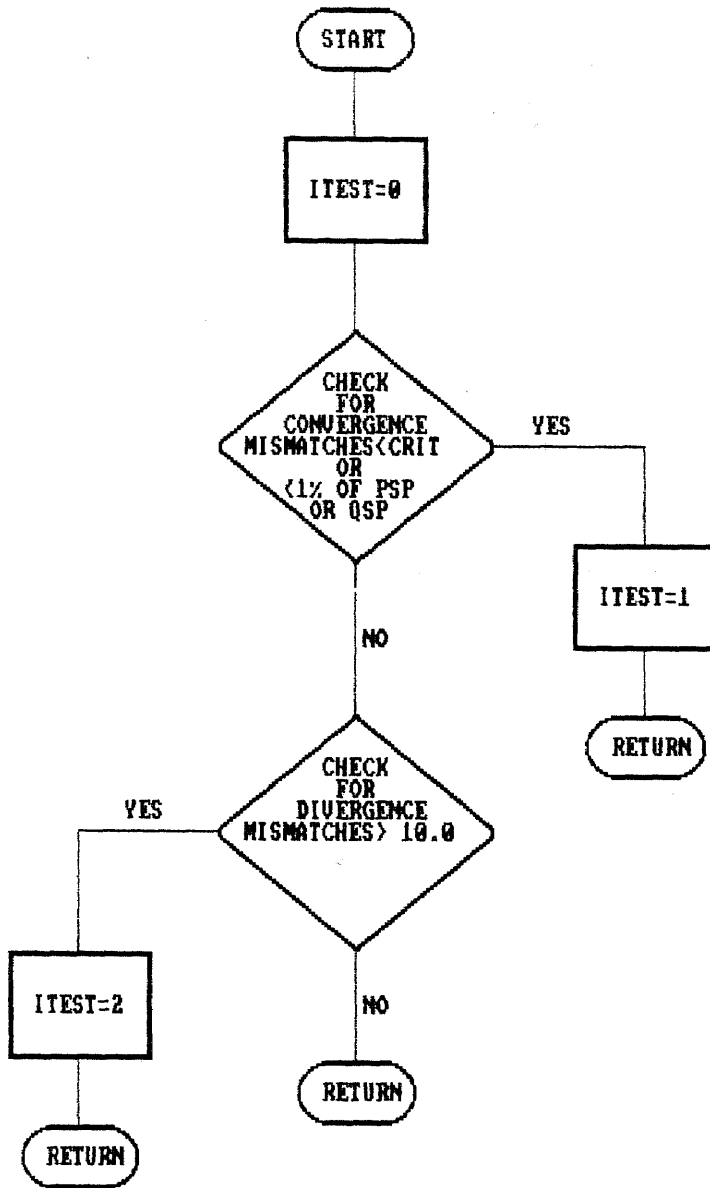


FIGURE 3.4 SUBROUTINE TESTPO

B. List of the subroutine TESTPO

```

*****
*                               SUBROUTINE TESTPO                               *
*****

      SUBROUTINE TESTPO(NB,NPN,K1,ITEST,CRIT,CON,PSP,
                        QSP,K,FJM,MFJM,NR)
      REAL CON(NPN),PSP(NB),QSP(NB)

C*****
C   IF ANY OF THE MISMATCHES IS > 10.0 THE METHOD IS
C   CONSIDERED DIVERGENT AND ITEST WILL BE = 2
C   CONVERGENCE CRITERION IS THAT ALL THE MISMATCHES
C   ARE EITHER LESS THAN CRIT OR LESS THAN 1 PERCENT
C   OF THE SPECIFIED P OR Q
C   ITEST = 1 IF CONVERGED
C   IF NO CONVERGENCE THEN ITEST = 0
C*****

      INTEGER FJM
200  FORMAT(1X,'DELTA P AT NODE',I2,3H IS,F10.5,
           'DIVERGENT SOLUTION AFTER',I2,'ITERATIONS'/)
      ITEST = 0

C*****
C   TESTING CONVERGENCE
*****

      DO 5 I=2,NB
      IF(PSP(I).LT.100.0*CRIT)GO TO 3
      APP=100.0*ABS(CON(I)/PSP(I))
      IF(APP.LT.1.0)GO TO 5
      GO TO 12
3     APA = ABS(CON(I))/CRIT
      IF(APA.LT.1.0)GO TO 5
      GO TO 12
5     CONTINUE
      DO 10 I=2,K1
      IPNB = I+NB
      IF(QSP(I).LT.100.0*CRIT)GO TO 7
      AQP = 100.0*ABS(CON(IPNB)/QSP(I))
      IF(AQP.LT.1.0)GO TO 10
      GO TO 12
7     AQA = ABS(CON(IPNB))/CRIT
      IF(AQA.LT.1.0)GO TO 10
      GO TO 12
10    CONTINUE
      ITEST=1

```

```

WRITE(6,220)K,FJM,MFJM,NR
220  FORMAT(10X,'CONVERGED IN',I3,' ITERATION BY
      METHOD FJM =',I2,' MFJM =',I2,' NR =',I2//)
      GO TO 50

*****
C    TESTING DIVERGENCE
*****

12   DO 15 I=2,NB
      AP=ABS(CON(I))
      IF(AP.LT.10.0)GO TO 15
      WRITE(6,200)I,CON(I),K
      ITEST=2
      GO TO 50
15   CONTINUE
      DO 20 I=2,K1
      IPNB=I+NB
      AQ=ABS(CON(IPNB))
      IF(AQ.LT.10.0)GO TO 20
      WRITE(6,210)I,CON(IPNB),K,FJM,MFJM
210  FORMAT(1X,'DELTA Q NODE',I2,' IS',F10.5,' DIVERGENT
1'AFTER',I2,' ITERATION ; BY METHOD ,FJM=',I2,' AND
2MFJM=',I2//)
      ITEST=2
      GO TO 50
20   CONTINUE
50   RETURN
      END

```

3.1.5 SUBROUTINE JACOB : Calculation of the Jacobian matrix.

A. Description:

Subroutine JACOB calculates the jacobian and the output is stored in the two dimensional array FM. The calculated values of P and Q are needed to calculate the diagonal elements of sub-jacobians. The flow chart is as shown in figure 3.5. Polar coordinates are used.

The steps involved are

1. Initialize the sub-jacobian elements.
2. Find the sin and cos values of the phase angle differences between the involved buses.
3. Calculate the sub-jacobian elements using equations (4.2-16) through (4.2-19).
4. Calculate the diagonal sub-jacobian elements using equations (4.2-20) through (4.2-23).
5. Form FM, the jacobian matrix, by deleting the rows and columns corresponding to the slack bus.
6. Return to the main routine.

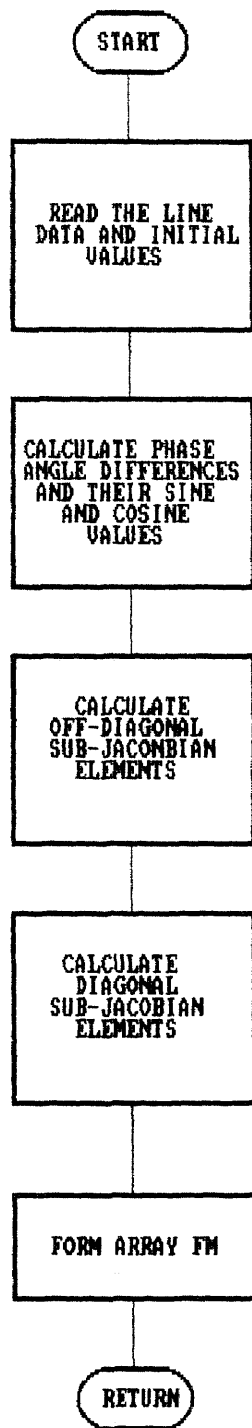


FIGURE 3.5 SUBROUTINE JACOB

B. List of the subroutine JACOB

```

*****
*                               SUBROUTNE JACOB                               *
*****

      SUBROUTINE JACOB(JK11,JK12,JK21,JK22,FM,
                     P,Q,V,D,NB,K1,B,G)

C*****
C   PCAL AND QCAL ARE NEEDED TO CALCULATE THE
C   DIAGONAL ELEMENTS OF THE SUBJACOBIANS. THE OUTPUT
C   IS FM.
C*****

      INTEGER DFM,SWN
      COMMON DFM,SWN
      REAL JK11(NB,NB),JK12(NB,NB)
      REAL JK21(NB,NB),JK22(NB,NB),
      1FM(DFM,DFM),P(NB),Q(NB),V(NB),
      2D(NB),B(NB,NB),G(NB,NB)

C*****INITIALIZATION*****
      DO 10 I=1,NB
      DO 10 J=1,NB
      JK11(I,J)=0.0
      JK12(I,J)=0.0
      JK21(I,J)=0.0
      JK22(I,J)=0.0
10    CONTINUE
      DO 20 I=1,NB
      DO 15 J=1,NB
      IF(I.EQ.J)GO TO 15
      IF(G(I,J).EQ.0.0.AND.B(I,J).EQ.0.0)GO TO 15
      DIJ = D(I) - D(J)
      SINDIJ=SIN(DIJ)
      COSDIJ=COS(DIJ)

C*****CALCULATION OF OFF-DIAGONAL ELEMENTS*****
      JK11(I,J)=V(I)*V(J)*(G(I,J)*SINDIJ +
                          B(I,J)*COSDIJ)
      JK12(I,J)=V(I)*V(J)*(G(I,J)*COSDIJ -
                          B(I,J)*SINDIJ)

      JK21(I,J) = -JK12(I,J)
      JK22(I,J) = JK11(I,J)
15    CONTINUE
20    CONTINUE

C*****CALCULATION OF DIAGONAL ELEMENTS OF JACOBIAN*****
      DO 25 I=1,NB

```

```
JK11(I,I) = V(I)*V(I)*B(I,I) - Q(I)
JK12(I,I) = V(I)*V(I)*G(I,I) + P(I)
JK21(I,I) = -V(I)*V(I)*G(I,I) + P(I)
JK22(I,I) = V(I)*V(I)*B(I,I) + Q(I)
25  CONTINUE
    DO 30 I = 2,NB
    DO 30 J = 2,NB
30  FM(I-1,J-1) = JK11(I,J)
    DO 35 I = 2,NB
    DO 35 J=2,K1
35  FM(I-1,J+NB-2) = JK12(I,J)
    DO 40 I=2,K1
    DO 40 J=2,NB
40  FM(I+NB-2,J-1) = JK21(I,J)
    DO 45 I=2,K1
    DO 45 J=2,K1
45  FM(I+NB-2,J+NB-2) = JK22(I,J)
    RETURN
    END
```

3.1.6 SUBROUTINE DATAGEN : Formation of packed density scheme.

A. Description:

This subroutine reads the FM matrix and stores the non-zero elements in arrays to enable a packed storage scheme. This does the book-keeping for columnwise (CE) and rowwise (RE) stored matrix terms and also the diagonal elements (DE). It also keeps track of the starting positions of columns (LCOL), number of non-zero elements in the column (NOZE), the location of the next non-zero term (LNXT) and the sequence of pivotal indices (NSEQ). The next vacant position in arrays CE, RE, and ITAG is kept in variable LF. The dimension of arrays LCOL, NOZE, NSEQ is in DFM while the dimensions of ITAG, LNXT, RE, CE cannot be predetermined due to creation of the fill-in terms. Hence this dimension has to be sufficiently large. Also the vacant positions of arrays CE and RE and the last position of the array LNXT must be set to zero. The other vacant positions of LNXT must be numbered consecutively. Appendix A is included for clarifying any doubts about these arrays. The flow chart is shown in figure 3.6.

The steps involved are

1. Initialize the arrays LCOL, NOZE, DE and NSEQ.
2. Create elements of LCOL. These are given by the start of columns of FM.
3. Create elements of DE. These are diagonal elements of FM.
4. Form ITAG which contains the row indices of the elements stored in CE as well as column indices of the elements stored in RE. Also form CE, RE and locate the next vacant position LF.
5. Form the arrays LNXT and NSEQ.
6. Return to the main routine.

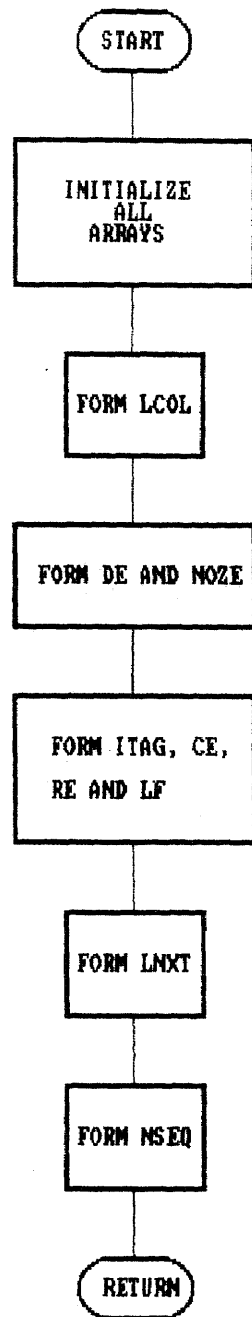


FIGURE 3.6 SUBROUTINE DATAGEN

B. List of the subroutine DATAGEN

```

*****
*           SUBROUTINE DATAGEN           *
*****
      SUBROUTINE DATAGEN(FM,LCOL,NOZE,DE,
                        ITAG, LNXT, CE, RE, NSEQ)

C*****
C      THIS IS THE SUBROUTINE WHICH GENERATES DATA
C      REQUIRED FOR OTHER SUBROUTINES.
C      DESCRIPTION OF PARAMETERS
C      FM      ORIGINAL SPARSE COEFICIENT MATRIX
C      LCOL    STARTING POSITION OF COLUMNS
C      NOZE    NUMBER OF NON ZERO TERMS IN THE COLUMN
C      DE      DIAGONAL TERMS
C      ITAG    ROW INDEX OF ELEMENTS STORED IN CE
C      LNXT    LOCATION OF NEXT NON ZERO TERM
C      CE      COLUMNWISE STORED MATRIX TERMS
C      RE      ROWWISE STORED MATRIX TERMS
C      NSEQ    SEQUENCE OF PIVOTAL INDICES
C*****
      INTEGER DFM, SWN
      COMMON DFM, SWN
      REAL FM(DFM,DFM), DE(DFM), RE(SWN), CE(SWN)
      INTEGER LCOL(DFM), LF, NOZE(DFM)
      INTEGER ITAG(SWN), LNXT(SWN), NSEQ(DFM)
      COMMON LF, DOZE(22)
      DO 2 I=1,DFM
      LCOL(I)=0
      NOZE(I)=0
      DE(I)=0
      NSEQ(I)=0
2      CONTINUE

C*****
C      THIS SEGMENT GENERATES ELEMENTS OF LCOL
C*****
      K=0
      DONE=0
      DO 20 J=1,DFM
      DO 10 I=1,DFM
      IF((J.EQ.1).AND.(I.EQ.1)) THEN
      LCOL(1)=1
      DONE=1
      ENDIF
      IF(I.EQ.J)GO TO 500
      IF(FM(I,J).EQ.0) THEN
      GO TO 500

```

```

ELSE
K=K+1
IF(DONE.EQ.0) THEN
LCOL(J)=K
DONE=1
ENDIF
ENDIF
500 IF(I.EQ.DFM) THEN
DONE=0
ENDIF
10 CONTINUE
20 CONTINUE

C*****
C THIS SEGMENT GENERATES DE AND NOZE
C*****
TOT=0
DO 30 J=1,DFM
DO 40 I=1,DFM
IF(FM(I,J).NE.0)TOT=TOT+1
IF(I.EQ.J)DE(J)=FM(I,J)
40 CONTINUE
NOZE(J)=TOT
TOT=0
30 CONTINUE

C*****
C THIS SEGMENT GENERATES ITAG,CE,RE AND GIVES THE
C NEXT VACANT POSITION LF
C*****
DO 270 I=1,SWN
ITAG(I)=0
CE(I)=0
RE(I)=0
270 CONTINUE
K=0
DO 50 J=1,DFM
DO 2060 I=1,DFM
IF(I.EQ.J)GO TO 2060
IF(FM(I,J).NE.0) THEN
K=K+1
ITAG(K)=I
CE(K)=FM(I,J)
RE(K)=FM(J,I)
ENDIF
2060 CONTINUE
50 CONTINUE
K=K+1
LF=K

```

```

C*****
C   THIS SEGMENT GENERATES THE ELEMENTS OF LNXT
C*****
      DO 665 I=1,SWN
      LNXT(I)=0
665   CONTINUE
      TOT=0
      DO 110 J=1,DFM
      DO 120 I=1,DFM
      IF ( (FM(I,J) .NE. 0) .AND. (I.NE.J) ) TOT=TOT+1
120   CONTINUE
      DOZE(J)=TOT
      TOT=0
110   CONTINUE
      M=0
      K=1
      L=1
      DO 850 J=1,DFM
      DO 860 I=1,DFM
      IF (J.EQ.I) THEN
      GO TO 860
      ENDIF
      IF (FM(I,J) .NE. 0) THEN
      IF ( (M+1) .EQ. DOZE(J) ) THEN
      LNXT(L)=0
      L=L+1
      M=M+1
      K=K+1
      ELSE
      K=K+1
      LNXT(L)=K
      L=L+1
      M=M+1
      ENDIF
      ENDIF
860   CONTINUE
      M=0
850   CONTINUE
      LNXT(L)=K+1
      Q=LNXT(L)
      L=L+1
      DO 750 I=L,125
      Q=Q+1
      LNXT(I)=Q
750   CONTINUE

```

```
C*****  
C   THIS SEGMENT GENERATES THE ELEMENTS OF NSEQ  
C*****  
    F=1  
    DO 756 I=1,DFM  
      NSEQ(I)=F  
      F=F+1  
756  CONTINUE  
      RETURN  
      END
```

3.1.7 SUBROUTINE RADO : Optimal ordering scheme.

A. Description:

One of the optimal ordering strategies explained in chapter 1 has been implemented in this subroutine RADO. Array NSEQ created by subroutine DATAGEN contains initially the numbers 1 to DFM in ascending order. This subroutine makes use of that array. It checks for columns having minimum number of non-zero elements and alters NSEQ so that the column having the least number of non-zero elements appears first in NSEQ and subsequently is solved first by subroutine FACTO resulting in less number of fill-in terms. Though RADO builds the pivotal sequence it does not actually interchange the columns, instead only the two respective indices within table NSEQ are interchanged such that the near optimal pivotal sequence is built up step by step. The detailed flow chart for RADO is as shown in figure 3.7.

The steps involved are

1. Building up of near optimal pivotal index step by step with respect to fewest number of non-zero terms

in subsequent columns. This is done by looking at arrays NSEQ and NOZE.

2. All columns the index of which is contained in the pivotal columns are compared term by term with the pivotal column and their accompanying indexing and addressing information is altered. This done in the following manner.

If the processed column contains the pivotal index, the related matrix term is cancelled. If any row index of the pivotal column is not contained in the column under consideration, this index is added to the row indices in the table ITAG (fill-in terms). The fill-in terms are stored not only in the vacant locations at the end of the tables CE, ITAG and LNXT but also in other locations becoming vacant in the course of the simulation process. The next vacant location is always indicated by LF.

Whenever a term is cancelled or added, the respective addressing information in LNXT and LCOL respectively must be altered appropriately. Furthermore the book-keeping of non-zero terms must be updated.

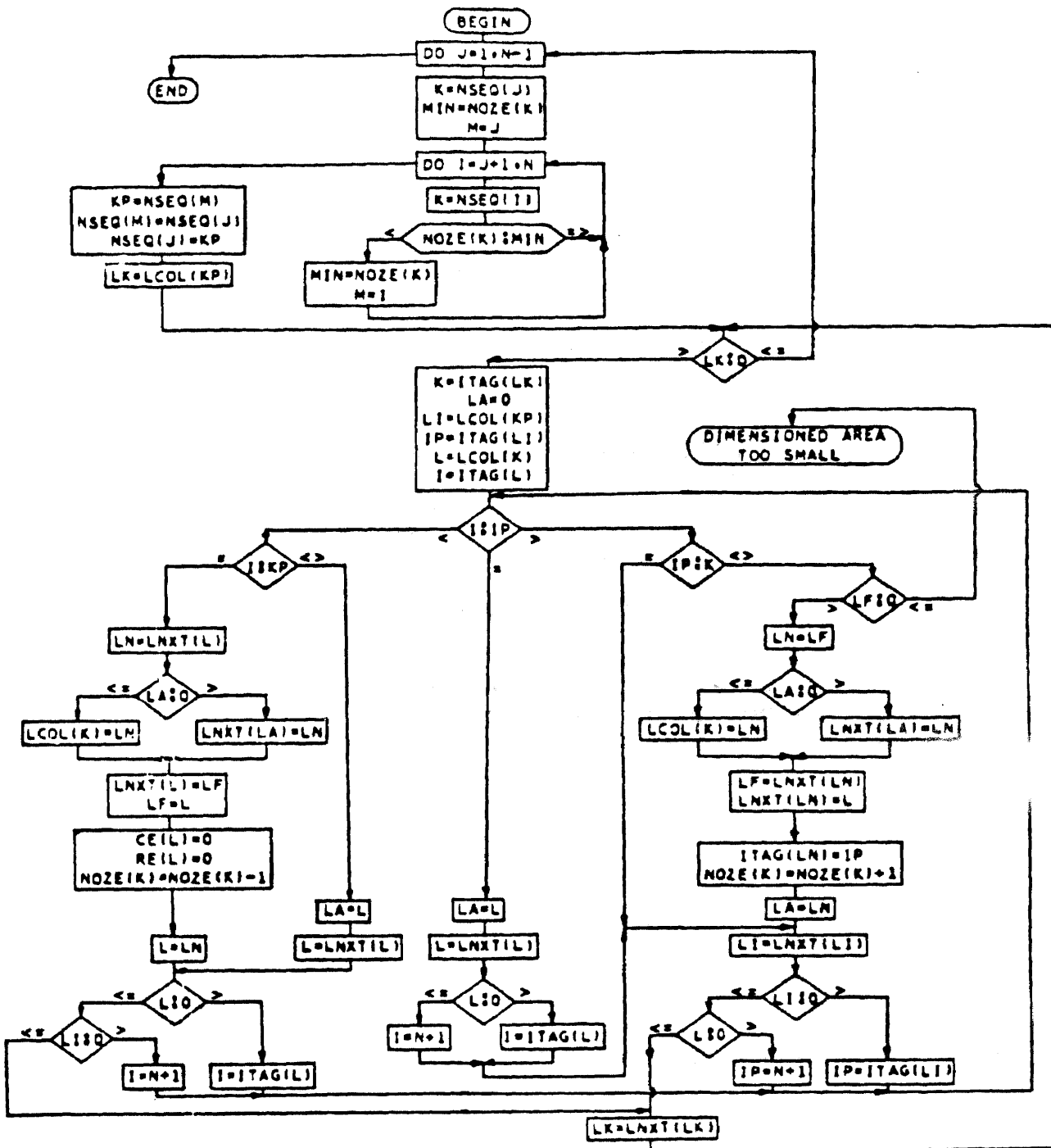


Fig 3-7 Flow chart of simulation and ordering subroutine (asymmetrical matrix)

B. List of the subroutine RADO

```

*****
*                               SUBROUTINE RADO                               *
*****

C   THIS SUBROUTINE DOES THE OPTIMAL ORDERING. THIS
C   SUBROUTINE MAKES USE OF ARRAY NSEQ WHICH HAS
C   COLUMN NUMBERS 1 TO DFM IN ASCENDING ORDER. IT
C   CHECKS FOR COLUMNS HAVING MINIMUM NUMBER OF NON
C   ZERO ELEMENTS AND ALTERS THE NSEQ SO THAT THE
C   COLUMN HAVING LESS NUMBER OF NONZERO ELEMENTS IS
C   SOLVED FIRST. HENCE RADO BUILDS THE PIVOTAL
C   SEQUENCE
C*****

      SUBROUTINE RADO(LCOL,NOZE,DE,ITAG,
                     LNXT,CE,RE,NSEQ)

      INTEGER DFM,SWN
      COMMON DFM,SWN
      INTEGER LCOL(DFM),NSEQ(DFM),LF
      INTEGER NOZE(DFM),ITAG(SWN),LNXT(SWN)
      REAL RE(SWN),DE(DFM),CE(SWN)
      COMMON LF

C*****
C FINDING COLUMN WITH LEAST NUMBER OF NONZERO ELEMENTS
C*****
      DO 5000 J=1,DFM-1
      K=NSEQ(J)
      MIN=NOZE(K)
      M=J
      DO 2000 I=J+1,DFM
      K=NSEQ(I)
      IF(NOZE(K).GE.MIN)GO TO 2000
      MIN=NOZE(K)
      M=I
      2000 CONTINUE

C****MODIFYING THE ADDRESSING INFORMATION*****
      KP=NSEQ(M)
      NSEQ(M)=NSEQ(J)
      NSEQ(J)=KP
      LK=LCOL(KP)
      210 IF(LK.LE.0)GO TO 5000
      K=ITAG(LK)
      LA=0
      LI=LCOL(KP)
      IP=ITAG(LI)

```

```

L=LCOL(K)
I=ITAG(L)
220 IF(I.LT.IP) THEN
IF(I.EQ.KP) THEN
LN=LNXT(L)
IF(LA.LE.0) THEN
LCOL(K)=LN
ELSE
LNXT(LA)=LN
ENDIF
LNXT(L)=LF
LF=L
CE(L)=0
RE(L)=0
NOZE(K)=NOZE(K)-1
L=LN
ELSE
LA=L
L=LNXT(L)
ENDIF
IF(L.LE.0) THEN
IF(LI.LE.0) THEN
240 LK=LNXT(LK)
GO TO 210
ELSE
I=DFM+1
GO TO 220
ENDIF
ELSE
I=ITAG(L)
GO TO 220
ENDIF
ELSEIF(I.EQ.IP) THEN
LA=L
L=LNXT(L)
IF(L.LE.0) THEN
I=DFM+1
ELSE
I=ITAG(L)
ENDIF
230 LI=LNXT(LI)
IF(LI.LE.0) THEN
IF(L.LE.0) THEN
GO TO 240
ELSE
IP=DFM+1
GO TO 220
ENDIF
ELSE
IP=ITAG(LI)

```

```
GO TO 220
ENDIF
ELSE
IF(IP.EQ.K) GO TO 230
IF(LF.GT.0) THEN
LN=LF
IF(LA.LE.0) THEN
LCOL(K)=LN
ELSE
LNXT(LA)=LN
ENDIF
LF=LNXT(LN)
LNXT(LN)=L
ITAG(LN)=IP
NOZE(K)=NOZE(K)+1
LA=LN
GO TO 230
ELSE

C*****ERROR MESSAGE IF SWN IS NOT LARGE ENOUGH*****
WRITE(6,*) ' DIMENSIONED AREA TOO SMALL '
ENDIF
ENDIF
5000 CONTINUE
RETURN
END
```

3.1.8 SUBROUTINE FACTO : Factorization of the optimally ordered scheme.

A. Description:

As discussed earlier in the conceptual model for load flow analysis and programming techniques in particular, Subroutine FACTO operates upon the simulated image created by subroutine RADO and does the actual reduction. The reduction process is guided by the sequence of column numbers in array NSEQ. The detailed flow chart is as shown in figure 3.8

The steps involved are

1. Read the array NSEQ.
2. Start with taking the reciprocal of the diagonal element of the column whose number appears first in the array NSEQ.
3. Divide the elements of the row corresponding to this column by the diagonal element and repeat for other row elements of the same row.
4. Further reduction consists of only those terms of the reduced residual matrix with subscripts corresponding to the row indices of the pivotal column which then have to be recalculated. For that purpose the

corresponding columns are compared term by term with the pivotal column in much the same way as in the subroutine RADO.

5. Return to the main routine.

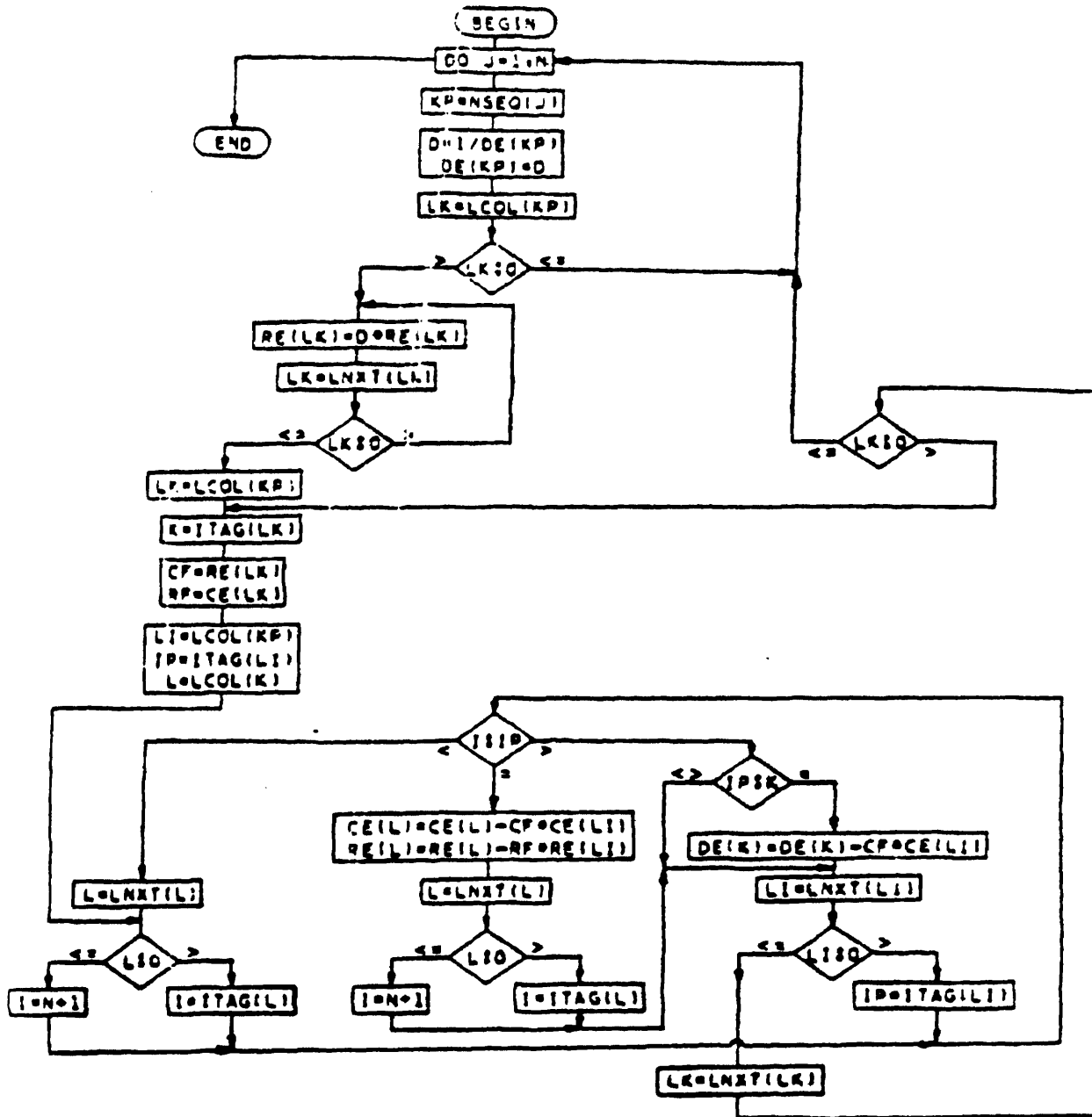


Fig 38 Flow chart of reduction subroutine (symmetrical matrix)

B. List of the subroutine FACTO

```

*****
*           SUBROUTINE FACTO           *
*****

C      THIS SUBROUTINE DOES THE PROCESS OF ACTUAL
C      REDUCTION.
C*****

      SUBROUTINE FACTO(LCOL,NOZE,DE,ITAG,
                      LNXT,CE,RE,NSEQ)

      INTEGER DFM,SWN
      COMMON DFM,SWN
      INTEGER LCOL(DFM),NOZE(DFM),NSEQ(DFM)
      INTEGER ITAG(SWN),LNXT(SWN)
      REAL CE(SWN),RE(SWN),DE(DFM)
      DO 1000 J=1,DFM
      KP=NSEQ(J)

C*****RECIPROCAL OF DIAGONAL ELEMENT*****
      D=1/DE(KP)
      DE(KP)=D
      LK=LCOL(KP)
      IF(LK.LE.0)GO TO 1000

C*****PROCESS THE ROW ELEMENTS*****
310  RE(LK)=D*RE(LK)
      LK=LNXT(LK)
      IF(LK.GT.0)GO TO 310
      LK=LCOL(KP)
320  K=ITAG(LK)
      CF=RE(LK)
      RF=CE(LK)
      LI=LCOL(KP)
      IP=ITAG(LI)
      L=LCOL(K)
330  IF(L.GT.0)THEN
      I=ITAG(L)
      ELSE
      I=DFM+1
      ENDIF
37   IF(I.LT.IP)THEN
      L=LNXT(L)
      GO TO 330
      ELSEIF(I.GT.IP)THEN
      IF(IP.EQ.K)DE(K)=DE(K)-CF*CE(LI)

```

```
41  LI=LNXT(LI)
    IF(LI.GT.0) THEN
    IP=ITAG(LI)
    GO TO 37
    ELSE
    LK=LNXT(LK)
    IF(LK.GT.0) THEN
    GO TO 320
    ELSE
    GO TO 1000
    ENDIF
    ENDIF
    ELSE
```

```
C*****GENERATE THE FILL IN TERMS*****
    CE(L)=CE(L)-CF*CE(LI)
    RE(L)=RE(L)-RF*RE(LI)
    L=LNXT(L)
    IF(L.GT.0) THEN
    I=ITAG(L)
    ELSE
    I=DFM+1
    ENDIF
    GO TO 41
    ENDIF
1000 CONTINUE
    RETURN
    END
```


3.1.9 SUBROUTINE SLAD : Adjustment for the slack bus.

A. Description:

Subroutine SLAD is used to adjust the array CON so as to delete the entries corresponding to the slack bus. This is done to ensure proper results from subroutine UTION which operates on array CON to get the solution vector.

The flow chart is as shown in figure 3.9.

The steps involved are

1. Since the voltage of V of the slack bus is the first entry in CON and the phase angle θ of the slack bus is the fifteenth entry in the CON for a 14 bus system, SLAD moves ΔP elements that follow ΔP_1 by one position up in array CON, and it moves ΔQ elements that follow ΔQ_1 by two positions up; one position to fill the vacated ΔP_{14} position and one to fill ΔQ_1 position.

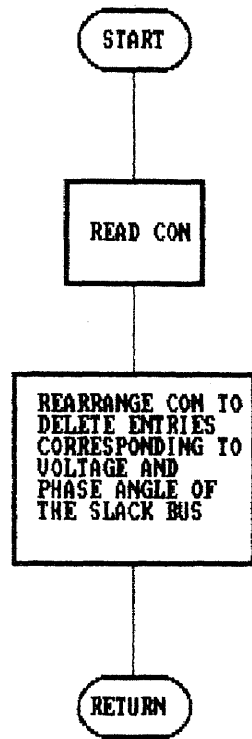


FIGURE 3.9 SUBROUTINE SLAD

B. List of the subroutine SLAD

```

*****
*                SUBROUTINE SLAD                *
*****
      SUBROUTINE SLAD(CON,NPN,NB)

C*****
C      SLAD ADJUSTS THE POWER MISMATCHES FOR SLACK BUS
C      SO THAT THEY DO NOT APPEAR IN CALCULATION BY
C      SHIFTING THE ARRAY ELEMENTS.
C*****

      DIMENSION CON(NPN)
      DO 10 I=2,NB
10     CON(I-1)=CON(I)
      DO 12 I=2,NB
12     CON(I+NB-2)=CON(I+NB)
      RETURN
      END

```

3.1.10 SUBROUTINE UTION : Final solution vector.

A. Description:

This is a direct solution subroutine. This subroutine reads in the array CON of power mismatches and operates upon it to get the solution vector. The operations done by FACTO on the matrix elements are repeated by UTION on the elements of CON to get the solution vector. The flow chart is as shown in figure 3.10.

The steps involved are similar to the steps in subroutine FACTO.

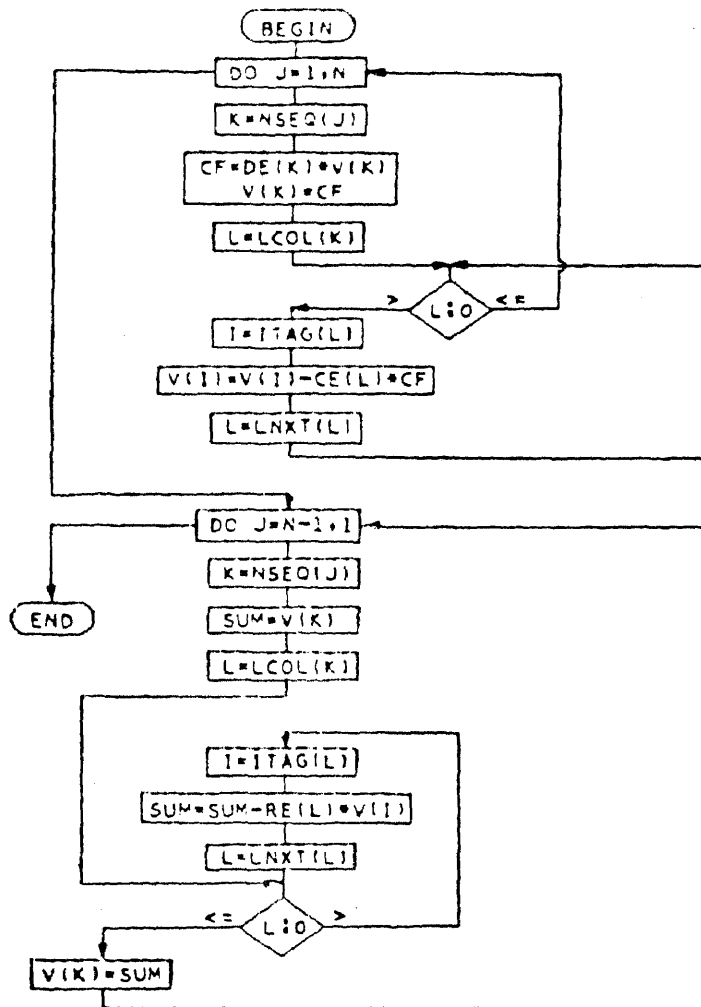


Fig 3.10 Flow chart of direct solution subroutine (asymmetrical matrix)

B. List of the subroutine UTION

```

C*****
*           SUBROUTINE UTION           *
C*****

C      THIS IS THE DIRECT SOLUTION SUBROUTINE
C*****

      SUBROUTINE UTION(LCOL,NOZE,DE,ITAG,LNXT,
                     CE,RE,NSEQ,CON,NPN)

      INTEGER DFM,SWN
      COMMON DFM,SWN
      INTEGER LCOL(DFM),NOZE(DFM),ITAG(SWN)
      INTEGER LNXT(SWN),NSEQ(DFM)
      REAL CE(SWN),RE(SWN),DE(DFM),CON(NPN)
      DO 1001 J=1,DFM
      K=NSEQ(J)
      CF=DE(K)*CON(K)
      CON(K)=CF
      L=LCOL(K)
3      IF(L.LE.0)GO TO 1001
      I=ITAG

C*****PROCESSING THE ELEMENTS OF CON*****
      CON(I)=CON(I)-CE(L)*CF
      L=LNXT(L)
      GO TO 3
1001  CONTINUE
      DO 1010 J=DFM-1,1,-1
      K=NSEQ(J)
      SUM=CON(K)
      L=LCOL(K)
7      IF(L.LE.0)GO TO 6
      I=ITAG(L)

C*****FINAL SOLUTION*****
      SUM=SUM-RE(L)*CON(I)
      L=LNXT(L)
      GO TO 7
6      CON(K)=SUM
1010  CONTINUE
      RETURN
      END

```

3.2 Exploiting sparsity without optimal ordering

(Program NOOPTIMAL).

The program NOOPTIMAL does not resort to optimal ordering. The results of NOOPTIMAL can be compared to those of JEET to seek the advantages of resorting to optimal ordering. The program has been initialized with data from 14 bus system. NOOPTIMAL is identical to JEET in all respects except in the subroutine DATAGEN. The description and listing of the modified DATAGEN is provided in the next section.

3.2.1 MODIFIED SUBROUTINE DATAGEN : bypassing optimal ordering.

A. Description

Subroutine DATAGEN reads the matrix FM and forms arrays to exploit the sparsity of FM and ensures a packed matrix storage scheme. We modify DATAGEN so that array NOZE which should contain the count of non-zero elements in each column, contains numbers 1 to DFM in ascending order instead. This creates an understanding that the columns have been arranged in the ascending order of their non-zero elements. Hence subroutine RADO does not rearrange the array NSEQ initialized with

numbers 1 to DFM in ascending order as the columns are already arranged in the order of their non-zero elements.

Hence the order in which the columns are to be operated remains the same as in original jacobian, eliminating optimal ordering from the actual program. Only the portion of modified DATAGEN has been included in the listing.

The flow chart is as shown in figure 3.11.

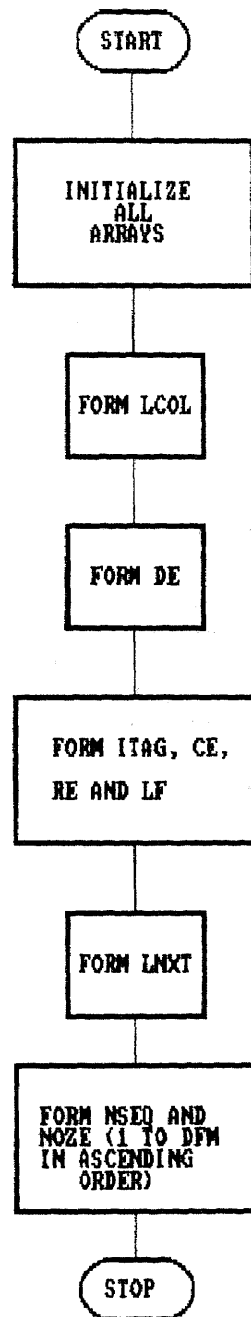


FIGURE 3.11 MODIFIED SUBROUTINE DATAGEN

B. List of the subroutine DATAGEN from program

NOOPTIMAL

```
*****
*   SUBROUTINE DATAGEN FROM PROGRAM NOOPTIMAL   *
*****
```

```
C   THIS LISTING CONTAINS ONLY THE VARIATION OF
C   NOOPTIMAL FROM JEET AND NOT THE ENTIRE LISTING.
C   THIS PROGRAM IS USED TO SOLVE A FOURTEEN BUS
C   POWER SYSTEM EXPLOITING SPARSITY BUT NOT USING
C   OPTIMAL ORDERING. DATAGEN READS FM AND FORMS
C   ARRAYS TO EXPLOIT THE SPARSITY OF FM. THIS
C   ENSURES OPTIMUM COMPUTATION TIME AND A PACKED
C   MATRIX STORAGE SCHEME. IN THIS WE SLIGHTLY MODIFY
C   DATAGEN SO THAT NOZE DOES NOT ACTUALLY CALCULATE
C   NON ZERO ELEMENTS IN EACH COLUMN BUT HAS NUMBERS
C   1 TO DFM IN ASCENDING ORDER INSTEAD SO THAT RADO
C   DOES NOT CHANGE NSEQ AS THE COLUMNS ARE SEEN
C   TO HAVE NONZERO ELEMENTS IN ASCENDING ORDER.
C   HENCE THE ORDER OF COLUMNS IN WHICH THEY ARE TO
C   BE OPERATED REMAINS THE SAME AS ORIGINAL JACOBIAN
C   HENCE OPTIMAL ORDERING CAN BE ELLIMINATED FROM
C   THE ACTUAL PROGRAM.
```

```
C*****
```

```
      SUBROUTINE DATAGEN(FM,LCOL,NOZE,DE,ITAG
                        LNXT,CE,RE,NSEQ)
```

```
C*****
C   THE SEGMENTS OF THE SUBROUTINE THAT APPEAR HERE
C   ARE THE ONLY CHANGES IN THE ORIGINAL DATAGEN.
C   REST OF THE SUBROUTINE REMAINS UNCHANGED.
```

```
C*****
```

```
C   THIS SEGMENT GENERATES DE
```

```
C*****
```

```
      TOT=0
      DO 30 J=1,DFM
      DO 40 I=1,DFM
      IF(I.EQ.J)DE(J)=FM(I,J)
```

```
40    CONTINUE
```

```
30    CONTINUE
```

```
C*****
C   THIS SEGMENT GENERATES THE ELEMENTS NSEQ AND
C   MANIPULATES ELEMENTS OF NOZE.
```

```
C*****
```

```
      F=1
```

```
DO 756 I=1,DFM  
NSEQ(I)=F  
NOZE(I)=F  
F=F+1  
756 CONTINUE  
RETURN  
END
```

3.3 Testing under different loading conditions

(Program DIFFLOAD).

3.3.1 Description:

This program is initialized to test the 14 bus system under different loading conditions. It randomly generates the values for voltages and phase angles for load buses. It only generates the values for phase angles for generator buses, and not values for voltages as they are system specified.

Using this data we calculate P and Q which are used by the program as system specified P and Q. Then the program proceeds like JEET. To change the loading condition the value of integer variable SEED in the data statement must be changed. The flowchart is as shown in the figure 3.12.

This listing contains only the variation of DIFFLOAD from JEET and not the complete listing.

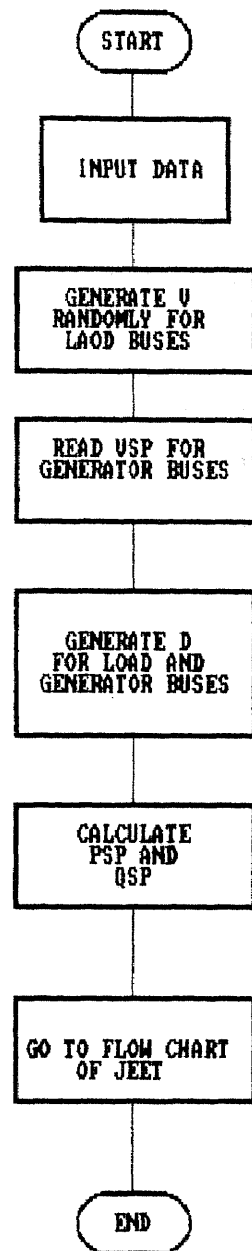


FIGURE 3.12 PROGRAM DIFFLOAD

3.3.2 Partial listing of DIFFLOAD

```

*****
*      PARTIAL LISTING OF PROGRAM DIFFLOAD      *
*****

C      THIS PROGRAM IS USED TO TEST THE FOURTEEN BUS
C      SYSTEM UNDER DIFFERENT LOADING CONDITIONS. WE
C      RANDOMLY GENERATE V AND THETA FOR LOAD BUSES AND
C      SINCE V IS SPECIFIED FOR GENERATOR BUSES WE
C      GENERATE THETA ONLY FOR THESE BUSES USING THIS WE
C      CALCULATE P AND Q WHICH ARE USED BY THE PROGRAM
C      AS INPUT PSP AND QSP. THEN THE PROGRAM PROCEEDS
C      LIKE JEET. TO CHANGE THE LOADING CONDITION SEED
C      HAS TO BE CHANGED SO THAT NEW VALUES FOR FOR V
C      AND THETA COULD BE GENERATED.
C*****

      INTEGER DFM,SWN
      COMMON DFM,SWN
      REAL PSP(14),QSP(14),V(14),D(14), CON(28),
1G(14,14),B(14,14),VSP(14),FM(22,22),P(14),Q(14),
2DE(22),RE(125),CE(125)
      REAL JK11(14,14),JK12(14,14)
      REAL JK21(14,14),JK22(14,14)
      REAL A,Z,NUMBER,C,T,NUMBER3(14)
      INTEGER FJM,MFJM,LF,NR,ITEST,SEED
      INTEGER LCOL(22),NOZE(22),NSEQ(22)
      INTEGER ITAG(125),LNXT(125),DOZE(22)
      DATA SEED,A,Z,C,T/1331,0.2,0.9,20,-10/
      DATA NB,NLINES,NL,K1,K2,NPN/14,20,9,10,11,28 /
      DATA DFM,SWN/22,125/
      DATA CRIT,KMAX/0.0005,40/
      DATA NR,FJM,MFJM/1,0,0/
      DATA V(1),D(1)/1.06,0.0/
      COMMON LF
      OPEN(UNIT=78,FILE='DIFFLOAD.OUT',STATUS='NEW')

C*****THIS READS THE LINE DATA*****
      CALL INIT(NB,B,G,NBL,NLINES)

C*****THIS GENERATES V FOR LOAD BUSES USING SEED*****
      CALL GENV(SEED,NUMBER,V,K1,A,Z)
C      THIS READS VSP FOR GENERATOR BUSES.
      CALL SPECV(NB,K2,V)
C      THIS GENERATES D FOR LOAD & GENERATOR BUSES
      CALL DGEN(SEED,NB,NUMBER,D,C,T,NUMBER3)
C      THIS CALCULATES P AND Q FROM B,G,V AND D
      CALL PQCAL(P,Q,V,D,B,G,NB)

```

```

C      K1 IS THE NO. OF LOAD NODES + 1
C      K2 IS THE NO. OF LOAD NODES + 2, NEEDED TO READ
C      VSP
C      NPN = 2*NB
C      DFM = NB + NL -1 AND IS THE DIMENSION OF FM
C      THIS EQUATES P AND Q CALCULATED ABOVE TO PSP AND
C      QSP.
      DO 15 I=1,NB
      PSP(I)=P(I)
      QSP(I)=Q(I)
15     CONTINUE
      SUBROUTINE GENV(SEED,NUMBER,V,K1,A,Z)
      INTEGER SEED,K1
      REAL NUMBER,V(K1),A,Z
      DO 100 I = 1,K1
      NUMBER = RAN(SEED)
      V(I) = (NUMBER * A) + Z
10     FORMAT(1X, F8.4, 3X, F8.4, 3X, F8.4)
10     CONTINUE
      RETURN
      END
      SUBROUTINE DGEN(SEED,NB,NUMBER,D,C,T,NUMBER3)
      INTEGER SEED,NB
      REAL NUMBER,D(NB),NUMBER3(NB),C,T
      SEED=1331
      DO 300 I=1,NB
      NUMBER = RAN(SEED)
      NUMBER3(I) = (NUMBER * C) + T
      D(I) = (NUMBER3(I) * 2 * 3.14) / 360.0
300    CONTINUE
10     FORMAT(1X, F8.4, 3X, F8.4, 3X, F8.4)
      RETURN
      END
      SUBROUTINE INIT(NB,B,G,NBL,NLINES)
      INTEGER NB,NBL,NLINES
      REAL B(NB,NB),G(NB,NB)
      OPEN(UNIT=15,FILE='DIFFLOAD.IN',STATUS='OLD')
      DO 5 I=1,NB
      DO 5 J=1,NB
      B(I,J)=0.0
      G(I,J)=0.0
5     CONTINUE
      NBL=NB+NLINES
100    FORMAT(2I3,2F10.5)
      DO 10 L=1,NBL
      READ(15,*)I,J,G(I,J),B(I,J)
      G(J,I)=G(I,J)
      B(J,I)=B(I,J)
10     CONTINUE
      RETURN
      END

```

```
SUBROUTINE SPECV(NB,K2,V)
INTEGER NB,K2
REAL V(NB)
OPEN(UNIT=17,FILE='VSP.IN',STATUS='OLD')
DO 111 I=K2,NB
111 READ(17,*)V(I)
RETURN
END
```


CHAPTER FOUR

HARMONIC LOAD FLOW ANALYSIS

While dealing with alternating voltages and currents in ideal situations, it is assumed that they have sinusoidal waveform. But it is nearly impossible to realize such a waveform in practice. Analyzing a non-sinusoidal waveform according to Fourier series shows that it consists of

1. fundamental.
2. an infinite number of other sinusoidal waves whose frequencies are an integral multiple of the fundamental.

Now the fundamental and its higher multiples form a harmonic series [10].

4.1 AC System Harmonics

Today's power system harmonic problems can be traced to a number of factors [11]. These factors include

1. The drastic increase in the use of nonlinear loads made possible by new technologies. Silicon Controlled Rectifiers (SCR's), power transistors and micro-processors are the most important in this respect.

These loads are generated, propagated, and injected in every line and feeder in the system. This can cause higher losses and if severe, can cause insulation degradation, flashover, equipment overheating and failure [8].

2. Another important aspect is the drastic change in the design philosophy of all power equipment and load equipment. In order to be competitive, power devices and equipment are critically designed. In the case of iron core devices, this means that the operating points are, more into one nonlinear characteristics, resulting in sharp rise in harmonics.

4.2 Summary of Harmonic Studies

Xia and Heydt, [12] and later, Grady, Kraft and Hart studied the reformulation of the Newton-Raphson power flow study to include harmonics. The conventional Newton-Raphson iteration method has to be reformulated to include nonlinear loads. This reformulation is based on the reduction to zero of the mismatch active power and reactive voltamperes, the unbalance current at harmonic frequencies and the mismatch apparent voltampares.

Network voltages and currents are represented by Fourier series. Harmonic bus voltage magnitudes and angles are unknowns for which additional equations are needed. The additional equations are based on Kirchhoff's current law for each harmonic and on conservation of apparent voltampers at certain buses. The assumption of a balanced bilateral system permits the exclusion of even numbered harmonics. The restriction to delta connected nonlinear devices eliminates zero sequence currents.

Triple harmonics ($3n$, $n=1,2,3,\dots$) in a balanced system are strictly zero sequence, and since zero sequence currents cannot flow from the network into a delta or ungrounded wye connected device, all triple harmonics are usually excluded from the formulation.

Power system components, loads and generators which produce harmonics in an otherwise pure fundamental frequency network are termed nonlinear. Standard PQ or PV power flow buses with no converters or other nonlinear devices are termed linear. All other buses are termed nonlinear.

Consider a voltage $v(t)$ and current $i(t)$ expressed in terms of its rms harmonic components [13]

$$v(t) = a_0 + \sum_{k=1}^{\infty} \left[a_k \sin (k\omega_0 t + \phi_k) \right] \quad (4.2-1)$$

$$i(t) = c_0 + \sum_{k=1}^{\infty} \left[c_k \sin (k\omega_0 t + \theta_k) \right] \quad (4.2-2)$$

The active power is given by P ,

$$P = a_0 c_0 + \sum_{k=1}^{\infty} \left[a_k c_k \cos (\theta_k - \phi_k) \right] \quad (4.2-3)$$

The reactive voltamperes are given by Q ,

$$Q = \sum_{k=1}^{\infty} \left[a_k c_k \sin (\theta_k - \phi_k) \right] \quad (4.2-4)$$

The apparent voltampere level is given by S

$$S = \sqrt{\left(\sum_{k=0}^{\infty} a_k^2 \right) \left(\sum_{l=0}^{\infty} c_l^2 \right)} \quad (4.2-5)$$

For $k \rightarrow 0$ to ∞ and $l \rightarrow 0$ to ∞

In the case of sinusoidal $v(t)$ and $i(t)$

$$S^2 = P^2 + Q^2 \quad (4.2-6)$$

However in non-sinusoidal cases (4.2-6) does not hold good. The discrepancy is termed as distortion voltamperes D ,

$$D = \sqrt{S^2 - P^2 - Q^2} \quad (4.2-7)$$

4.2.1 Reformulation Of NR Method To Allow For Harmonics

Consider bus 1 to be the slack bus, buses 2 through N to be conventional load buses and buses M through n to be nonlinear load buses. The power balance equations are that ΔP and ΔQ at all non-slack buses is zero for all harmonics. The functional forms of ΔP and ΔQ as a function of V_{bus} and θ_{bus} (i.e., bus voltage magnitudes and phase angles) is as in the conventional case but the Y_{bus} must be modified at harmonic frequencies. Newton's method is formulated by forcing the appropriate mismatches, ΔM , to zero using a Jacobian matrix, J , and obtaining appropriate correction terms, ΔU . Hence

$$\Delta M = J \Delta U \quad (4.2-8)$$

where all elements in the matrix are sub-vectors and sub-matrices partitioned from ΔM , J and ΔU .

Thus in the case of the harmonic power flow study, the matrix formulation of the problem is [19]:

$$\begin{bmatrix} \Delta W \\ \Delta I(3) \\ \cdot \\ \Delta \dot{I}(h) \\ \Delta I(1) \end{bmatrix} = \begin{bmatrix} J(1) & J(3) & \dots & J(h) & 0 \\ TG(3,1) & TG(3,3) & \dots & TG(3,h) & H(3) \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ TG(h,1) & TG(h,3) & \dots & TG(h,h) & H(h) \\ TG(1,1) & TG(1,3) & \dots & TG(1,h) & H(1) \end{bmatrix} \begin{bmatrix} \Delta V(1) \\ \Delta V(3) \\ \cdot \\ \dots \\ \Delta \phi \end{bmatrix}$$

$$(4.2-9)$$

Note that the first $2N$ components of ΔI^k do not contain nonlinear device load current.

The k th harmonic bus voltage magnitude and angle corrections vector at all buses is

$$\Delta V^k = \left[\begin{array}{cccccc} V_1^k & \Delta \theta_1^k & V_1^k & \dots & V_n^k & \Delta \theta_n^k & V_n^k \end{array} \right]^T, \quad k=1,3,5,7,\dots,h \quad (4.2-13)$$

The control variable corrections for each nonlinear device are contained in

$$\Delta \phi = [\Delta \beta_M, \Delta \tau_M, \dots, \Delta \beta_n, \Delta \tau_n]^T \quad (4.2-14)$$

where β and τ are nonlinear device control variables.

The conventional fundamental frequency jacobian matrix is J^1 and is given by

$$\left[\begin{array}{c|c} \frac{JK11}{JK21} & \frac{JK12}{JK22} \end{array} \right] \quad (4.2-15)$$

where

$$JK11_{ik} = \delta P_i / \delta \theta_k = V_i V_k (G_{ik} \sin \theta_{ik} - B_{ik} \cos \theta_{ik}) \quad (4.2-16)$$

$$JK12_{ik} = V_k \delta P_i / \delta V_k = V_i V_k (G_{ik} \cos \theta_{ik} + B_{ik} \sin \theta_{ik}) \quad (4.2-17)$$

$$JK21_{ik} = - JK12_{ik} \quad (4.2-18)$$

$$JK22_{ik} = JK11_{ik} \quad (4.2-19)$$

and the diagonal elements are given by

$$JK11_{ii} = V^2_i B_{ii} - Q_i \quad (4.2-20)$$

$$JK12_{ii} = V^2_i G_{ii} + P_i \quad (4.2-21)$$

$$JK21_{ii} = -V^2_i G_{ii} + P_i \quad (4.2-22)$$

$$JK22_{ii} = -V^2_i B_{ii} + Q_i \quad (4.2-23)$$

The harmonic K Jacobian matrix $J^{(k)}$, has the form

$$= \left[\begin{array}{c} \text{0}_{2N \text{ by } 2n} \\ \hline \text{partial derivatives of PCAL \& QCAL} \\ \text{with respect to } V^{(k)} \text{ and } \theta^{(k)}. \\ \text{These are formed in the conventional} \\ \text{way. (nonlinear buses only)} \end{array} \right] \quad (4.2-24)$$

where $0_{j,k}$ denotes a j by k array of zeroes.

The TG matrices are comprised of two parts

$$(TG)_{(k,j)} = \left\{ \begin{array}{ll} T(k,k) + G(k,k), & k=j \\ G(k,j), & k \neq j \end{array} \right\}; k=1,3,5,\dots,h$$

(See Appendix D for an example) (4.2-25)

where $T(k,k)$ is an array of partial derivatives of the k th harmonic injection currents with respect to the k th harmonic bus voltages as derived from the system admittance matrix. Note that the fundamental frequency T matrix is dimensioned not to contain rows corresponding to the linear buses (top $2N$ rows). The k th harmonic bus injection current vector is defined as

$$I^k = Y^k * V^k \quad (4.2-26)$$

where

Y^k : kth harmonic system admittance matrix

V^k : kth harmonic bus voltage vector.

The injection at any bus t may be written as the t th row of Y^k multiplied by V^k or

$$I_t^k = \sum_{m=1}^n Y_{t,m}^k * V_m^k = a_t^k + j d_t^k \quad (4.2-27)$$

where

a_t^k : Real component of current

d_t^k : Imaginary component of current

n : Number of buses.

Writing V_m^k and $Y_{t,m}^k$ in phasor notation,

$$\underline{V}_m^k = V_m^k e^{j\theta(k)}, \quad \underline{Y}_{t,m}^k = Y_{t,m}^k e^{j\phi(k)} \quad (4.2-28)$$

then

$$a_t^k + j d_t^k =$$

$$= \sum_{m=1}^n Y_{t,m}^k V_m^k \left[\cos(\phi_{t,m}^k + \theta_m^k) + j \sin(\phi_{t,m}^k + \theta_m^k) \right] \quad (4.2-29)$$

The partial derivatives of injection currents with respect to bus voltages are,

therefore

$$\left[\frac{1}{V_m} \right]^k * \left[\frac{\delta a_t}{\delta \theta_m} \right]^k = - Y_{t,m}^k \sin(\phi_{t,m}^k + \theta_m^k) \quad (4.2-30)$$

$$\left[\frac{\delta a_t}{\delta V_m} \right]^k = Y_{t,m}^k \cos(\phi_{t,m}^k + \theta_m^k) \quad (4.2-31)$$

$$\left[\frac{1}{V_m} \right]^k * \left[\frac{\delta d_t}{\delta \theta_m} \right]^k = Y_{t,m}^k \cos(\phi_{t,m}^k + \theta_m^k) \quad (4.2-32)$$

$$\left[\frac{\delta d_t}{\delta V_m} \right]^k = Y_{t,m}^k \sin(\phi_{t,m}^k + \theta_m^k) \quad (4.2-33)$$

Now $T(k,k)$ is formed using the relation

$$\left[\Delta I^k \right] = \left[\Delta T(k,k) \right] * \left[\Delta V^k \right] \quad (4.2-34)$$

Where

$$\Delta I^k = \left[\Delta a_1^k, \Delta d_1^k, \dots, \Delta a_n^k, \Delta d_n^k \right] \quad (4.2-35)$$

$$\Delta V^k = \left[V_1^k, \Delta \theta_1^k, \Delta V_1^k, \dots, V_n^k, \Delta \theta_n^k, \Delta V_n^k \right] \quad (4.2-36)$$

and the individual terms of $T(k,k)$ are calculated from Equations (4.2-31 through 4.2-33). Note that $T(k,k)$ has the same sparsity as Y^k . Appendix D gives the example to calculate $T(k,k)$ matrices.

The partial derivatives for the device load currents can be obtained as follows:

The bus injection current vector is given by

$$i(t) = \sum_{k=1}^h \left[c^k \left[\cos(k\omega_0 t) \cos(\phi^k) - \sin(k\omega_0 t) \sin(\phi^k) \right] \right] \quad (4.2-37)$$

or in phasor notation with sine reference,

$$i(t) = \sum_{k=1}^h \left[a^k \cos(k\omega_0 t) + d^k \sin(k\omega_0 t) \right] \quad (4.2-38)$$

where $I^{(k)}$ is the magnitude of the k th harmonic phasor component of $i(t)$, and a^k and d^k signify the real and imaginary components. Since $i(t)$ is formulated as load current, the load currents are the same and may be written as

$$g(t) = i(t) = \sum_{k=1}^h \left[f^k \cos(k\omega_0 t) + x^k \sin(k\omega_0 t) \right] \quad (4.2-39)$$

Then,

$$f^k = a^k ; x^k = d^k \quad (4.2-40)$$

The partial derivatives of f^k and x^k with respect to v^j , θ^j are the sums of the partial derivatives of the individual terms and the G matrix consists of these elements. The partial derivatives are obtained as shown. The truncated Fourier series of the line commutated converter current can be expressed by
(4.2-38)

where

$$x^k = d^k = \frac{2}{\pi} \int_{\alpha}^{\alpha+\pi} i_a(t) \cos(k\omega_0 t) d(\omega_0 t) \quad (4.2-41)$$

$$f^k = a^k = \frac{2}{\pi} \int_{\alpha}^{\alpha+\pi} i_a(t) \sin(k\omega_0 t) d(\omega_0 t) \quad (4.2-42)$$

Equation (4.2-38) has constant, exponential, and forcing terms in each of the six possible periods. Currents f^k and x^k can be calculated by performing the integration separately for each period and adding the results. Within each period, the three components can also be integrated separately and summed. Let c^1 , $E^1 e^{pt}$, and $F^1(t)$ represent the constant, exponential, and forcing terms in period 1.

Then

$$x^k = \frac{2}{\pi} \sum_{l=1}^6 \left[\int_{B_l}^{T_l} c^1 \cos(k\omega_0 t) d(\omega_0 t) \right. \\ \left. + \int_{B_l}^{T_l} E^1 e^{pt} \cos(k\omega_0 t) d(\omega_0 t) \right. \\ \left. + \int_{B_l}^{T_l} F^1(t) \cos(k\omega_0 t) d(\omega_0 t) \right] \quad (4.2-43)$$

$$\begin{aligned}
 f = \frac{k}{2\pi} \sum_{l=1}^6 & \left[\int_{B_1}^{T_1} c^l \sin(kw_0t) d(w_0t) \right. \\
 & + \int_{B_1}^{T_1} E^l e^{P^l t} \sin(kw_0t) d(w_0t) \\
 & \left. + \int_{B_1}^{T_1} F^l(t) \sin(kw_0t) d(w_0t) \right] \quad (4.2-44)
 \end{aligned}$$

where

l : Period

B_1 : Lower integral limit of period l

T_1 : Upper integral limit of period l .

The necessary integrals are [18]

$$\int_{B_1}^{T_1} c^l \cos(kw_0t) d(w_0t) = (c^l/k) \sin(kw_0t) \quad \left. \begin{array}{l} w_0t=T_1 \\ w_0=B_1 \end{array} \right] \quad (4.2-45)$$

$$\int_{B_1}^{T_1} c^l \sin(kw_0t) d(w_0t) = (c^l/k) \cos(kw_0t) \quad \left. \begin{array}{l} w_0t=T_1 \\ w_0=B_1 \end{array} \right] \quad (4.2-46)$$

$$\begin{aligned}
 & \int_{B_1}^{T_1} E^{(1)} e^{pt} \cos(kw_0 t) d(w_0 t) = \\
 & = \frac{E^1 e^{(p^1/w_0)t}}{(p^1/w_0)^2 + k^2} \left[(p^1/w_0) \cos(kw_0 t) + k \sin(kw_0 t) \right]_{w_0 t=B_1}^{w_0 t=T_1} \\
 & \hspace{20em} (4.2-47)
 \end{aligned}$$

$$\begin{aligned}
 & \int_{B_1}^{T_1} E^1 e^{pt} \sin(kw_0 t) d(w_0 t) = \\
 & = \frac{E^1 e^{(p^1/w_0)t}}{(p^1/w_0)^2 + k^2} \left[(p^1/w_0) \sin(kw_0 t) - k \cos(kw_0 t) \right]_{w_0 t=B_1}^{w_0 t=T_1} \\
 & \hspace{20em} (4.2-48)
 \end{aligned}$$

The forcing function is comprised of sine series terms and these integrals also may be performed term by term. $G(k, j)$ and $G(k, k)$ represent the partial derivatives of f, x with respect to applied network voltages. These partial derivatives are obtained directly from the integrals of the forcing functions previously described, the partial derivatives evaluated at the lower limits are subtracted from the partial derivatives evaluated at the upper limits. Actually, T_1 and B_1 are functions of the commutation angle μ which is, in turn, a function of the applied voltages.

Hence $G^{(k,j)}$ in matrix form is given by

$0_{2N, 2N}$	$0_{2N, 2m}$		
	$\frac{\frac{\partial \delta f_M^k}{\partial V_M^j \delta \theta_M^j}}{\frac{\partial \delta f_M^k}{\partial V_M^j \delta \theta_M^j}}$	$\frac{\frac{\partial \delta f_M^k}{\partial V_M^j \delta \theta_M^j}}{\frac{\partial \delta f_M^k}{\partial V_M^j \delta \theta_M^j}}$	$\frac{\frac{\partial \delta f_M^k}{\partial V_M^j \delta \theta_M^j}}{\frac{\partial \delta f_M^k}{\partial V_M^j \delta \theta_M^j}}$
	$\frac{\frac{\partial \delta x_M^k}{\partial V_M^j \delta \theta_M^j}}{\frac{\partial \delta x_M^k}{\partial V_M^j \delta \theta_M^j}}$	$\frac{\frac{\partial \delta x_M^k}{\partial V_M^j \delta \theta_M^j}}{\frac{\partial \delta x_M^k}{\partial V_M^j \delta \theta_M^j}}$	$\frac{\frac{\partial \delta x_M^k}{\partial V_M^j \delta \theta_M^j}}{\frac{\partial \delta x_M^k}{\partial V_M^j \delta \theta_M^j}}$
$0_{2m, 2N}$	0	...	0
	0	0	$\frac{\frac{\partial \delta f_n^k}{\partial V_n^j \delta \theta_n^j}}{\frac{\partial \delta f_n^k}{\partial V_n^j \delta \theta_n^j}}$
	0	0	$\frac{\frac{\partial \delta x_n^k}{\partial V_n^j \delta \theta_n^j}}{\frac{\partial \delta x_n^k}{\partial V_n^j \delta \theta_n^j}}$

(4.2-49)

The calculation for $G^{(k,j)}$ matrices has been explained in Appendix D with an example. The above matrix consists of partial derivatives of the k th harmonic nonlinear device load currents f^k, x^k with respect to the j th harmonic applied voltages as determined by the nonlinear device equations. Note that the fundamental frequency G matrices are dimensioned not to contain rows corresponding to the linear buses (top $2N$ rows).

The H matrix is a diagonal matrix of individual 2 by 2 submatrices which contain partial derivatives of nonlinear device load currents f^k, x^k with respect to nonlinear device control variables, or

$$H(k) = \text{diag} \begin{bmatrix} \frac{\delta f_t^k}{\delta \beta_t} & \frac{\delta f_t^k}{\delta \tau_t} \\ \frac{\delta x_t^k}{\delta \beta_t} & \frac{\delta x_t^k}{\delta \tau_t} \end{bmatrix} \quad \begin{array}{l} t=M, \dots, n \\ k=1, 3, 5, 7, \dots, h \end{array} \quad (4.2-50)$$

Also

h: Highest harmonic considered

m: Number of nonlinear buses

M: First nonlinear bus

n: Number of buses (linear and nonlinear)

N: Number of linear buses.

The H^k matrices are determined using the method of finite differences. Currents f^k, x^k are evaluated using slightly different values of nonlinear device control variables β or τ . The derivatives are then approximated using

$$\begin{bmatrix} \frac{\delta f^k}{\delta \beta} & \frac{\delta f^k}{\delta \tau} \\ \frac{\delta x^k}{\delta \beta} & \frac{\delta x^k}{\delta \tau} \end{bmatrix} \approx \begin{bmatrix} \frac{\Delta f^k}{\Delta \beta} & \frac{\Delta f^k}{\Delta \tau} \\ \frac{\Delta x^k}{\Delta \beta} & \frac{\Delta x^k}{\Delta \tau} \end{bmatrix} \quad (4.2-51)$$

The dimensions of the Harmonic power flow Jacobian matrices are shown in Table 4.1. Also the nomenclature for the terms used is given in Appendix C. The calculation for $H^{(k)}$ has been explained in Appendix D with an example.

Matrix	Dimensions (rows, columns)
ΔW	$2n, 1$
$\Delta I(1)$	$2m, 1$
$\Delta I(k), k > 1$	$2n, 1$
$\Delta V(k), k \geq 1$	$2n, 1$
$\Delta \phi$	$2m, 1$
$\Delta J(k), k \geq 1$	$2n, 2n$
$\Delta T(1,1)$	$2m, 2n$
$\Delta G(1,1)$	$2m, 2n$
$\Delta T(1,1), k > 1$	$2n, 2n$
$\Delta G(1,1), k > 1$	$2n, 2n$
$\Delta G(k,j), k > 1, j \geq 1$	$2n, 2n$
$H(1)$	$2m, 2m$
$H(k), k > 1$	$2m, 2m$

Table 4.1 Harmonic Power Flow
Jacobian Related Matrix Dimensions

where

j, k : Harmonic multiples

n : Number of buses (linear and nonlinear)

m : Number of nonlinear buses.

Conventional power flow studies are formulated on basis that power sources are system generators and power "sinks" are the loads. Harmonic power flow studies on the other hand, are more general in that loads may be the "source" of harmonic energy. The nature of harmonic signals depends on the type of nonlinearity.

Now counting the unknowns in the power flow

Fundamental bus voltage magnitude and angle
 at all but the swing bus..... $2(n-1)$

Real and reactive power at the swing bus..... 2

Harmonic bus voltage magnitude and angle
 at all buses..... $2nh$

Total reactive voltamperes at each nonlinear bus..... m

Two state variables (α, β) describing each
 nonlinear bus..... $2m$

$$2n(1+h)+3m$$

where m is the number of nonlinear buses and h is number of harmonics considered (excluding fundamental).

The equations in harmonic power flow are

Real and reactive power at all but the swing bus.....	$2(n-1)$
Fundamental voltage magnitude and angle at the swing bus.....	2
Fundamental real and imaginary current balance at each nonlinear bus.....	$2m$
Harmonic real and imaginary current balance at each bus.....	$2nh$
Apparent voltampere balance at each nonlinear bus.....	m

$$2n(1+h)+3$$

4.2.1 Effects of Harmonics in Power Studies

Some of the major known effects are summarised below

1. Capacitor bank overloading.
2. Interference with ripple control and power line carrier systems.
3. Harmonic currents in induction and synchronous machines cause additional losses and heating in these machines [14].
4. Overvoltages and excessive currents on the system due to resonance of harmonics on network.
5. Dielectric instability of insulated cables.
6. Interference with telecommunication systems.
7. Errors in induction kilowatt hour meters.
8. Signal interference and relay malfunction in transmission and distribution system.
9. Generator failure due to excitation control problem.

The sources of harmonics may be divided into two categories

1. Established and Known sources
 2. New and Future sources
1. Established and Known Sources
 - a. Tooth ripple in A/C machines.
 - b. Variation of air gap reluctance.

- c. Load effects.
- d. Non-sinusoidal distribution of flux.
- e. Limited transformer current harmonics.
- f. Nonlinear networks.

2. New and Future Sources

- a. Energy conservation measures.
- b. Motor control devices.
- c. High voltage direct current transmission.
- d. Interconnection of wind and solar power converters.
- e. The use of modern arc welders.
- f. Potential development and wide use of electric vehicles.
- g. Potential use of direct energy conversion devices.

Analysis of Harmonics

Networks containing nonlinear current parameters will carry non-sinusoidal currents, even when the applied voltage is a pure sine wave. Devices such as rectifiers, saturated transformers and transmission networks are typical of the causes circuit nonlinearity. This in turn causes distorted wave forms. Once the wave shape of both current and voltage are determined the harmonic contents can be analysed.

4.3 Recalculating line elements at different harmonic frequencies (Program HARMONIC).

4.3.1 Description:

Program HARMONIC calculates the line parameters (G and B) at the harmonic frequency desired. The elements in Y matrix change with harmonic order since they depend upon R and X. Reactance changes with harmonic order as it is frequency dependent. The skin resistance effect has been neglected in the programming. Hence we recalculate the concerned values by dividing them by the order of harmonic and then reformulating the admittance matrix thereby achieving our goal.

The flowchart of HARMONIC is shown in figure 4.1.

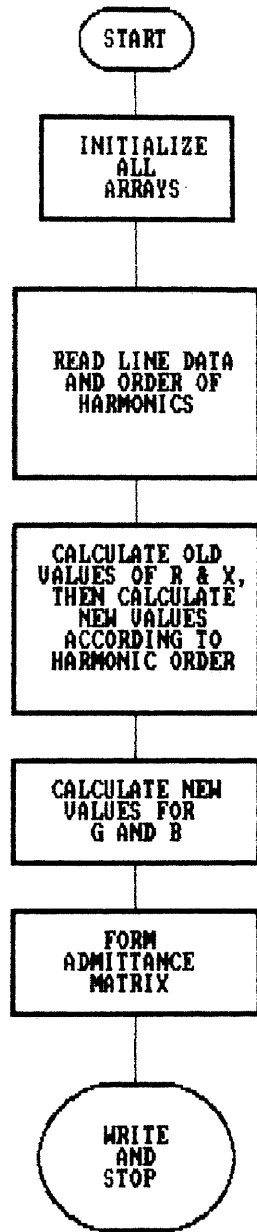


FIGURE 4.1 PROGRAM HARMONIC

4.3.2 List of the program HARMONIC

```

*****
*                               PROGRAM HARMONIC                               *
*****

C      THIS PROGRAM IS USED TO EVALUATE THE EFFECT OF
C      HARMONICS ON CIRCUIT ELEMENTS AND HENCE
C      CALCULATE THE CHANGED VALUES DUE TO HARMONICS.
C      DESCRIPTION OF PARAMETERS
C
C      N      ORDER OF HARMONIC
C      ROLD   R BEFORE INTRODUCTION OF HARMONIC
C      XOLD   X BEFORE INTRODUCTION OF HARMONIC
C      RNEW   R AFTER INTRODUCTION OF HARMONIC
C      XNEW   X AFTER INTRODUCTION OF HARMONIC
C      NL     NUMBER OF LINES
C      Y      ADMITTANCE MATRIX
C      DIMY   DIMENSION OF Y MATRIX
C*****

      DIMENSION Y(22,22),ROLD(34),XNEW(34)
      1G(34),B(34),K(34),M(34),XOLD(34),RNEW(34)
      COMPLEX Y
      REAL G,B,ROLD,XNEW,DIMY,RNEW,XOLD
      INTEGER K,M,NL,N

C*****
C      THIS IS CONSIDERED TO BE THIRD ORDER HARMONIC
C      DATA N,NL,DIMY/3,34,22/

C*****INIT INITIALISES ALL THE ARRAYS*****
      CALL INIT(ROLD,XNEW,Y,G,B,NL,DIMY,RNEW,XOLD)

C*****
C      READDATA READS INPUT DATA WHICH IS THE BUS
C      ADMITTANCE MATRIX
C*****
      CALL READDATA(NL,I,J,G,B,Y,DIMY,K,M)

C*****
C      XYCAL CALCULATES OLD VALUES OF R AND X
C      AND THEN FINDS THE NEW VALUES OF R AND X
C      CONSIDERING THE ORDER OF HARMONIC
C*****
      CALL XYCAL(ROLD,XNEW,G,B,N,NL,RNEW,XOLD)

```

```

C*****
C      GBCAL FINDS THE NEW VALUES FOR CIRCUIT ELEMENTS
C      FROM CHANGED VALUES OF R AND X
C*****
      CALL GBCAL(XNEW,G,B,NL,RNEW)

```

```

C*****
C      FORMYAG FORMS THE NEW BUS ADMITTANCE MATRIX
C*****

      CALL FORMYAG(G,B,K,M,NL,Y,DIMY)
      STOP
      END

```

```

*****
      SUBROUTINE READDATA(NL,I,J,G,B,Y,DIMY,K,M)
      COMPLEX Y(22,22)
      INTEGER I,J,DIMY,NL,K(34),M(34)
      REAL G(34),B(34),GEE,BEE
      OPEN(UNIT=15,FILE='LAST.IN',STATUS='OLD')
      DO 10 P=1,NL
      READ(15,*)I,J,GEE,BEE
      K(P)=I
      M(P)=J
      G(P)=GEE
      B(P)=BEE
10     CONTINUE
      RETURN
      END

```

```

*****
      SUBROUTINE XYCAL(ROLD,XNEW,G,B,N,NL,RNEW,XOLD)
      REAL XNEW(34),ROLD(34),G(34)
      REAL B(34),XOLD(34),RNEW(34)
      INTEGER N,NL
      DO 34 I=1,NL
      XOLD(I)=B(I)/(G(I)**2+B(I)**2)
      ROLD(I)=-G(I)/(G(I)**2+B(I)**2)
      XNEW(I)=N*B(I)/(G(I)**2+B(I)**2)
      RNEW(I)=ROLD(I)
34     CONTINUE
      RETURN
      END

```

```

*****
      SUBROUTINE GBCAL(XNEW,G,B,NL,RNEW)
      REAL ROLD(34),XNEW(34),G(34)
      REAL B(34),XOLD(34),RNEW(34)
      INTEGER NL
      DO 30 I=1,NL

```

```

      G(I)=-RNEW(I)/(RNEW(I)**2+XNEW(I)**2)
      B(I)=XNEW(I)/(RNEW(I)**2+XNEW(I)**2)
30    CONTINUE
      RETURN
      END
*****

      SUBROUTINE FORMYAG(G,B,K,M,NL,Y,DIMY)
      COMPLEX Y(22,22)
      REAL G(34),B(34)
      INTEGER K(34),M(34),NL,DIMY
      DO 40 P=1,NL
      WRITE(6,*)K(P),M(P),G(P),B(P)
40    CONTINUE
      RETURN
      END
*****

      SUBROUTINE INIT(ROLD,XNEW,Y,G,B,
                    NL,DIMY,RNEW,XOLD)

      COMPLEX Y(22,22)
      REAL ROLD(34),XNEW(34),G(34)
      REAL B(34),RNEW(34),XOLD(34)
      INTEGER NL,DIMY
      DO 45 I=1,NL
      G(I)=0.0
      B(I)=0.0
      ROLD(I)=0.0
      XNEW(I)=0.0
      RNEW(I)=0.0
      XOLD(I)=0.0
45    CONTINUE
      RETURN
      END
*****

```

4.4 Evaluation of HARMFLO

As mentioned earlier, HARMFLO is a computer program used to calculate the magnitude and propagation of harmonics in balanced power systems. This program is useful for predicting or diagnosing design or operating problems, such as resonance detection and prevention, insulation coordination and protection, and HVDC filter design operation. HARMFLO requires to be operated under a special environment.

4.4.1 Special environment instructions

1. Test Computers

IBM PC compatible computer with one disk drive, one hard Disk and 640 kbytes memory (hard disk used).

2. Operating Systems

Microsoft DOS 3.1 (IBM PC compatible)

3. Programming Language

IBM Professional Fortran

4. Installation Dependant Requirements

PC must be equipped with the INTEL 8087 coprocessor chip in order to execute HARMFLO.

5. Other Recommended Accessories

- a. IBM PC with 640 RAM (512k minimum)
- b. Lahey Fortran Compiler Version 2.11

4.4.2 System Size

In assessing the size of the system to be studied the total number of buses, the total number of nonlinear buses and number of lines have to be considered. In addition, two additional parameters are needed, the number of separate harmonics to be analysed and the number of types of nonlinear devices. The table 4.1 shows the parameters as they appear in HARMFLO Version 4.0.

4.4.3 Time and Memory requirements

It is very difficult to estimate the time and memory requirements of a harmonic power flow study because these parameters are highly system dependant.

4.4.4 Data preparation

Because program HARMFLO is generic in the sense that power systems of all voltage levels may be studied, most input data are perunitized.

Input Data Types ("codes"): The input data file is organised into types denoted "input data codes".

There are nine input data codes. They are preceded by header cards which contain the code number, right justified, in 14 format. Thus the input data code header occurs in columns 1 through 4 with the integer code type indicated.

The table below shows the function of the input data code types.

The input data should be entered in the following order:

Control code 3 - Bus data

Control code 4 - Line and transformer data

Control codes 11,12, 13 - Additional bus data for
nonlinear loads/sources

Control code 5 - Request solution and read options

Control code 15 - End of data.

Control codes 1 and 2 can be entered at any time. Note that the code 3 header card is also used to select the bus ordering technique.

Input control code 3 bus data : The control 3 deal with type, subtypes and other aspects of bus data as shown in tables 4.2 through 4.5.

Input control code 4 line data : Control code 4 data are used to specify information concerning transmission lines, transformers and impedance ties to neutral. Code 4 data must follow the code 3 data and precede the code 5 (solution request) card. The table 4.6 gives the summary of the code 4 information.

Input control code 5 solution options : Control code 5 is used to initiate the power flow solution and select solution options. Immediately following the code 5 card, a solution card must read. There are 16 available options which may be specified regarding the solution [8].

Input control code 11 nonlinear type 1 bus data (Converter without dc voltage source) : Control code 11 data specifies additional information for buses which are designated as passive six or twelve pulse line commutated converters in the code 3 data. This is shown in table 4.7.

Input control code 12 nonlinear type 2 bus data

(Converter with dc voltage source) : Control code 12 data are very similar to code 11 data. Table 4.7 applies. The specified and estimated parameters according to the device types are given in table 4.8.

Input control code 13 nonlinear type 3 bus data

(Nonlinear resistor) : Control code 13 data specify additional information for buses which are designated as nonlinear resistors in the code 3 data.

This is shown in table 4.9.

Input control code 15 end of data

Control code 15 is used to signify the end of the harmonic power flow study data. This is the last entry in the data input.

Quantity	Remarks	Maximum	Program variable
Number of buses	Compressed	50	MAXNB
Number of lines	Includes shunt elements	200	MAXLIN
Number of harmonics	-	7	MAXHAR
Diagonal entries of Jacobian matrix	-	706	MAXJAC
Highest user bus number	Not compressed	999	MAXBNC
Types of nonlinear devices	-	3	-
Exponent of voltage in nonlinear resistor	Maximum	7	MAXEXP
Maximum harmonic multiple	-	37	MAXHI
Nonzero entries per row in Jacobian matrix	Maximum	9	MAXDIA
Nonzero entries in Jacobian matrix	Off diagonal	5000	MAXSPR
Number of entries in sparsity programming vector	Maximum	15	MAXNCU

Table 4.1 System Size for HARMFLO Version 4.0

Column(s)	Format	Information
1-4	I4	Bus number
5-16	3A4	Bus name
17	I1	Bus type
18	I1	Bus sub-type
19-26	F8.0	Active power generation(%)
7-34	F8.0	Reactive voltampere gen(%)
35-42	F8.0	Apparent voltamperes (%)
43-50	F8.0	Active power load (%)
51-58	F8.0	Reactive voltampere load (%)
59-66	F8.1	Controlled bus voltage (%)
67-74	F8.0	Shunt reac/cap load (%)

Table 4.2 HARMFLO Control Code 3 Information

Type	Description
0	Fundamental
1	Converter without dc voltage source
2	Converter with dc voltage source
3	Nonlinear Resistor

Table 4.3 Allowed Bus Types (HARMFLO)

Sub-Type	Description
1	Fundamental swing bus (SB, only one allowed)
2	Fundamental voltage controlled bus (PV)
3	Other fundamental bus (PQ)
4	Nonlinear bus with real power and apparent voltamperes specified (PS)
5	Nonlinear bus with real power and delay angle specified ($P\alpha$)
6	Nonlinear bus with dc voltage and delay angle specified ($E\alpha$)
7	Nonlinear bus with real power only specified (P)
8	Induction motor load (IM)

Table 4.4 Allowed Bus Sub-Types (HARMFLO)

Type	Sub-Type
Fundamental	SB, PV, PQ, IM
Converter without dc voltage source	PQ, PS, $P\alpha$
Converter with dc voltage source	PQ, PS, $P\alpha$, $E\alpha$
Nonlinear Resistor	P

Table 4.5 Allowed combinations of Bus Types and Sub-Types (HARMFLO)

Columns	Format	Information
1-4	I4	From bus number
5-8	I4	To bus number
9-16	F8.2	R(+) (%)
17-24	F8.2	X(+) (% @ 60 Hz)
25-32	F8.2	Pos. sequence line charging (%)
33-40	F8.2	R(0) (%)
41-48	F8.2	X(0) (% @ 60 Hz)
49-56	F8.2	Zero sequence line charging (%)
57-62	F6.3	Transformer tap
63-68	F6.1	Line length (miles)
69-76	F8.2	Base impedance
77	I1	Transformer connection type
78	I1	Shunt impedance connection type
79	I1	Designator causes shunt impedance element to be effective for harmonics only
80	I1	Number of conductors per phase

Table 4.6 HARMFLO Control Code 4 Information

Columns	Format	Information
1-4	I4	Bus number
5-12	F8.2	Converter resistance R (%)
13-20	F8.2	DC circuit reactance F (% @ 60 Hz)
21-28	F8.2	DC voltage E(%)
29-36	F8.2	Delay angle α (degrees)
37-44	F8.2	Transfor. resis. (%)
45-52	F8.2	Transfor. reac. L (% @ 60 Hz)
53	I1	Transformer connection type
54	I1	Bipolar/monopolar configuration
55	I	DC return circuit type
56-63	F8.2	Transfor phase shift

Table 4.7 HARMFLO Control Codes 11 and 12 Information

Converter without dc voltage source			
Nonlinear Device	Knowns	Unknowns	Data Type
Type PQ	$P, Q, F, L, E=0$	R, α, μ	Code 11
Type PS	$P, S, F, L, E=0$	R, Q, α, μ	Code 11
Type P α	$P, \alpha, F, L, E=0$	R, Q, μ	Code 11
Converter with dc voltage source			
Nonlinear Device	Knowns	Unknowns	Data Type
Type PQ	P, Q, F, L, R	E, α, μ	Code 12
Type PS	P, S, F, L, R	E, Q, α, μ	Code 12
Type P α	P, α, F, L, R	E, Q, μ	Code 12
Active converter			
Nonlinear Device	Knowns	Unknowns	Data Type
Type E α	E, α, F, L, R	P, Q, μ	Code 12
Nonlinear resistor			
Nonlinear Device	Knowns	Unknowns	Data Type
Type P	$P, b_k, k=1, 3, 5$	B, Q	Code 13

Table 4.8 Specified and Estimated Parameters
According to Device Type

Column(s)	Format	Information
1-4	I4	bus number
5-12	F8.0	b ₁
13-20	F8.0	b ₃
21-28	F8.0	b ₅
29-36	F8.0	b ₇
37	I1	resistor connection

Table 4.9 Control Code 13 Information

CHAPTER FIVE

RESULTS AND OBSERVATIONS

While the first section of this chapter deals with the power systems used as test input, the subsequent sections list the results and observations from the following conventional and harmonic load flow programs.

JEET This program exploits sparsity and utilizes optimal ordering.

HARMFLO This is a commercial software program developed by Energy Power Research Institute, California.

NOOPTIMAL This program exploits sparsity but does not utilize optimal ordering.

DIFFLOAD This program is used for testing JEET under different loading conditions.

HARMONIC This program is used to calculate the values of line elements for different harmonic frequencies.

5.1 The Test Systems

Three power systems in all have been used to test the programs mentioned above. They have been categorized as shown in the table 5.1.

5.1.1 14 Bus system

This system is a well known IEEE test system [15]. It has 14 nodes and 20 lines and is shown in figure 5.1. The input data file structure for programs JEET, NOOPTIMAL, DIFFLOAD and HARMONIC is as shown in the table 5.2. The names of the files which contain this input data is different for different programs. The names would be found in the "OPEN" statement at the beginning of each program. Same 14 bus system used in testing of JEET has been used as input to HARMFLO to simplify comparison.

Input to HARMFLO is organised into types denoted by input data codes. Codes 1 and 2 are the page heading and comment card respectively.

Control code 3 is the bus data. The buses have been numbered so that the first bus is the slack bus which is a voltage specified bus, buses 2 through 10 are load buses for which P(load) and Q(load) are specified and

BUS SYSTEM	PROGRAMS TESTED
14	JEET NOOPTIMAL DIFFLOAD HARMFLO HARMONIC
118	JEET NOOPTIMAL DIFFLOAD
15	HARMFLO

Table 5.1 Input classification

C THESE ARE THE LINE ADMITTANCES

From bus	To bus	G	B
1	1	6.02503	19.44705
1	5	-1.02590	-4.23498
1	11	-4.99913	-15.26309
2	2	3.83591	8.49701
2	10	-1.88088	-4.40294
2	13	-1.95503	-4.09407
3	3	4.01499	5.42794
3	6	-2.48903	-2.25198
3	13	-1.52597	-3.17596
4	4	10.51299	38.66307
4	5	-6.84098	-21.57854
4	7	0.00000	-4.88951
4	9	0.00000	-1.86900
4	11	-1.68603	-5.11584
4	12	-1.98598	-5.06882
5	5	9.56801	34.92735
.	.	.	.
.	.	.	.
14	14	0.00000	5.67698

C THIS IS SPECIFIED POWER

Bus	SP
1,	(0.0,0.0)
2,	(0.035,0.018)
3,	(.061,.016)
4,	(.478,-0.039)
.	.
10,	(0.09,0.058)
11,	(-0.183,-0.2434)
12,	(0.942,-0.03988)
13,	(.112,-0.46956)
14,	(0.0,-0.28348)

C THESE ARE THE SPECIFIED VOLTAGE FOR GENERATOR BUSES

1.045,1.010,1.070,1.090

Table 5.2 Sample input file format (JEET,
DIFFLOAD, NOOPTIMAL - 14 bus system)

buses 11 through 14 are generator buses for which active power generation and specified voltage are known.

Control code 4 is the line data which is used to specify information concerning transmission lines that go from bus to bus, their resistances, reactances and line chargings. The numerals 9999 indicate the end of specific kind of data.

Control code 5 gives the choice on output.

Unless specified that an entry in the particular field is a must, the program assumes default values for those fields.

The program reads the input from the file FORT5. The structure of the input file is shown in the table 5.3. The statements starting and ending with * are placed as comments. They should be excluded from the actual input file. Also different control codes have been separated by double line spacing for clarity and each entry for particular field has to be in specific column as mentioned in chapter four.

```

INPUT FOR HARMFLO (LINEAR NODE)

0001 * control code 1 page heading *
DATA FOR FOURTEEN BUS SYSTEM

0002 * control code 2 comment card *
COMPARISON OF SOLUTION

0003 * control code 3 bus data *

*bus no.  bus  Pgen  Qgen          Pload  Qload  Vsp *
          type
-----
0001      01                106.0
0002      03                3.5    1.8
0003      03                6.1    1.6
.         .
.         .
0010      03                9.0    5.8
0013      0211.2          -46.956
0014      020.0           -28.348
-----

9999 * end of type 3 data *

0004 * control code 4 line data *

*fr bus to  R    X    line charging *
-----
000100111.938  5.917  2.64
000100055.403  22.304  2.46
001100124.699  19.797  2.19
000900103.181  8.450   0.0
0009000812.711 27.038  0.0
001000028.205  19.207  0.0
0003000622.092 19.988  0.0
0006000817.093 34.802  0.0
-----

9999 * end of type 4 data *

0005 * control code 5 output data *
12

0015 * end of file *
#eor
#eof

```

Table 5.3 Sample input file format (HARMFLO - 14 bus system)

5.1.2 118 bus system

This system has 118 nodes and 179 lines. It has been used to test conventional load flow programs.

Significant results were obtained from NOOPTIMAL in terms of memory consumption and execution time using 118 bus system than 14 bus system. The input data file structure for 118 bus system is the same as for 14 bus system shown in previous section.

5.1.3 15 bus system

By adding a nonlinear node to the IEEE 14 bus test system described above, HARMFLO has been used to find the effect of harmonics generated on the network. This nonlinear bus has been added to the bus 6 of the 14 bus system. The bus has been designated as a passive six pulse line commutated converter in control code 3 [16]. The converter is without a dc voltage source and is taken to be a nonlinear bus with real power and apparent voltamperes specified ($P\alpha$ type). This nonlinear type bus data has been included in control code 11. The known parameters for such a bus are active power load, delay angle, DC circuit reactance, transformer reactance and DC voltage which is equal to zero.

Since the nonlinear bus is connected to bus 6, the active power load of bus 6 is the active power load of the nonlinear bus. The delay angle and the DC circuit reactance are assumed to be zero. Since the transformer reactance is to be in the 5% - 20% range on the coverter power base it is taken to be 100%. If it is too far outside this range, difficulties would occur in solving for the commutation angle. The structure of the input file is the same as shown in table 5.3 except for the addition of control code 11 data. It is shown in table 5.4. For more detailed explanation on data input please refer to section 4.4.4 in chapter 4.

5.2 Interpretation of the output

The output file for JEET, NOOPTIMAL and DIFFLOAD has the structure shown in table 5.5. The output is the array CON. The output begins with the statement commenting on the status of the program on termination which is also shown in table 5.5.

The output from HARMFLO for system with linear nodes like the 14 bus system discussed above consists of fundamental load flow iteration summary as also the bus voltage summary. Since the complete listing of HARMFLO output is too exhuastive a partial listing is shown in table 5.6.

INPUT FOR HARMFLO (NONLINEAR NODE)

0001
DATA FOR FOURTEEN BUS SYSTEM

0002
COMPARISON OF SOLUTION

0003				
0001	01			106.0
0002	03	3.5	1.8	
.	.			
.	.			
0010	03	9.0	5.8	
0011	02-18.3			104.5
.	.			
0014	020.0			109.0
0015	15	13.5		
9999				

0004				
000100111.938	5.917	2.64		
000100055.403	22.304	2.46		
001100124.699	19.797	2.19		
001100045.811	17.632	1.87		
001100055.695	17.388	1.70		
001200046.701	17.103	1.73		
000400051.335	4.211	0.64		
000400070.0	20.912	0.0		
.	.			
00060015	0.01			
9999				

0011 * control code 11 data nonlinear bus *

0015	0.0	0		100.0
9999				

0005
0 0

0015
#eor
#eof

Table 5.4 Sample input file format (HARMFLO -
15 bus system)

CONVERGED IN 3 ITERATION BY METHOD FJM = 0 MFJM = 0
NR = 1

* These are the elements of CON. Since the array CON contains the power mismatches most are the elements are zero on convergence. *

-2.3266	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	
0.2066	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000
0.0001	0.0000			
-.0003	-.0001			

Table 5.5 Sample output (JEET, NOOPTIMAL and DIFFLOAD)

DATA FOR FOURTEEN BUS SYSTEM

HARMFLO VER 4.0 PAGE = 5

FUNDAMENTAL LOADFLOW ITERATION SUMMARY

ABSOLUTE ACTIVE

ABSOLUTE REACTIVE

POWER MISMATCH

POWER MISMATCH

WORST ITERATION	AVERAGE	WORST	WORST	AVERAGE	WORST	
BUS			BUS			
5	0	21.97	96.21	12	30.49	77.36
4	1	1.04	3.52	12	2.22	6.34
4	2	0.01	0.02	4	0.01	0.03
4	3	0.00	0.00	4	0.00	0.00

Table 5.6 Sample output (HARMFLO)

The output of HARMFLO for system with nonlinear nodes is similar to the one discussed above. Initially, the fundamental frequency load flow study is performed for the purpose of initializing the harmonic load flow study. The results of this initialization are not printed unless the specific option is chosen. Having initialized the harmonic study, the harmonic study begins. As this study progresses, iteration-by-iteration results are printed. After having reached convergence, a line flow and bus voltage summary is printed.

The highlights of the line flow output are

- * Line flow format is very similar to the output of a conventional power flow study.
- * Each harmonic is printed separately with the harmonic number, frequency and sequence indicated in the header. The fundamental frequency is also printed in this way.
- * Each line load, at a given frequency, is indexed with a "FROM" bus and a "TO" bus.
- * The "FROM" bus voltage magnitude and angle is printed.
- * The "FROM" and "TO" bus numbers are printed.
- * A tie to ground (neutral) is indicated by the

designator "NEUT" in the "TO" bus position. If this tie is an impedance tie to neutral, the name "Z SHUNT" is used. If a nonlinear device is used, "NONLIN DEV" is indicated.

- * Active (P) and reactive (Q) flows are printed.
- * If a fixed tap transformer is present, the tap, position is indicated.
- * Beyond the printing of each individual harmonic flow, a line flow summary is printed after the line flows for the highest harmonic frequency. This summary contains the line flows at the fundamental frequency, rms current, peak current, THD and IT for the current.
- * P, Q, D, and S are printed for each line in the line flow summary. These are total values.

The highlights of the bus voltage output are

- * Individual bus voltage (magnitude and angle) are printed for each harmonic frequency in the line flow output.
- * A bus voltage summary follows the line flow summary. The bus voltage summary consists of the bus voltage at the fundamental frequency, rms value and peak value.
- * Total harmonic distortion is printed.

- * Total rms positive, negative, and zero sequence content is printed.
- * Telephone influence factor (TIF) is printed for each bus [17].

The output from HARMONIC consists of updated elements of line admittances. The system frequency changes with the order of the harmonic hence the frequency dependent variables have to be updated. HARMONIC reads in the admittances, updates them in accordance with the harmonics and outputs them in the same format. It is as shown in table 5.7.

5.3 Observations and Discussion

5.3.1. JEET

By changing the initial data statements JEET can be used to study systems other than the 14 bus system. Special consideration is needed to decide the value of SWN which is the length of arrays ITAG, LNXT, RE and CE and since it is appointed by the number of fill-in terms it cannot be determined in advance.

OUTPUT FROM HARMONIC

From bus	To bus	G	B
1	1	0.7259633	7.029597
1	5	-0.1198963	-1.484818
1	11	-0.6078016	-5.567128
2	2	0.5017132	3.334069
2	10	-0.2422132	-1.700986
2	13	-0.2601681	-1.634471
3	3	0.6506402	2.638838
3	6	-0.5409760	-1.468364
3	13	-0.2034751	-1.270462
4	4	1.244254	13.72779
4	5	-0.8272659	-7.828349

Table 5.7 Sample output (HARMONIC)

The observations while testing the 14 bus system on JEET are as follows:

Number of unknowns	22
Number of non-zero off diagonal terms	
Before reduction	104
After reduction	124
Ratio of non-zero off diagonal terms	
After/before reduction	1.19
Maximum dimension of arrays	125

5.3.2 HARMFLO

Both the programs HARMFLO and JEET converged in three iterations while being tested on the 14 bus system for conventional load flow. The calculated voltages at the load buses at the end of three iterations are as shown in table 5.8.

HARMFLO is a significant improvement over other methods [11][17] used for harmonic analysis because loads are stated by power demand levels. Thus the harmonic levels need not be known before the study. Moreover, the solution does not depend upon assumed linear superposition. Input to HARMFLO is similar to existing power flows.

HARMFLO in conclusion has met with the needs of the initial aim of the thesis that was to generate a program for Harmonic Load Flow Analysis.

5.3.3 NOOPTIMAL

Though both JEET and NOOPTIMAL converged in three and five iterations for 14 bus and 118 bus system respectively, entries in tables 5.9 and 5.10 indicate that the memory requirement and computation time in NOOPTIMAL increased significantly for systems with greater number of buses since the number of fill-in terms increased in a greater ratio.

The larger the system used for input, more significant achievement in the computation time and memory can be achieved by resorting to optimal ordering as can be noticed from the tables.

5.3.4 DIFFLOAD

This program takes input from two files. It reads line admittances from the file shown in table 5.2. The second file consists of only the specified voltages of the generator buses. Since integer variable SEED in the data statement is used to generate different values of loading conditions, SEED needs to be changed to change the

loading conditions. Also initial data statements could be changed to test a system other than the 14 bus system. DIFFLOAD writes the output on the Screen.

5.3.5 HARMONIC:

This program reads the admittance matrix and operates on the line elements considering the order of the harmonic. The program has been initialized with the harmonic of order 3.

The data statement variable N needs to be changed considering other order harmonics.

Bus #	Bus voltages (HARMFLO)	Bus voltages (JEET)	Difference
2	104.42	104.07	0.35
3	105.30	105.22	0.08
4	102.63	101.93	0.70
5	103.5	102.84	0.66
6	104.59	104.47	0.12
7	104.43	104.41	0.02
8	101.6	101.10	0.04
9	102.6	101.77	0.83
10	102.63	101.93	0.70

Table 5.8 Comparison of results (HARMFLO & JEET)

PARAMETER	JEET	NOOPTIMAL
Fill-in terms	125	250
Execution time	55 secs.	58 secs.

Table 5.9 Comparison of results (JEET & NOOPTIMAL
14 bus system)

PARAMETER	JEET	NOOPTIMAL
Fill-in terms	1500	6500
Execution time	4.19 mins.	9.94 mins.

Table 5.10 Comparison of results (JEET & NOOPTIMAL
118 bus system)

APPENDIX A

STORAGE SCHEME FOR AN EXAMPLE 6x6 MATRIX

	1	2	3	4	5	6
1	x	x		x		
2	x	x	x		x	
3		x	x			x
4	x			x	x	
5		x		x	x	x
6			x		x	x

x = non-zero term

Asymmetrical matrix A:

For the 6x6 matrix shown above the various arrays are

	LCOL	NOZE	DE
1	1	3	a ₁₁
2	3	4	a ₂₂
3	6	3	a ₃₃
4	8	3	a ₄₄
5	10	4	a ₅₅
6	13	3	a ₆₆
7	-	-	-
8	-	-	-

	ITAG	LNXT	CE	RE
1	2	2	a21	a12
2	4	0	a41	a14
3	1	4	a12	a21
4	3	5	a32	a23
5	5	0	a52	a25
6	2	7	a23	a32
7	6	0	a63	a36
8	1	9	a14	a41
9	5	0	a54	a45
10	2	11	a25	a52
11	4	12	a45	a54
12	6	0	a65	a56
13	3	14	a36	a63
14	5	0	a56	a65
15	-	16	0	0
16	-	17	0	0
17	-	18	0	0
18	-	19	0	0
19	-	20	0	0
20	-	21	0	0
21	-	22	0	0
22	-	23	0	0
23	-	24	0	0
24	-	0	0	0

APPENDIX B

DESCRIPTION OF PARAMETERS

Integer Variables

I	row index of terms in processed column K or running index
IP	row index of terms in pivotal column
J	number of reduction step
K	index of column under consideration or running index
KP	pivotal index related to reduction step J
L	location of terms in processed column K
LA	location of preceding term in processed column K
LF	indicator for next vacant location
LI	location of terms in pivotal column (inner loop)
LK	location of terms in pivotal column (outer loop)
LN	location of new added fill-in term
LP	location of intermediately stored terms of pivotal column
M	intermediate integer variable
MIN	minimum number of non-zero terms
N	number of unknowns, order of the matrix

Real Variables

CF	multiplier for columns
D	diagonal (pivotal) term
RF	multiplier for rows
SUM	sum of products

Integer arrays

ITAG	row index of elements stored in CE
LCOL	starting position of columns
LNXT	location of next term
NOZE	number of non-zero terms
NSEQ	sequence of pivotal indices

Real Arrays

CE	columnwise stored matrix term
DE	diagonal term
RE	rowwise stored matrix term
V	vector of unknowns, solution vector

APPENDIX C

NOMENCLATURE

a,d	Real and imaginary injection currents.
B_1	Integral limit.
c	Fourier current coefficient.
C	Converter solution constant. Capacitance.
$C^{(1)}$	Solution term for converter current.
e	Error.
$E^{(1)}$	Solution term for converter current.
$F^{(1)}$	Solution term for converter current.
f,x	Nonlinear device real and imaginary currents.
G	Partial derivatives of nonlinear device currents.
h	Highest harmonic of interest.
H	Matrix of nonlinear device current partial derivatives.
I	Current.
J	Jacobian matrix.
k	Harmonic order.
m	Number of nonlinear buses.
n	Number of buses.
N	Number of linear buses.
P	Active power.
Q	Reactive power.
S	Steady state.
t	Time.

T	Matrix of partial derivatives of harmonic injection currents.
T_1	Integral limit.
TG	Sum of T and G matrices.
α	Delay angle.
β	Nonlinear device control variable.
τ	Nonlinear device control variable.
δ	Voltage phase angle.
Δ	Newton-Raphson solution update. Increment.
ΔW	Power mismatch vector.
θ	Phase angle.
μ	Commutation angle.
Σ_k	Summation from k=1 to h.
ϕ	Fourier coefficient for converter AC voltage angle.
ω_0	Fundamental frequency.

APPENDIX D

EXAMPLE TO CALCULATE TG & H MATRICES

Consider a 3 bus power system as described below.

Bus 1 - slack bus

Bus 2 - linear bus

Bus 3 - nonlinear bus

Let the harmonic order (including the fundamental) to be considered be 1,3 and 5.

The possible TG(k,j) matrices for k = j are

$$TG(1,1) = T(1,1) + G(1,1) \quad (D.1)$$

$$TG(3,3) = T(3,3) + G(3,3) \quad (D.2)$$

$$TG(5,5) = T(5,5) + G(5,5) \quad (D.3)$$

The possible TG(k,j) matrices for k ≠ j are

$$TG(3,1) = G(3,1) \quad (D.4)$$

$$TG(3,5) = G(3,5) \quad (D.5)$$

$$TG(5,1) = G(5,1) \quad (D.6)$$

$$TG(5,3) = G(5,3) \quad (D.7)$$

The first superscript k refers to the harmonic order of the injection or load current under consideration and the second superscript j refers to the harmonic order of the voltage and phase angle under consideration.

Let us consider the case for $k = 1$ and $j = 1$. The $TG(1,1)$ matrix is the sum of two matrices as shown in (D.1). The $T(1,1)$ and $G(1,1)$ would have the dimension $(2m, 2n)$ where

m is number of nonlinear buses = 1 in our case

n is total number of buses = 3 in our case

For fundamental frequency ($k = 1$) we consider only the fundamental frequency real and imaginary currents at each nonlinear bus. In our case we only have bus 3 as nonlinear bus thus we consider only bus 3 real and imaginary currents for calculating $T(1,1)$ and $G(1,1)$.

Hence $T(1,1)$ is given by

$$\begin{bmatrix} A & B & C & D & E & F \\ A1 & B1 & C1 & D1 & E1 & F1 \end{bmatrix}$$

which is a $2 * 6$ matrix, where

$$A = \begin{bmatrix} 1 \\ 1/V_1 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ \delta a_3/\delta \theta_1 \end{bmatrix} \quad A1 = \begin{bmatrix} 1 \\ 1/V_1 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ \delta d_3/\delta \theta_1 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 & 1 \\ \delta a_3/\delta V_1 \end{bmatrix} \quad B1 = \begin{bmatrix} 1 & 1 \\ \delta d_3/\delta V_1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 \\ 1/V_2 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ \delta a_3/\delta \theta_2 \end{bmatrix} \quad C1 = \begin{bmatrix} 1 \\ 1/V_2 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ \delta d_3/\delta \theta_2 \end{bmatrix}$$

$$D = \begin{bmatrix} 1 & 1 \\ \delta a_3/\delta V_2 \end{bmatrix} \quad D1 = \begin{bmatrix} 1 & 1 \\ \delta d_3/\delta V_2 \end{bmatrix}$$

$$E = \begin{bmatrix} 1 \\ 1/V_3 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ \delta a_3/\delta \theta_3 \end{bmatrix} \quad E1 = \begin{bmatrix} 1 \\ 1/V_3 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ \delta d_3/\delta \theta_3 \end{bmatrix}$$

$$F = \begin{bmatrix} 1 & 1 \\ \delta a_3/\delta V_3 \end{bmatrix} \quad F1 = \begin{bmatrix} 1 & 1 \\ \delta d_3/\delta V_3 \end{bmatrix}$$

The matrix $G^{(1,1)}$ is given by

$$\begin{bmatrix} G & H & I & J & K & L \\ G1 & H1 & I1 & J1 & K1 & L1 \end{bmatrix}$$

which is a $2 * 6$ matrix

where

$$G = \begin{bmatrix} 1 \\ 1/V_1 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ \delta f_3/\delta\theta_1 \end{bmatrix} \quad G1 = \begin{bmatrix} 1 \\ 1/V_1 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ \delta x_3/\delta\theta_1 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 1 \\ \delta f_3/\delta V_1 \end{bmatrix} \quad H1 = \begin{bmatrix} 1 & 1 \\ \delta x_3/\delta V_1 \end{bmatrix}$$

$$I = \begin{bmatrix} 1 \\ 1/V_2 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ \delta f_3/\delta\theta_2 \end{bmatrix} \quad I1 = \begin{bmatrix} 1 \\ 1/V_2 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ \delta x_3/\delta\theta_2 \end{bmatrix}$$

$$J = \begin{bmatrix} 1 & 1 \\ \delta f_3/\delta V_2 \end{bmatrix} \quad J1 = \begin{bmatrix} 1 & 1 \\ \delta x_3/\delta V_2 \end{bmatrix}$$

$$K = \begin{bmatrix} 1 \\ 1/V_3 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ \delta f_3/\delta\theta_3 \end{bmatrix} \quad K1 = \begin{bmatrix} 1 \\ 1/V_3 \end{bmatrix} * \begin{bmatrix} 1 & 1 \\ \delta x_3/\delta\theta_3 \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & 1 \\ \delta f_3/\delta V_3 \end{bmatrix} \quad L1 = \begin{bmatrix} 1 & 1 \\ \delta x_3/\delta V_3 \end{bmatrix}$$

Now let us consider the case for $k > 1$ and $j > 1$ but $k = j$. The $TG^{(k,j)}$ matrix is the sum of two matrices as shown in equations (D.2-D.3). The $T^{(k,j)}$ and $G^{(k,j)}$ would have the dimension $(2n,2n)$ where

n is total number of buses = 3 in our case

For frequency ($k > 1$ and $j > 1$) we consider only the frequency k real and imaginary currents at each bus. In our case we have totally 3 buses thus we consider real and imaginary currents for frequency k at all 3 buses for calculating $T^{(k,j)}$ and $G^{(k,j)}$.

Hence $T(3,3)$ is given by

$$\begin{bmatrix} M & N & O & P & Q & R \\ M1 & N1 & O1 & P1 & Q1 & R1 \\ S & T & U & V & W & X \\ S1 & T1 & U1 & V1 & W1 & X1 \\ A & B & C & D & E & F \\ A1 & B1 & C1 & D1 & E1 & F1 \end{bmatrix}$$

The dimension of this matrix is $2n * 2n$ which is $6 * 6$ in this case.

where

$$M = \begin{bmatrix} 1/V_1 \end{bmatrix} * \begin{bmatrix} \delta a_1 / \delta \theta_1 \end{bmatrix} \quad M1 = \begin{bmatrix} 1/V_1 \end{bmatrix} * \begin{bmatrix} \delta d_1 / \delta \theta_1 \end{bmatrix}$$

$$N = \begin{bmatrix} \delta a_1 / \delta V_1 \end{bmatrix} \quad N1 = \begin{bmatrix} \delta d_1 / \delta V_1 \end{bmatrix}$$

$$O = \begin{bmatrix} 1/V_2 \end{bmatrix} * \begin{bmatrix} \delta a_1 / \delta \theta_2 \end{bmatrix} \quad O1 = \begin{bmatrix} 1/V_2 \end{bmatrix} * \begin{bmatrix} \delta d_1 / \delta \theta_2 \end{bmatrix}$$

$$P = \begin{bmatrix} \delta a_1 / \delta V_2 \end{bmatrix} \quad P1 = \begin{bmatrix} \delta d_1 / \delta V_2 \end{bmatrix}$$

$$Q = \begin{bmatrix} 1/V_3 \end{bmatrix} * \begin{bmatrix} \delta a_1 / \delta \theta_3 \end{bmatrix} \quad Q1 = \begin{bmatrix} 1/V_3 \end{bmatrix} * \begin{bmatrix} \delta d_1 / \delta \theta_3 \end{bmatrix}$$

$$R = \begin{bmatrix} \delta a_1 / \delta V_3 \end{bmatrix} \quad R1 = \begin{bmatrix} \delta d_1 / \delta V_3 \end{bmatrix}$$

$$S = \begin{bmatrix} 1/V_1 \end{bmatrix} * \begin{bmatrix} \delta a_2 / \delta \theta_1 \end{bmatrix} \quad S1 = \begin{bmatrix} 1/V_1 \end{bmatrix} * \begin{bmatrix} \delta d_2 / \delta \theta_1 \end{bmatrix}$$

$$T = \begin{bmatrix} \delta a_2 / \delta V_1 \end{bmatrix} \quad T1 = \begin{bmatrix} \delta d_2 / \delta V_1 \end{bmatrix}$$

$$U = \begin{bmatrix} 1/V_2 \end{bmatrix} * \begin{bmatrix} \delta a_2 / \delta \theta_2 \end{bmatrix} \quad U1 = \begin{bmatrix} 1/V_2 \end{bmatrix} * \begin{bmatrix} \delta d_2 / \delta \theta_2 \end{bmatrix}$$

$$V = \begin{bmatrix} \delta a_2 / \delta V_2 \end{bmatrix}$$

$$V1 = \begin{bmatrix} \delta d_2 / \delta V_2 \end{bmatrix}$$

$$W = \begin{bmatrix} 1/V_3 \end{bmatrix} * \begin{bmatrix} \delta a_2 / \delta \theta_3 \end{bmatrix}$$

$$W1 = \begin{bmatrix} 1/V_3 \end{bmatrix} * \begin{bmatrix} \delta d_2 / \delta \theta_3 \end{bmatrix}$$

$$X = \begin{bmatrix} \delta a_2 / \delta V_3 \end{bmatrix}$$

$$X1 = \begin{bmatrix} \delta d_2 / \delta V_3 \end{bmatrix}$$

$$A = \begin{bmatrix} 1/V_1 \end{bmatrix} * \begin{bmatrix} \delta a_3 / \delta \theta_1 \end{bmatrix}$$

$$A1 = \begin{bmatrix} 1/V_1 \end{bmatrix} * \begin{bmatrix} \delta d_3 / \delta \theta_1 \end{bmatrix}$$

$$B = \begin{bmatrix} \delta a_3 / \delta V_1 \end{bmatrix}$$

$$B1 = \begin{bmatrix} \delta d_3 / \delta V_1 \end{bmatrix}$$

$$C = \begin{bmatrix} 1/V_2 \end{bmatrix} * \begin{bmatrix} \delta a_3 / \delta \theta_2 \end{bmatrix}$$

$$C1 = \begin{bmatrix} 1/V_2 \end{bmatrix} * \begin{bmatrix} \delta d_3 / \delta \theta_2 \end{bmatrix}$$

$$D = \begin{bmatrix} \delta a_3 / \delta V_2 \end{bmatrix}$$

$$D1 = \begin{bmatrix} \delta d_3 / \delta V_2 \end{bmatrix}$$

$$E = \begin{bmatrix} 1/V_3 \end{bmatrix} * \begin{bmatrix} \delta a_3 / \delta \theta_3 \end{bmatrix}$$

$$E1 = \begin{bmatrix} 1/V_3 \end{bmatrix} * \begin{bmatrix} \delta d_3 / \delta \theta_3 \end{bmatrix}$$

$$F = \begin{bmatrix} \delta a_3 / \delta V_3 \end{bmatrix}$$

$$F1 = \begin{bmatrix} \delta d_3 / \delta V_3 \end{bmatrix}$$

The $G(3,3)$ matrix would be calculated in the same fashion except the partial derivatives would be of real and imaginary components of load currents which is f and x instead of a and d .

Needless to say $T(5,5)$ and $G(5,5)$ matrices would be calculated on similar lines.

Now $TG(k,j)$ matrix equals $G(k,j)$ for $k > 1$ and $j \geq 1$ and $k \neq j$ as given in equations (D.4-D.7). Since $G(k,j)$ matrices contain partial derivatives of device load currents, all other entries except those for device currents are zero. Since we consider real and imaginary components of device load currents at all buses (linear and nonlinear) the matrix is a diagonal matrix of order $(2n * 2n)$ which is $6 * 6$ in this case.

Hence $G(3,1)$ is given by

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & Z1 & Z2 \\ 0 & 0 & 0 & 0 & Z3 & Z4 \end{bmatrix}$$

The first four rows comprise of entries for real and imaginary device load currents for buses 1 and 2. Since they are linear buses these entries are zero. The last two rows are entries for bus 3 where $Z1$, $Z2$, $Z3$ and $Z4$ are the partial derivatives of the real and imaginary components of device load currents at bus 3 with respect to the voltage and phase angle at bus 3 as shown below.

where

$$Z1 = \begin{bmatrix} 3 & 1 & 1 \\ \delta f_3/V_3 & \delta\theta_3 & \end{bmatrix} \quad Z2 = \begin{bmatrix} 3 & 1 \\ \delta f_3/\delta V_3 & \end{bmatrix}$$

$$Z3 = \begin{bmatrix} 3 & 1 & 1 \\ \delta f_3/V_3 & \delta\theta_3 & \end{bmatrix} \quad Z4 = \begin{bmatrix} 3 & 1 \\ \delta x_3/\delta V_3 & \end{bmatrix}$$

Similarly $G(3,5)$, $G(5,1)$ and $G(5,3)$ can be calculated on similar lines.

The $H^{(k)}$ matrix for our system under consideration for $k = 1,3,5$ is a diagonal matrix of order $(2m * 2m)$ where m is the number of nonlinear buses.

Hence H^1 is given by

$$\begin{bmatrix} 1 & 1 \\ [\delta f_3/\delta\beta_3] & [\delta f_3/\delta\Gamma_3] \\ 1 & 1 \\ [\delta x_3/\delta\beta_3] & [\delta x_3/\delta\Gamma_3] \end{bmatrix}$$

Similarly H^3 and H^5 can be calculated on similar lines.

SELECTED BIBLIOGRAPHY

- [1] R. N. Dhar, "Computer Aided Power System Operation and Analysis", McGraw Hill Publishing Co., India.

- [2] George L. Kusic, "Computer Aided Power System Analysis", Prentice Hall, Englewood Cliffs, New Jersey 07632.

- [3] J. W. Walker, "Solutions of sparse network equations", Proc. IEEE, Vol. 55, pp. 1806-1810, 1967.

- [4] K. Zollenkopf, "Basic computation algorithm and programming techniques", 3rd Power System Computation Conference, PSCC Proc., Vol. 1, pp. 2-4, 1969.

- [5] N. Sato and W. F. Tinney, "Techniques for exploiting the sparsity of the network admittance matrix", IEEE Transactions on Power Apparatus and Systems, Vol. 82, pp. 944-950, 1963.

- [6] W. F. Tinney and J. W. Walker, "Solutions of sparse network equations by optimally ordered triangular factorisation", Proc. IEEE, Vol 55, pp. 1801-1809, 1967.

- [7] W. F. Tinney, "Some examples of sparse matrix methods for power network problems." in "Proceedings of the 3rd Power Systems Computation Conference" Rome, Vol. 1, pp. 210-215, 1969.

- [8] Electric Power Research Institute, " HARMFLO, version 4.0, Documentation."

- [9] A. Ralston and H. S. Wilf, "Mathematical Methods for Digital Computers", J. Wilky and Sons, Inc., New York, 1960.

- [10] B. Theraja, "A Textbook in Electrical Technology"
McGraw Hill Publication, India.
- [11] Aly A. Mahmoud, "Power system harmonics : An
assessment", Proc. IEEE, Vol. 72, pp. 1-11,
1984.
- [12] D. Xia, G.T. Heydt, "Harmonic power flow
studies, part I - formulation and solution,
IEEE Transaction Power Apparatus and Systems
Vol. PAS-101, No. 6, pp. 1257-1269, 1982.
- [13] W. Shepherd and P. Zand, "Energy Flow and Power
Factor In Non-sinusoidal Circuits", Cambridge
University Press, Cambridge, U.K., 1979.
- [14] A. P. (Sakis) Meliopoulos, K. R. Chakravati and
T. Ortmeier, "Effects of harmonics on power
equipment and loads", Proc. IEEE, Vol. 72,
pp. 21-31, 1984.
- [15] L. L. Freris, "Investigation of load flow
problem" Proceedings of IEE, Vol. 115,
pp. 1459-1470, 1968.
- [16] R. L. Smith and R. P. Stratford, "Power system
harmonics - effects from adjustable speed
drives", Proc. IEEE, Vol.72, pp. 69-73, 1984.
- [17] M. F. McGranaghan, "Example cases illustrating
harmonic analysis procedures", Proc. IEEE,
Vol. 72, pp. 134-138, 1984.
- [18] I. S. Gradshtyn, I. M. Ryzhik, "Tables of
Integrals, Series and Products", Academic
Press, New York, 1965.
- [19] William Grady, "Harmonic Power Flow Studies, A
thesis submitted to the faculty of Purdue
University", 1983.