

DIGITAL LOAD FLOW ANALYSIS
USING GENERAL MATRIX STORAGE AND
NON-ZERO ELEMENT STORAGE

by

William Moore Heldreth

Thesis submitted to the Graduate Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
ELECTRICAL ENGINEERING

APPROVED:

Dr. L. L. Grigsby, Chairman

Dr. L. Hasdorff

Dr. A. W. Bennett

March 1973

Blacksburg, Virginia

ACKNOWLEDGMENTS

The author wishes to thank _____ for suggesting this project and for his guidance during the initial stages. _____ and _____ were also very helpful to the author during the initial computer program development.

Special thanks are extended to Dr. L. L. Grigsby for his guidance during the important final phase of this project and to Dr. L. Hasdorff who aided the author over several obstacles during the development of this project.

The author is very grateful to _____ who typed this thesis. Finally, thanks must go to _____ for her moral support in the final stages of this project.

This research was sponsored by the Energy Research Group at Virginia Polytechnic Institute and State University.

TABLE OF CONTENTS

Chapter	Page
ACKNOWLEDGMENTS	ii
LIST OF FIGURES	v
LIST OF TABLES	vi
I. INTRODUCTION	1
II. THE SOLUTION OF THE DIGITAL LOAD FLOW PROBLEM	4
2.1 Introduction	4
2.2 Definition of Variables	5
2.3 The Newton Raphson Method	6
2.4 Basic Computer Algorithm Used	7
III. SOLUTION TECHNIQUES	14
3.1 Introduction	14
3.2 Solution Using General Matrix Storage	16
3.3 Solution Using Sub-Optimal Ordering and Non-Zero Element Storage	16
IV. RESULTS AND CONCLUSIONS	22
4.1 The Examples	22
4.2 Results Using General Matrix Storage	22
4.3 Results Using Non-Zero Element Storage with Sub-Optimal Ordering	24
4.4 Comparison of the Results	27
4.5 Suggestions for Further Development	30
BIBLIOGRAPHY	32

Chapter	Page
APPENDIX A - NEWTON RAPHSON METHOD FOR AC LOAD FLOW ANALYSIS	33
A.1 Modeling the System Components	33
A.2 Calculation of the Jacobian	42
A.3 The Iterative Process	44
A.4 Calculation of the Line Flows	48

VITA

LIST OF FIGURES

Figure	Page
2.1a Flow Chart of Newton Raphson Load Flow Technique	8
2.1b Flow Chart of Newton Raphson Load Flow Technique	9
2.1c Flow Chart of Newton Raphson Load Flow Technique	10
3.1 <u>YBUS</u> Non-Zero Element Storage Example	17
3.2 Jacobian Non-Zero Element Storage Example	19
A.1 π Equivalent of a Transmission Line	35
A.2a Transformer Equivalent Circuit	37
A.2b π Equivalent Transformer Circuit	37
A.3 Transmission Line π Equivalent	49

LIST OF TABLES

TABLE		PAGE
I	Computer Execution Times Using General Matrix Storage	23
II	Storage Requirements for General Matrix Storage Algorithm	25
III	Computer Execution Times Using Non-Zero Element Storage with Sub-Optimal Ordering	26
IV	Storage Requirements for Non-Zero Element Storage Algorithm	28
V	Comparison of Results	31
VI	Elements of the Jacobian	47

CHAPTER I

Introduction

The load flow study is of great importance to the electric utility. Exactly what is a load-flow analysis? "A continuous evaluation of the performance of a power system and analyzing the effectiveness of alternate plans for future expansion" (1). For many years the load flow problem has been used extensively in the power industry for many different types of problems.

The devices used to solve the problem have changed through the years. In the past the AC Network Analyzer was the major tool in this solution. The analyzer actually simulates power system components such as transmission lines, loads, generators, and other system elements. This unit uses high frequencies to reduce the voltage, current, and sizes of network components. This unit is still in use today on a limited basis. With the development of the digital computer, the load flow studies gradually made the transition from the analyzer method to the faster and more accurate digital computer. In addition to the digital computer solution, some work has been done using a hybrid computer. This hybrid computer is a combination of a digital computer and an analyzer. The digital machine connects, runs, and takes readings from the analyzer portion of the unit. The ultimate goal of this type of solution is a real time load flow analysis along with a real time control system for the power network. The load flow problem has been solved many different ways in the past. The outlook for the future is bright for another and even better solution technique.

The power companies have many reasons for solving a load flow study. These reasons break down into two major areas. One is the steady state solution of the operating system, and the other is the study of transients on the system. In the steady state analysis of a power network the load flow solution furnishes a great deal of information. The power, voltage, and current on a particular transmission line, the amount of generation needed to supply a known system load condition, the effect of deleting a generator, transmission line, or a load in the system, are all furnished via the load flow analysis.

This study is particularly important in showing the effect on the system of new components before the final plans are drawn up. This allows the utility to analyze the different types of proposed components to determine the one that will do the job in an efficient way. The load on a system is constantly changing, so many different cases of load flow studies are needed to maintain system security. An extension of the load flow problem is the optimal load flow which gives the utility an optimal route for power flow from generator to load to minimize transmission losses.

In addition to the steady state analysis, the load flow study is also used in transient studies. The knowledge of the line flows and generator outputs, before, during, and after the occurrence of a fault, is necessary to system stability studies. These transient studies take on many different forms depending on the type of fault or component failure. So, numerous case studies must be made in order to predict the effect of a fault on the system before it occurs so that preventative measures can be predetermined to minimize the instabilities in the

network. Thus, we see that the load flow study is used for many answers to power system questions.

The objectives of this thesis are (1) to develop a very general AC load-flow computer algorithm using the Newton Raphson technique with general matrix storage, and (2) to modify this program to use sub-optimal ordering and non-zero element storage. Once this has been accomplished, a comparison of the results from both algorithms will be made.

The following chapters develop the Newton Raphson solution technique for the load flow analysis, the two solution algorithms used, the results, and conclusions.

CHAPTER II

The Solution of the Digital Load Flow Problem

2.1 Introduction

In any engineering situation where the digital computer is involved there are two basic considerations that must be resolved in the development of an effective engineering computer program. One is "formulation of a mathematical model of the system or problem under consideration and second is the application of a numerical method to solve the model" (1).

In the load flow study each power system component must be simulated by an accurate model. The models or representations of these various components vary from scheme to scheme. The models used in the computer program developed for this application are summarized in Section A.1 of Appendix A. Also, many numerical techniques have been developed to solve the different load flow representations. These methods all have somewhat the same solution techniques, but the approach to the solution is different. It is generally accepted that the conditions on the selection of one of these methods are computer storage and execution time. A method that has a good balance between minimum storage and execution time is the most desirable. The basic technique used in this application was the Newton Raphson technique. A brief discussion of the Newton Raphson method will be included in Section 2.3 of this chapter, along with additional comments in Section A.2 of Appendix A.

2.2 Definition of Variables

Another important consideration in any engineering problem is the defining of the variables. The variables in the load flow study have been fairly well defined over the years. However, these variables do vary somewhat depending on the solution technique used. In most power network representations, four quantities at each bus can be defined as variables. These are the magnitude of the bus voltage, the angle of the bus voltage, the real power and the reactive power either entering or leaving a bus. In order to determine all four of these quantities at a bus, two of them must be known or specified.

The bus is typed depending upon which quantities are known. There are four classifications for buses: slack, load, generator, or voltage control. At the slack bus the voltage magnitude and angle are fixed, and the real and reactive power are unknown. There is only one slack bus in the system usually. This bus serves as a reference bus, and here the angle is zero. Since initially the transmission losses are not known, the generation at this bus is allowed to vary to compensate for these transmission losses. The load bus usually has a static load connected to it, so the known quantities here are real and reactive power. The voltage magnitude and angle are allowed to change to meet the load demand. The generator bus has a constant power source connected to it; and as in the load bus, the known quantities are real and reactive power. At many points in the system it is desired to maintain the voltage within certain limits. This type of bus is called a voltage control bus. The voltage magnitude and real power are known quantities at these

buses while the voltage angle and reactive power are allowed to fluctuate to maintain the voltage magnitude. In some cases a bus may not fall directly into one of these four types. So for these buses the given quantities are the real and reactive power as in a load or generator bus.

2.3 The Newton Raphson Method

In past years many digital load flow solution techniques have been developed. As the formulation of the load flow study became more and more sophisticated, the solution methods are advanced. In the last decade one solution technique has emerged to become the recognized standard upon which other methods are now judged. This solution technique is the Newton Raphson method. This technique is an iterative process for the solution of a set of simultaneous non-linear algebraic equations. In the load flow study these equations represent the power (real and reactive) at a bus as a function of the bus voltages. The actual Newton Raphson process is only about one-fourth of the entire program. The remaining three-fourths of the program involves the input, output, and preliminary calculations necessary to use this algorithm. "The Newton Raphson method is considered to have the best speed of convergence without much risk of divergence. This solution is also recognized to be very flexible as well as the standard solution technique" (2).

In the development of any computer program the problem usually dictates the amount of generality the program will have. The program developed for the AC load flow solution is very general and will handle most power system problems. This program is not an optimal computer

algorithm; however, in the development of this routine great care was taken to conserve computer storage and minimize execution time.

The program will handle such power system components as voltage control buses, static capacitor or inductor banks, off-nominal turns ratio transformers and synchronous condensers. The results of this program are the voltage magnitude and angle at each bus, the real and reactive bus powers, the line flows, the line losses, and total system losses. Since the development of the Newton Raphson method for load flow analysis has been formulated in both Stagg and El-Abiad (1) and Elgerd (3), an extensive development of this method will not be included in this paper. However, in order to better understand the operation of the basic computer algorithm developed, the next section of this chapter will include a flow chart and a short explanation of the basic routine. Appendix A contains a more detailed development of the Newton Raphson method.

2.4 Basic Computer Algorithm Used

The computer algorithm is basically the same for both solution techniques that were used. Figure 2.1 shows a general flow chart of the basic computer algorithm. This section will be concerned with a short explanation of the flow chart. A more detailed explanation of the method used is given in Appendix A. The first step in the algorithm is to read the number of each type of components in the system, the tolerances to which the system must converge, the data associated with the voltage control buses, and data for each transmission line. The next step depends upon which solution technique is to be used. If the general matrix storage method is to be used, the execution follows the

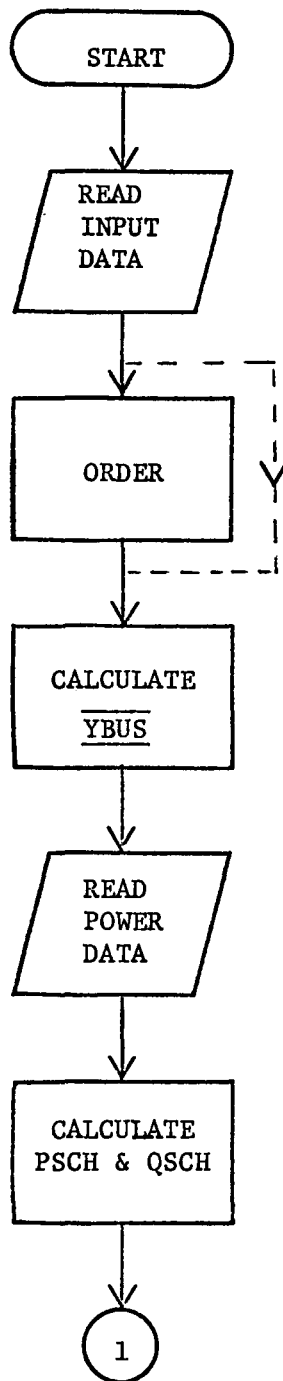


Figure 2.1a - Flow Chart of Newton Raphson Load Flow Technique

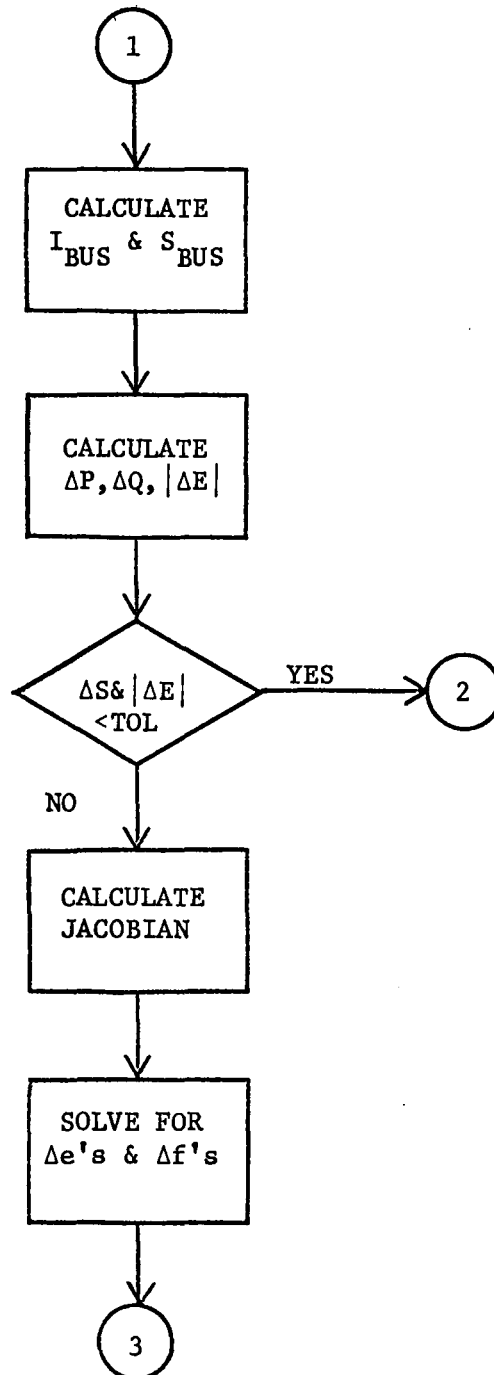


Figure 2.1b - Flow Chart of Newton Raphson Load Flow Technique

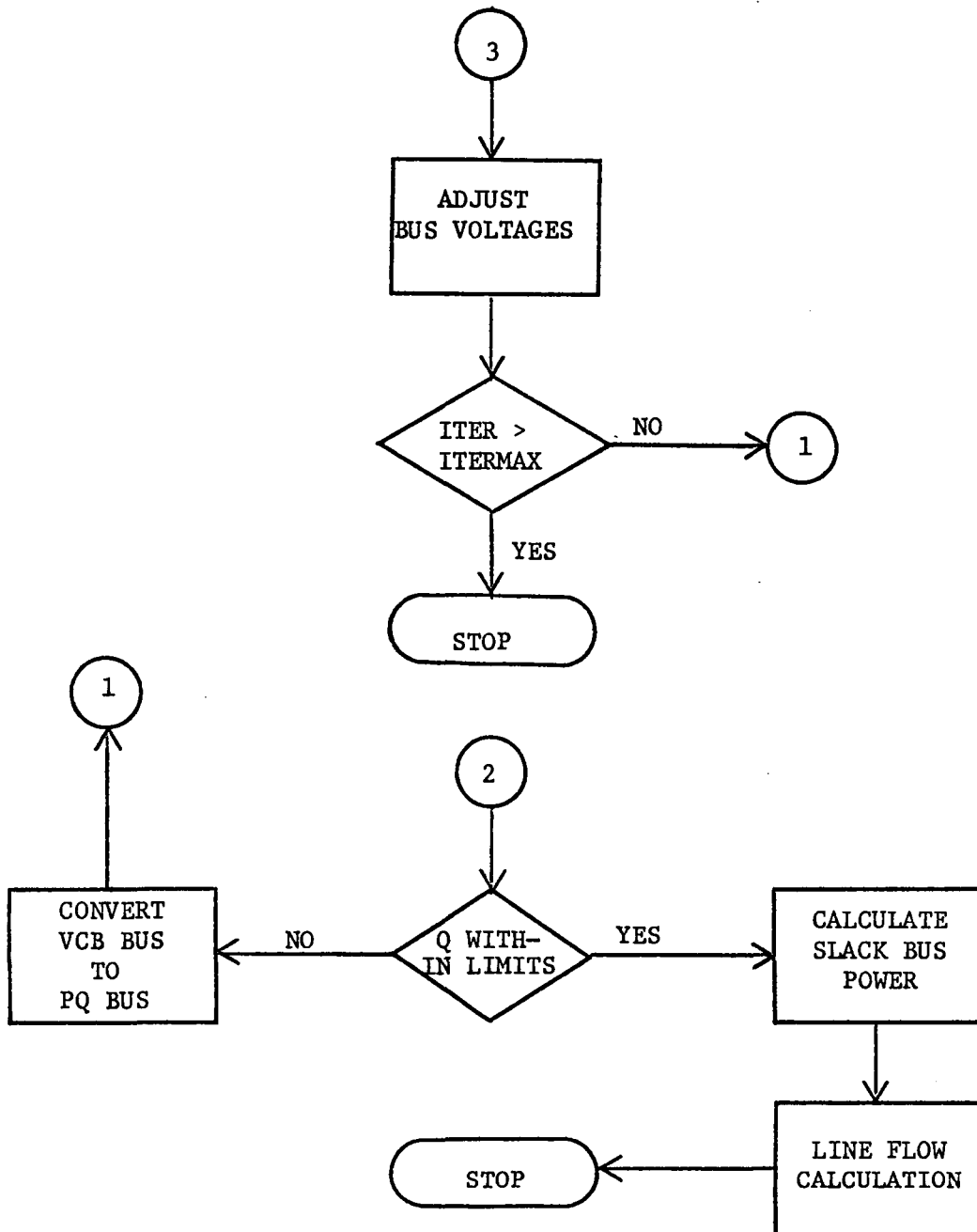


Figure 2.1c - Flow Chart of Newton Raphson Load Flow Technique

dotted line shown in Figure 2.1. However, if the non-zero element storage with sub-optimal ordering is to be used, the system is now ordered using the ordering technique which will be discussed in Chapter III. Now the $\overline{\text{YBUS}}$ is calculated using the two basic rules given in Section A.1 of Appendix A. The method of storage is determined by the solution technique that will be used.

The next step in the program is to read the values of fixed generation, static loads, and synchronous condensers along with the value of the slack bus voltage. From these values the computer determines the target values for convergence using the following equations

$$\overline{\text{PSCH}}_{\text{bus}} = \overline{\text{GP}}_{\text{bus}} - \overline{\text{LP}}_{\text{bus}} \quad (\text{a}) \quad 2.3-1$$

$$\overline{\text{QSCH}}_{\text{bus}} = \overline{\text{GQ}}_{\text{bus}} - \overline{\text{LQ}}_{\text{bus}} \quad (\text{b})$$

where the quantities GP, GQ are the generated real and reactive power and LP and LQ are the consumed real and reactive power.

Now using the $\overline{\text{YBUS}}$ and the initial guess for the bus voltages, the bus currents are calculated using equation

$$\overline{\text{I}}^{\circ}_{\text{bus}} = \overline{\text{YBUS}} \cdot \overline{\text{E}}^{\circ}_{\text{bus}} \quad 2.3-2$$

where the superscript $^{\circ}$ is the iteration count.

Then using these values of bus current, the bus powers are determined using equation

$$\overline{\text{S}}^{\circ}_{\text{bus}} = \overline{\text{E}}^{\circ}_{\text{bus}} \cdot \overline{\text{I}}^{\circ*}_{\text{bus}} \quad 2.3-3$$

Next the difference between the scheduled values of power and voltage magnitude at voltage control buses is calculated using equations

$$\Delta P_{bus}^{\circ} = \underline{PSCH}_{bus} - P_{bus}^{\circ} \quad 2.3-4$$

$$\Delta Q_{bus}^{\circ} = \underline{QSCH}_{bus} - Q_{bus}^{\circ} \quad 2.3-5$$

$$(\Delta |E|^{\circ})^2 = (|\underline{ESCH}|)_{bus}^2 - (|E|^{\circ})_{bus}^2 \quad 2.3-6$$

These values are then checked to see if these differences are less than the specified tolerances. If the mismatches are not less than the tolerances, the next step is to calculate the Jacobian. The method used varies between the two solution techniques, but the basic steps which are common to both methods are displayed in Section A.2 of Appendix A. The method of storage is essentially the only difference in the two methods. The methods used will be discussed in Chapter III. The solution of the equation

$$\underline{\Delta S}^{\circ} = \underline{J}^{\circ} \cdot \underline{\Delta E}^{\circ} \quad 2.3-7$$

is the next step. The two methods used will also be discussed in Chapter III.

Once the corrections for the bus voltages have been obtained from the solution technique and these corrections have been added to the old bus voltage values, the iteration number is checked against the iteration limit. This step eliminates a problem of excess run time on a diverging system. If the iteration limit has not been reached yet, the bus currents and bus powers are recalculated using the corrected values of bus voltages. This process continues until all the mismatches are equal to or less than the specified tolerances. When this becomes true, the reactive

power limits on the voltage control buses are checked. If these limits are violated, the bus is converted to a load bus; and the scheduled value of the reactive power is set to the limit violated and the iterative process starts over again. However, if these limits are satisfied, the slack bus power is calculated. Then the final step in the program is to calculate the line flows and the transmission line losses. The computer algorithm follows the formulation given in Section A.4 of Appendix A.

These are the basic steps both computer algorithms use in order to obtain convergence. The next chapter will discuss the two different solution techniques that were used.

CHAPTER III

Solution Techniques

3.1 Introduction

It was shown in Chapter II and Appendix A that the Newton Raphson Load Flow technique is essentially the solution of a set of non-linear simultaneous equations which have been linearized by the use of a Taylor series expansion around a starting point. The result of this linearization is a set of linear simultaneous equations in the form of equation A.3-2 which is repeated here for reference.

$$\underline{\Delta S}^{\circ} = \underline{J}^{\circ} \underline{\Delta E}^{\circ} \quad 3.1-1$$

Where $\underline{\Delta S}$ is the differences between the scheduled and calculated values of power, the \underline{J}° is the Jacobian matrix and $\underline{\Delta E}^{\circ}$ is the vector of unknown voltage corrections.

There are many different numerical methods that can be used to solve this set of equations for the corrections in the bus voltages. Each of these different methods essentially determines the inverse of the Jacobian matrix and premultiplies this inverse on Equation A.3-2 resulting in equation

$$\underline{J}^{\circ -1} \underline{\Delta S}^{\circ} = \underline{\Delta E}^{\circ} \quad 3.1-2$$

Most of the solution time in any Newton Raphson load flow solution is taken up in finding the inverse of the Jacobian, as will be shown later. As the size of the system under study doubles, the number of Jacobian matrix elements quadruples in size. So, the problem of finding the inverse of the Jacobian for a large power system is very difficult and lengthy.

Another consideration is the computer memory needed to store this Jacobian. The storage requirements for a conventional power system would be almost impossible if every element in the matrix was stored. For example, to store the Jacobian for a 118 bus system it would take approximately 220 K of computer storage. At this point several properties of this Jacobian must be stated. Due to the fact that very rarely is every bus connected to every other bus in a power system, there will be zero entries in the Jacobian where there is no connection between the bus corresponding to the row and the bus corresponding to the column. In a typical power system there are many cases where buses are not directly connected, so the Jacobian is a sparse matrix. And as the size of the system increases, the matrix becomes even more sparse. Also, the Jacobian matrix is structurally symmetric if no voltage control buses are present. Structurally symmetric means that the four submatrices of the Jacobian have non-zero elements in the same locations. However, the values of these non-zero elements is different. However, the inclusion of these buses destroy this symmetry and most power systems have voltage control buses. These buses also contribute to the sparsity of the Jacobian.

With these properties in mind, two different computer algorithms were developed to solve the load flow problem. One solution technique uses general matrix storage and the other uses a form of sub-optimal ordering with non-zero element matrix storage. The next two sections develop each of these methods and gives the merits and demerits of each. The next chapter shows the results of the two methods on several power system examples.

3.2 Solution Using General Matrix Storage

In this computer algorithm all matrices were stored and operated on with all elements represented. The YBUS and the Jacobian are calculated using a step-by-step approach which was very easy to implement on the digital computer. Since all elements were retained in both of these matrices, no problems were encountered in their calculation.

The solution technique used in this algorithm to invert the Jacobian and obtain the voltage corrections was a Gauss Elimination method with back substitution. In this method the largest pivotal element is used which is accomplished by the interchanging of rows and columns. The back substitution is calculated by successive substitutions. This method has been developed in many books on numerical analysis, and the detailed development will not be given here. The actual computer routine used for this solution was obtained from the IBM Scientific Subroutine Manual (4). The name of the routine used is SIMQ. The results of this solution technique will be shown in Chapter IV.

3.3 Solution Using Sub-Optimal Ordering and Non-Zero Element Storage

It was noted in the Introduction of this chapter that both YBUS and the Jacobian matrix were sparse matrices. Since both of the matrices contain zero elements, a computer algorithm was written to eliminate the storage of these zero elements. Since the YBUS is symmetric, further storage could be saved by only storing the diagonal and the upper or lower triangular off-diagonal terms. The upper triangular off-diagonal terms were chosen in this case. The method used to accomplish this non-zero element symmetrical storage is as follows. First, a complete row

of the YBUS was calculated using the rules quoted in Appendix A. This row was stored in a work vector. Then this work vector was searched for non-zero elements from the diagonal term to the last column. There was three vectors used to store YBUS non-zero. YROW is an integer vector that contains the starting location in YCOL which corresponds to a new row. YCOL is an integer vector that contains the column numbers of the non-zero elements in that row and the complex vector Y which contains the actual values of the non-zero elements. In Figure 3.1 an example that illustrates this method is shown.

$$\underline{\text{YBUS}} = \begin{bmatrix} \bar{Y}_{11} & 0 & \bar{Y}_{13} \\ 0 & \bar{Y}_{22} & \bar{Y}_{23} \\ \bar{Y}_{31} & \bar{Y}_{32} & \bar{Y}_{33} \end{bmatrix}$$

Note $\bar{Y}_{13} = \bar{Y}_{31}$ and $\bar{Y}_{23} = \bar{Y}_{32}$.

The resulting non-zero storage representation will be

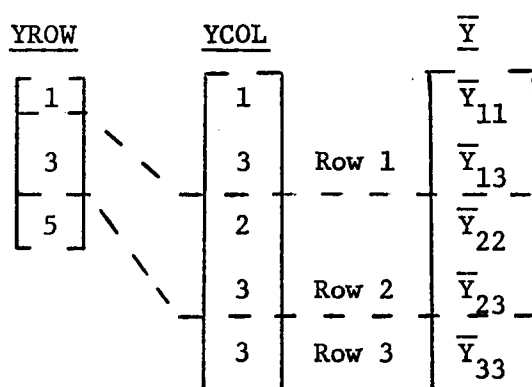


Figure 3.1 - YBUS Non-Zero Element Storage Example

In the case of the Jacobian, symmetric non-zero storage could not be used. In order to use the solution technique, which will be discussed later, the diagonal elements of the Jacobian were required to be in a separate vector. So, in the case of the Jacobian, four vectors were used. JD is a real vector of all the diagonal terms. JROW is an integer vector containing the starting location in JCOL which corresponds to a new row. JCOL is an integer vector that contains the column numbers of the non-zero elements in that row and the real vector JN which contains the values of the off-diagonal elements of the Jacobian.

The method to accomplish this storage is that a single row of the Jacobian is calculated and stored in a work vector. Then the diagonal element is stored in the location of JD which corresponds to that row. Then the rest of the work vector is searched, and the non-zero off-diagonal elements are stored in JN with the appropriate values in JCOL. The JROW vector is updated each time a new row is calculated. To illustrate this procedure, Figure 3.2 shows a sample row calculation of the Jacobian in the work vector W.

$$\underline{W} = [J_{11} \ J_{12} \ 0 \ 0 \ J_{15} \ J_{16} \ J_{17} \ 0 \ 0 \ J_{110}]^t$$

This being the first row of the Jacobian, the non-zero storage vectors will have the following form:

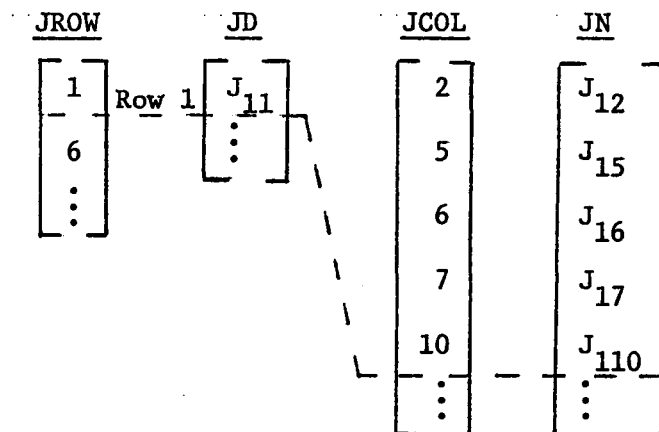


Figure 3.2 - Jacobian Non-Zero Element Storage Example

The calculation of both the YBUS and the Jacobian and then storing both of these matrices non-zero required considerable more computer statements. The addition of voltage control buses, off-nominal turns ratio transformers and static capacitors also add to the programming complexity. Another aspect that creates some problems is that since only the upper half of the YBUS is available when the Jacobian is calculated, an involved search procedure must be used to get all of the needed information from the YBUS. The section in the computer algorithm that calculates the YBUS and the Jacobian is approximately three times as long as the simple technique used in the previous algorithm. Some comparisons between these two methods will be made in the next chapter.

After the implementation of this algorithm and the solution technique was made, it was noted that very poor convergence was obtained and on many systems that the storage requirements and execution times were not better. Due to this fact it was decided to incorporate a pre-ordering technique. The objective of any ordering routine is to

preserve the sparsity in the Jacobian as much as possible during the solution process. Thus, by arranging the terms around the diagonal the time needed for the triangularization is reduced. In many cases this method of ordering is called optimal ordering. Most of the time, however, an optimal arrangement of numbering the buses is not achieved; so the term of sub-optimal ordering has been adopted for use in this paper. The method essentially does the reordering of the system by counting the number of lines connected to each bus excluding the slack bus which maintains its number of one. Then the routine orders the remaining buses in order of increasing number of lines connected. If two buses have the same number of lines, either one can be picked first and then the other. Two vectors return the reordered system to the main routine.

LIST (K) equals the number of bus ordered K^{Th} .

LISTJ (L) equals the order of the L^{Th} bus.

After the incorporation of this ordering routine, better storage and run times were obtained.

The solution technique used in this method to solve for the changes was obtained from Stott (5). This method is somewhat similar to Crout's method. This method works on a row at a time. The first is normalized (all elements are divided by diagonal element) and then it is restored. Then row two is put into the working vector, and all elements to the left of the diagonal are eliminated by combination with row one and then normalized and restored. At each step the appropriate operations are also carried out on the independent vector as in this case. This

process continues until the upper-triangular reduction of the Jacobian is available, and the corresponding elements in the ΔS vector become the corrections to the bus voltages. This solution technique requires that the vectors JCOL and JN have some unused storage space at the top of these vectors to prevent storage overlapping. The amount necessary varies and was found by experiment in most cases.

The next chapter will give the results of both of the two solution techniques.

CHAPTER IV

Results and Conclusions

4.1 The Examples

Seven different power system examples were used to compare the two solution techniques used in this project. The examples range from a simple 3 bus case to one with 118 buses. The following are the seven examples used and the sources from which they were obtained. The three bus example is a textbook example from Elgerd (3). The first five bus example is also a textbook example from Stagg and El-Abiad (1). The second five bus example is again a textbook problem taken from Stevenson (6). The remaining four examples are actual power system problems which were obtained from American Electric Power (7).

The results of both computer algorithms for all the test runs were checked with the published results to verify the validity of the two solution techniques. In all cases the answers obtained by the two separate algorithms had at most a one percent derivation from the published results. The next two sections of this chapter give the solution times and storage requirements of the two solution techniques.

4.2 Results Using General Matrix Storage

Table I shows the computer cpu times for the total execution time as well as times for several major calculations within the program. The times shown for the YBUS calculation includes the line data input time as well as the actual loop that forms the YBUS. The line input

TABLE I

COMPUTER EXECUTION TIMES USING GENERAL MATRIX STORAGE

System	YBUS Calcu- lation	Jacobian Calcu- lation	Inversion Time	1 Iteration Time	Total Run Time	No. of Iterations
Elgerd 3 BUS	.01	~	~	.01	.07	3
STAGG 5 BUS	.04	~	.01	.02	.15	3
Stevenson 5 BUS	.05	~	.01	.02	.13	2
AEP 14 BUS	.1	.02	.27	.37	.91	2
AEP 30 BUS	.22	.12	3.11	3.30	9.44	3
AEP 57 BUS	.40	.44	21.28	22.05	64.48	3
AEP 118 BUS	.92	2.0	185.86	185.71	594.35	3

~ Denotes time value was too small for use in the timing routine.

All times shown are in seconds.

time was included in this part due to the fact that as the line data is read in all off-nominal turns, ratio transformer data is converted to a π equivalent transmission line representation. This allows the YBUS to be calculated directly. The times shown for the calculation of the Jacobian are the average times for the actual calculation. The solution times given include the solution process as well as the back substitution process. These times have also been averaged. The times given for one iteration include the data input through the correction of the bus voltages. The data read in time is only included in the time for the first iteration. The times given are an average value for total number of iterations. The total times include the input and output times as well as the computation time. All times are in seconds. The starting voltages used for these examples for all buses except the slack bus were 1.0-j.1 per unit voltage. This starting voltage was used so that a comparison could be made with the non-zero element storage solutions.

Table II gives the storage requirements for the seven examples using general matrix storage. All values include the storage of the program object code as well as the array storage. All values used in the program were stored in the single precision (REAL *4) except for the integer numbers, and they were stored in half precision (Integer *2).

4.3 Results Using Non-Zero Element Storage with Sub-Optimal Ordering

Table III shows the computer cpu times which correspond to the same format used in Table I. The sections of Table III correspond to the same parts of the program timed as explained for Table I except that the times for the calculation of YBUS, times for one iteration and the

TABLE II

STORAGE REQUIREMENTS FOR GENERAL MATRIX STORAGE ALGORITHM

System	Storage
Elgerd 3 BUS	35.45 K
STAGG 5 BUS	36.34 K
Stevenson 5 BUS	36.16 K
AEP 14 BUS	43.2 K
AEP 30 BUS	67.59 K
AEP 57 BUS	146 K
AEP 118 BUS	487.54 K

K = 1,000 Bytes

TABLE III
COMPUTER EXECUTION TIMES USING NON-ZERO ELEMENT STORAGE WITH
SUB-OPTIMAL ORDERING

System	YBUS Calculation	Jacobian Calculation	Inversion Time	1 Iteration Time	Total Run Time	No. of Iterations
Elgerd 3 BUS	.02	~	~	.01	.07	3
STAGG 5 BUS	.03	~	.01	.02	.17	3
Stevenson 5 BUS	.03	~	~	.02	.13	2
AEP 14 BUS	.1	.02	.10	.16	.60	2
AEP 30 BUS	.28	.13	.71	.85	3.28	3
AEP 57 BUS	.6	.44	4.19	4.71	15.54	3
AEP 118 BUS	1.99	1.76	78.65	83.25	278.6	3

~ Denotes time value was too small for use in the timing routine.

All times shown are in seconds..

total run times now also include the time used to order the system. All times are again given in seconds. The starting voltages for all buses except the slack bus were $1.0-j.1$ per unit voltage. The reason this value of starting voltage was used is that the non-zero solution technique requires that the diagonal terms of the Jacobian be non-zero. A starting voltage of $1.0 + j0.0$ per unit creates a zero diagonal term in the Jacobian where voltage control buses occur. However, almost any value within plus or minus 10 percent of $1.0 + j0.0$ per unit would be allowable as a starting value.

Table IV gives the storage requirements for the examples using non-zero element storage. Again, all values include the storage of the program object code as well as the array storage. It must be noted here that the actual program using the non-zero element storage with sub-optimal ordering technique is considerably longer than the program which used general matrix storage. Therefore, the object code generated by this program will require more storage space.

4.4 Comparison of the Results

In comparing Tables I and III no real differences exist between the two solution techniques for the smaller systems. This is due to the fullness of the YBUS and the Jacobian for these systems. So, the time savings using non-zero element storage are not significant for small systems. However, as the size of the system increased, considerable time was saved in every case. The times needed to calculate the YBUS using non-zero storage increased mainly due to the inclusion of the time taken to order the system in these values. The most significant savings

TABLE IV

STORAGE REQUIREMENTS FOR NON-ZERO ELEMENT STORAGE ALGORITHM

System	Storage
Elgerd 3 BUS	41.74 K
STAGG 5 BUS	42.3 K
Stevenson 5 BUS	42.24 K
AEP 14 BUS	45.14 K
AEP 30 BUS	52.67 K
AEP 57 BUS	71.86 K
AEP 118 BUS	381.54 K

K = 1,000 Bytes

using non-zero storage was in the time needed to triangularize the Jacobian and perform the back substitution. Approximately an 80 percent time saving was experienced in the use of non-zero storage technique. This time saving is reflected also in the times for one iteration and the total run times. However, it must be noted that the times for the 118 bus system were not as good as expected. The author feels that the reason behind this problem is that the Jacobian for the 118 bus system is very sparse, but the inverse is very full which causes many operations to be performed for triangularization to be accomplished. One solution to this problem could be the addition of a preordering technique that simulates this elimination process before it takes place and orders the system in such a way to minimize the number of non-zero terms in the inverse.

The comparison of Tables II and IV shows that for the smaller systems the storage requirements are more using non-zero storage. However, as noted before, the values also include the storage of the program object code, and in the case of the non-zero element storage program approximately 6,000 more bytes are needed to store the non-zero element storage program object code. However, as the size of the system increases, the storage requirements using non-zero element storages show a savings in computer storage. This is shown by comparing the storage requirements of some of AEP test cases. Again, the 118 bus gave less storage saving than some of the other systems. Here again the author feels that due to the great amount of storage to store the full triangularized Jacobian is one of the main reasons for this result. Again, the addition of an

ordering technique that would minimize the number of non-zero terms in the inverse could be used to solve this storage problem also.

Table V shows the comparison of the results for the AEP test cases. The total run times, as well as the storage requirements for these systems, are shown. Only these systems were used due to the simplicity of the other examples. Again, all times shown are in seconds and the K is equal to 1,000 Bytes.

Overall the author feels that the algorithm using non-zero element storage with the form of sub-optimal ordering used is a vast improvement over the algorithm that uses general matrix storage. Both major considerations in a good engineering computer program, time and storage have been reduced with the non-zero storage technique. Both objectives of this thesis have been accomplished, and the results essentially display the results expected.

4.4 Suggestion for Further Development

The non-zero element storage program could now be changed to extend its capability. One significant addition would be the addition of a sensitivity analysis section. Another development could be the modification of the ordering routine to minimize the number of terms in the inverse of the Jacobian. This addition could further reduce storage and execution time. The addition of a section to simulate a DC link in the AC network could be an additional advancement. Also, the capability of area interchange could have some practical applications. One minor change could be to modify the input to the program to conform to the Common Format for Load Flow Data.

TABLE V
COMPARISON OF THE RESULTS

System	Total Times		Storage Requirements	
	GMS	NZES	GMS	NZES
AEP 14 BUS	.91	.60	43.2 K	45.14 K
AEP 30 BUS	9.44	3.28	67.59 K	52.67 K
AEP 57 BUS	64.48	15.54	146.0 K	71.86 K
AEP 118 BUS	594.35	278.6	487.54 K	381.54 K

All times shown are in seconds.

K = 1,000 Bytes

BIBLIOGRAPHY

Literature Cited

1. Stagg, Glenn W. and El-Abiad, Ahmed H. Computer Methods in Power System Analysis. New York, New York: McGraw-Hill, 1968.
2. Tinney, W. F. "Notes on Newton Raphson Method for Solution of AC Power Flow Problem." Bonneville Power Administration, Portland, Oregon, April 1971.
3. Elgerd, Olle I. Electric Energy Systems Theory: An Introduction, New York, New York: McGraw-Hill, 1971.
4. 'System/360 Scientific Subroutine Package Version III Programmer's Manual' No. GH20-0205, International Business Machines Corporation, White Plains, New York, 1972.
5. Stott, Brian. "Load Flows for A.C. and Integrated A.C./D.C. Systems." The Victoria University of Manchester, Manchester, England, 1971.
6. Stevenson, William D., Jr. Elements of Power System Analysis, Second Edition, New York, New York: McGraw-Hill, 1962.
7. 'AEP Standard Test Systems,' American Electric Power Service Corporation, New York, New York, 1961.

APPENDIX A

The Newton Raphson Method for AC Load Flow Analysis

This appendix deals with the actual scheme used to develop the AC load flow computer algorithm. The modeling of various system components will be summarized here. Also, some basic aspects of the Newton Raphson method will be summarized so that the modifications made in this method for the addition of the DC solution technique can be better understood. A short discussion of the particular solution technique will be presented again to give some general knowledge of the method used. The convergence criteria will be discussed along with a summary of the line flow calculation in the program.

The Newton Raphson technique used here uses the bus admittance frame of reference. The number of buses in the system is represented by n , and it is assumed that by renumbering the slack bus always has the number one. This computer program was written in Fortran IV and run on an IBM model 370 computer.

A.1 Modeling the System Components

There are numerous different representations for power system components depending on which solution technique is used. These models have been derived in most power system textbooks so the derivation will not be discussed here. The reader should refer to either Stagg and El-Abiad (1) or Elgerd (3) for the details. The different components will be discussed in the order that the computer algorithm considers them. The first model that is considered is that of the transmission line. All

transmission lines are represented as π equivalent lines as shown in Figure A.1. In order to interpret these π equivalents into a matrix representation that the computer routine can use, a nodal admittance matrix (YBUS) is calculated from the values of the series admittance of the line \bar{Y}_{SER} and the shunt admittances $\bar{Y}_{SH/2}$ for each line. The routine to accomplish this task is very straightforward. Two basic rules have been formulated for this purpose.

"Rule 1. The diagonal elements of YBUS, the \bar{Y}_{ii} terms are obtained as the algebraic sum of all the admittances incident to node i .

"Rule 2. The off-diagonal elements \bar{Y}_{ij} and \bar{Y}_{ji} are obtained as the negative of the admittance connecting nodes i and j " (3).

Noting also that all the off-diagonal terms that correspond to two buses that are not directly connected are equal to zero, the YBUS can be formulated.

In addition to the transmission lines two other components are modeled in such a way that they too are part of the YBUS. These elements are the static inductor or capacitor bank and the off-nominal turns ratio transformer. The known quantities of these elements allow easy formulation for their incorporation into the YBUS. In the case of the static capacitor or inductor bank, the representation or model used is a shunt element to ground at the buses where they are connected. Since this element can either be an inductor or capacitor bank, the value of the element determines the type. A positive susceptance is a capacitive bank, and a negative susceptance is an inductive bank. Since these static elements are simply a shunt to ground at a bus, by the use of Rule #1 for the formulation of the YBUS the effect of this element can

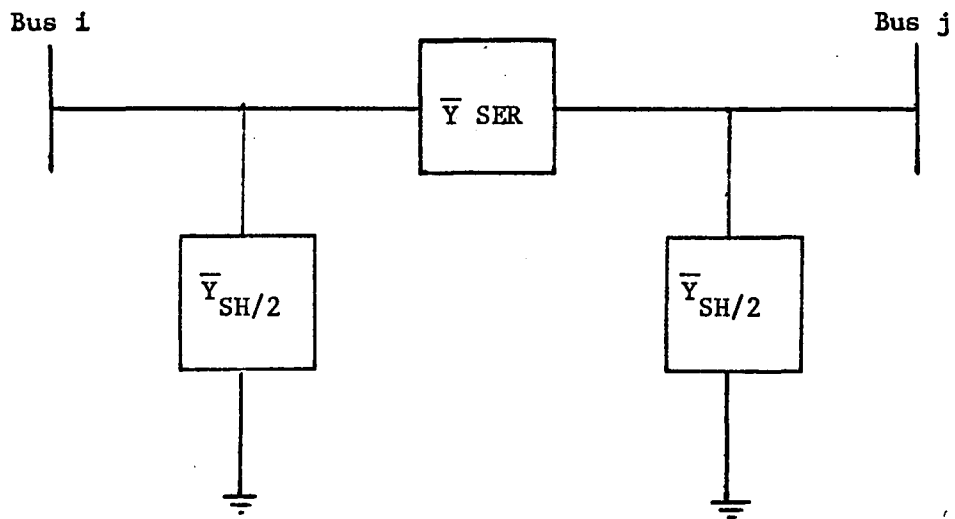


Figure A.1 π Equivalent of a Transmission Line

be included in the YBUS. The diagonal term of the matrix that corresponds to the bus where the element is located will be the only element affected. Thus, to include a static element the susceptance of the element is added to the appropriate diagonal term.

In most load-flow studies there are some off-nominal turns ratio transformers present. These transformers are needed to maintain voltages at certain critical points in the system. In order to incorporate these transformers into the YBUS, the transformer must be modeled in the form of a π equivalent transmission line. Then the two basic rules given earlier can be used to add the effect of the transformer to the YBUS. So, in effect, the transformer circuit must be converted to an equivalent π circuit. Shown in Figure A.2a is an off-nominal turns ratio transformer with a turns ratio of $a:1$ located between buses i and j of a power system. Using the derivation in Stagg and El-Abiad (1) it can be shown that the π representation of such a transformer is the one shown in Figure A.2b. Once the π equivalent circuit has been obtained, the transformer can be included in the YBUS.

Once the YBUS has been formulated, the next step is to calculate the bus currents and powers. A set of initial bus voltages must be supplied in order to perform these calculations. These voltages represent the initial guess, and they are usually equal to $1.0 + j 0.0$. Also, the modeling of the generators, loads, and synchronous condensers must be accomplished to set target values for convergence purposes.

Using the YBUS and the set of initial guesses for the bus voltages \bar{E}_{bus}^o , the current at each bus can be calculated using

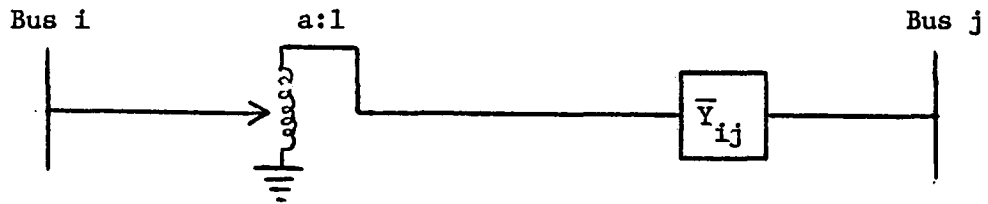


Figure A.2a Transformer Equivalent Circuit

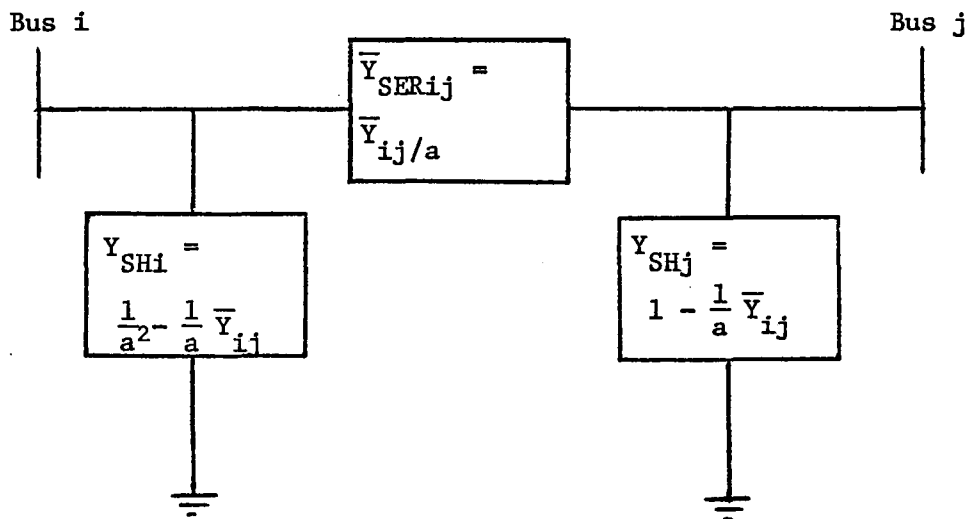


Figure A.2b π Equivalent Transformer Circuit

$$\underline{\bar{I}}_{bus}^{\circ} = \underline{\bar{Y}}_{BUS} \cdot \underline{\bar{E}}_{bus}^{\circ} \quad A.1-1$$

To illustrate the expanded equation for current bus p will be used, and using equation A.1-1 the current at bus p will be

$$\underline{\bar{I}}_p^{\circ} = \underline{\bar{Y}}_{pp} \cdot \underline{\bar{E}}_p^{\circ} + \sum_{\substack{q=1 \\ p \neq q}}^n \underline{\bar{Y}}_{pq} \cdot \underline{\bar{E}}_q^{\circ} \quad A.1-2$$

Now by representing each complex quantity by its real and imaginary parts where

$$\underline{\bar{Y}}_{pp} = G_{pp} - j B_{pp} \quad (a)$$

$$\underline{\bar{E}}_p^{\circ} = e_p^{\circ} + j f_p^{\circ} \quad (b) \quad A.1-3$$

$$\underline{\bar{I}}_p^{\circ} = c_p^{\circ} + j d_p^{\circ} \quad (c)$$

and substituting A.1-3 into A.1-2 and performing the indicated multiplication, the expanded equation for the bus current at bus p is

$$\begin{aligned} \underline{\bar{I}}_p^{\circ} = c_p^{\circ} + j d_p^{\circ} = \{ & G_{pp} \cdot e_p^{\circ} + B_{pp} f_p^{\circ} - \sum_{\substack{q=1 \\ q \neq p}}^n (G_{pq} \cdot e_q^{\circ} + B_{pq} \cdot f_q^{\circ}) \} \\ & + j \{ G_{pp} \cdot f_p^{\circ} - B_{pp} \cdot e_p^{\circ} - \sum_{\substack{q=1 \\ q \neq p}}^n (G_{pq} \cdot f_q^{\circ} - B_{pq} \cdot e_q^{\circ}) \} \end{aligned} \quad A.1-4$$

Using the current and the initial guess voltages at the buses, the power at all the buses excluding the slack bus can be calculated using

$$\underline{\bar{S}}_{bus}^{\circ} = \underline{\bar{E}}_{bus}^{\circ} \cdot \underline{\bar{I}}_{bus}^{\circ *} \quad A.1-5$$

Again, bus p will be used to illustrate the expanded equation for power.

$$\overline{S}_p^{\circ} = \overline{E}_p^{\circ} \cdot \overline{I}_p^{\circ*} \quad \text{A.1-6}$$

Now by representing each complex quantity by its real and imaginary parts where \overline{S}_p° is

$$\overline{S}_p^{\circ} = P_p^{\circ} + j Q_p^{\circ} \quad \text{A.1-7}$$

the following equation results

$$P_p^{\circ} + j Q_p^{\circ} = (e_p^{\circ} + j f_p^{\circ}) \cdot (c_p^{\circ} + j d_p^{\circ})^* \quad \text{A.1-8}$$

Taking the complex conjugate of the bus current equation A.1-4, substituting this current into equation A.1-8 and performing the indicated multiplication, the expanded equation for bus power at bus p becomes

$$\begin{aligned} P_p^{\circ} + j Q_p^{\circ} = & \{ [e_p^{\circ} (G_{pp} \cdot e_p^{\circ} + B_{pp} \cdot f_p^{\circ}) + f_p^{\circ} (G_{pp} \cdot f_p^{\circ} - B_{pp} \cdot e_p^{\circ})] \\ & - \sum_{\substack{q=1 \\ q \neq p}} \{ e_p^{\circ} (G_{pq} \cdot e_q^{\circ} + B_{pq} \cdot f_q^{\circ}) + f_p^{\circ} (G_{pq} \cdot f_q^{\circ} - B_{pq} \cdot e_q^{\circ}) \}] \\ & + j [\{ e_p^{\circ} (B_{pp} \cdot e_p^{\circ} - G_{pp} \cdot f_p^{\circ}) + f_p^{\circ} (G_{pp} \cdot e_p^{\circ} + B_{pp} \cdot f_p^{\circ}) \} \\ & - \sum_{\substack{q=1 \\ q \neq p}} \{ e_p^{\circ} (B_{pq} \cdot e_q^{\circ} - G_{pq} \cdot f_q^{\circ}) + f_p^{\circ} (G_{pq} \cdot e_q^{\circ} + B_{pq} \cdot f_q^{\circ}) \}] \end{aligned} \quad \text{A.1-9}$$

Using equations A.1-4 and A.1-9 the current and power for each bus (excluding the slack bus) can be calculated.

In modeling power system elements such as generators, loads and synchronous condensers, the common representation for these elements is a static source or sink of power. Using these representations, a set of values that the load flow solution must satisfy is obtained. Since

a bus could have a load, a generator or a synchronous condenser, the effect of each of these elements must be considered when calculating the scheduled values for bus powers. Representing static generation by \overline{GEN} , a static load by \overline{LOAD} , synchronous condensers by \overline{SYSNC} and the scheduled power by \overline{SSCH} the following equation can be written

$$\overline{SSCH}_{bus} = \overline{GEN}_{bus} + \overline{SYSNC}_{bus} - \overline{LOAD}_{bus} \quad A.1-10$$

Now taking only bus p under consideration and representing all the elements of equation A.1-10 by their real and imaginary parts, the following equations result

$$\overline{SSCH}_p = PSCH_p + j QSCH_p \quad (a) \quad A.1-11$$

$$\overline{GEN}_p = GP_p + j GQ_p \quad (b)$$

$$\overline{SYSNC}_p = \pm j SYSN_p \quad (c) \quad A.1-11$$

$$\overline{LOAD}_p = LP_p + j LQ_p \quad (d)$$

Since the synchronous condenser is a reactive power compensation element, it has no component of real power and the imaginary component can have either a positive or negative value depending on whether it is a sink or a source of reactive power. Substituting A.1-11 into A.1-10 and separating the real and imaginary parts, the following equations result:

$$PSCH_p = GP_p - LP_p \quad (a) \quad A.1-12$$

$$QSCH_p = GQ_p \pm SYSN_p - LQ_p \quad (b)$$

A.1-12 will give the scheduled real and reactive power at bus p, and by changing the value of p all necessary scheduled values for power can be obtained.

\overline{PSCH}_{bus} and \overline{QSCH}_{bus} having been calculated, the difference between these scheduled values and the actual calculated power must be determined. This simple process is very critical as the absolute value of this difference between the scheduled value and calculated value for real and reactive power are obtained by using:

$$\Delta P_{bus}^{\circ} = PSCH_{bus} - P_{bus}^{\circ} \quad (a)$$

$$\Delta Q_{bus}^{\circ} = QSCH_{bus} - Q_{bus}^{\circ} \quad (b)$$

A.1-13

At voltage control buses, a somewhat different procedure is used. Since real power (PSCH) and voltage magnitude $|ESCH|$ are known, these are the values that are monitored for convergence. The real power difference ΔP_p° is calculated using equation A.1-13a. The difference between the scheduled and calculated voltage magnitude is obtained by

$$(\Delta |E|^{\circ})_{bus}^2 = (|ESCH|)_{bus}^2 - (|E|^{\circ})_{bus}^2 \quad A.1-14$$

Having determined all the ΔP 's, ΔQ 's and $|\Delta E|^2$'s, a vector of these values is set up to allow their use in the Newton Raphson method. Since the slack bus is not included in these calculations, there are $2(n-1)$ values. In order to use these values, they are arranged in a vector as:

$$\underline{\Delta S}_{bus}^{\circ} \triangleq [\Delta P_2^{\circ}, \Delta P_p^{\circ}, \dots, \Delta P_n^{\circ}, \Delta Q_2^{\circ}, \Delta Q_3^{\circ}, \dots, \Delta Q_n^{\circ}]^t \quad A.1-15$$

At voltage control buses the ΔQ term is replaced by $|\Delta E|^2$. The ΔQ section of the vector will contain $|\Delta E|_p^2$ terms where p corresponds to a voltage control bus.

In this section the basic models for the power system components have been discussed. The use of these models allows the load-flow

problem to be formulated into acceptable form for the Newton Raphson solution method. The following sections will show how this basic formulation is used to solve the AC load-flow problem.

A.2 Calculation of the Jacobian

In the previous section equation A.1-9 was developed for the actual power at any bus in the system. From this equation it can be shown that the actual bus power is a function of the \overline{YBUS} and the bus voltages. Since the slack bus voltage is fixed, the actual power at this bus is not calculated until the solution is reached. By separating A.1-9 into its real and imaginary and noting that elements of \overline{YBUS} are known or can be calculated, it can be shown that there are n-1 equations for the real powers and n-1 equations for the reactive power. So, in all there are 2(n-1) equations in 2(n-1) unknowns which are the real and imaginary parts of the bus voltages.

In the previous section the calculation of a set of initial powers was developed given a set of initial bus voltages $\overline{E}_{bus}^{\circ}$. Now by defining voltage correction vector

$$\underline{\Delta E}^{\circ} = [\Delta e_2^{\circ}, \Delta e_3^{\circ}, \dots, \Delta e_n^{\circ}, \Delta f_2^{\circ}, \Delta f_3^{\circ}, \dots, \Delta f]^t \quad A.2-1$$

which added to the $\overline{E}_{bus}^{\circ}$ gives scheduled power. The bus voltage solution vector will contain

$$\overline{E}_{bus}^1 = [e_2^{\circ} + \Delta e_2^{\circ}, e_3^{\circ} + \Delta e_3^{\circ}, \dots, e_n^{\circ} + \Delta e_n^{\circ}, f_2^{\circ} + \Delta f_2^{\circ}, f_3^{\circ} + \Delta f_3^{\circ}, \dots, f^{\circ} + \Delta f]^t$$

A.2-2

Selecting the initial condition bus voltages close enough to the solution, the correction vector will contain small values of Δe and Δf .

The desired power can be expressed in terms of the initial calculated power and the increments of Δe and Δf . Now by applying the Taylor Series expansion to the equation for power, and if the voltage corrections are small, all terms of higher order than first order can be neglected. To illustrate this, A.1-9 is shown with this expansion. The real and imaginary parts have been separated for simplicity.

$$P_p = P_p^\circ + \left. \frac{\partial P_p}{\partial e_2} \right|^\circ \Delta e_2^\circ + \left. \frac{\partial P_p}{\partial e_3} \right|^\circ \Delta e_3^\circ + \dots + \left. \frac{\partial P_p}{\partial e_n} \right|^\circ \Delta e_n^\circ + \left. \frac{\partial P_p}{\partial f_2} \right|^\circ \Delta f_2^\circ + \dots$$

$$\left. \frac{\partial P_p}{\partial f_n} \right|^\circ \Delta f_n^\circ \quad (a)$$

$$Q_p = Q_p^\circ + \left. \frac{\partial Q_p}{\partial e_2} \right|^\circ \Delta e_2^\circ + \dots + \left. \frac{\partial Q_p}{\partial e_n} \right|^\circ \Delta e_n^\circ + \left. \frac{\partial Q_p}{\partial f_2} \right|^\circ \Delta f_2^\circ + \dots \quad (b)$$

$$\left. \frac{\partial Q_p}{\partial f_n} \right|^\circ f_n^\circ \quad A.2-3$$

$$p = 2, 3, \dots, n$$

Taking P_p° and Q_p° terms to the left-hand side of A.2-3 and grouping Δe and Δf terms. A.2-3 can be written in matrix form:

$$\begin{bmatrix} P_2 - P_2^\circ \\ \vdots \\ P_n - P_n^\circ \\ \hline Q_2 - Q_2^\circ \\ \vdots \\ Q_n - Q_n^\circ \end{bmatrix} = \begin{bmatrix} \left. \frac{\partial P_2}{\partial e_2} \right|^\circ & \dots & \left. \frac{\partial P_2}{\partial e_n} \right|^\circ & \left. \frac{\partial P_2}{\partial f_2} \right|^\circ & \dots & \left. \frac{\partial P_2}{\partial f_n} \right|^\circ \\ \vdots & & \vdots & \vdots & & \vdots \\ \left. \frac{\partial P_n}{\partial e_2} \right|^\circ & \dots & \left. \frac{\partial P_n}{\partial e_2} \right|^\circ & \left. \frac{\partial P_n}{\partial f_2} \right|^\circ & \dots & \left. \frac{\partial P_n}{\partial f_n} \right|^\circ \\ \hline \left. \frac{\partial Q_2}{\partial e_2} \right|^\circ & \dots & \left. \frac{\partial Q_2}{\partial e_n} \right|^\circ & \left. \frac{\partial Q_2}{\partial f_2} \right|^\circ & \dots & \left. \frac{\partial Q_2}{\partial f_n} \right|^\circ \\ \vdots & & \vdots & \vdots & & \vdots \\ \left. \frac{\partial Q_n}{\partial e_2} \right|^\circ & \dots & \left. \frac{\partial Q_n}{\partial e_n} \right|^\circ & \left. \frac{\partial Q_n}{\partial f_2} \right|^\circ & \dots & \left. \frac{\partial Q_n}{\partial f_n} \right|^\circ \end{bmatrix} \begin{bmatrix} \Delta e_2^\circ \\ \vdots \\ \Delta e_n^\circ \\ \hline \Delta f_2^\circ \\ \vdots \\ \Delta f_n^\circ \end{bmatrix}$$

programming. Once the complete Jacobian has been calculated, all preliminary calculations have been completed prior to initiating the iterative process.

A.3 The Iterative Process

Once the Jacobian has been calculated, all necessary information is available to perform one iteration. Recalling A.2-5, which is shown here for reference

$$\begin{bmatrix} \Delta P^\circ \\ \Delta Q^\circ \end{bmatrix} = \begin{bmatrix} J_1^\circ & : & J_2^\circ \\ \cdot\cdot & & \cdot\cdot \\ J_3^\circ & : & J_4^\circ \end{bmatrix} \begin{bmatrix} \Delta e^\circ \\ \Delta f^\circ \end{bmatrix} \quad \text{A.3-1}$$

and remembering that the ΔQ portion has some $\Delta|E|^2$ terms contained in it, it can be shown that the A.3-1 can be written as

$$\underline{\Delta S}^\circ = \underline{J}^\circ \cdot \underline{\Delta E}^\circ \quad \text{A.3-2}$$

To solve the equation A.3-2 for the vector $\underline{\Delta E}^\circ$, some form of numerical method can be used. The methods used in this thesis take the inverse of the Jacobian and multiplies equation A.3-2 by the inverse resulting in

$$\underline{J}^{\circ-1} \cdot \underline{\Delta S}^\circ = \underline{\Delta E}^\circ \quad \text{A.3-3}$$

There are numerous techniques to perform this solution. Chapter III gives a more detailed explanation of the two different solution techniques used in this thesis to obtain the corrections to the bus voltages.

Upon obtaining the corrections for the bus voltages, one iteration is complete. The next step is to add these corrections to the bus voltages used in the last iteration using

$$\underline{E}_{bus}^1 = \underline{E}_{bus}^0 + (\underline{\Delta e}_{bus}^0 + j\underline{\Delta f}_{bus}^0) \quad A.3-4$$

Using the corrected bus voltages, the bus currents and power are recalculated using A.1-4 and A.1-9. Again, the differences between the scheduled powers and the values of power calculated are determined by using A.1-13. At voltage control buses, the difference between voltage magnitude scheduled and calculated magnitude is determined using A.1-15. The maximum absolute values of ΔP_{bus}^1 , ΔQ_{bus}^1 and $A|E|_{bus}^1$ are then determined by a sort procedure. The maximum values are checked against the convergence criteria; and if all criteria are satisfied, solution has been reached to the accuracy needed. However, if a convergence criteria is not satisfied, another iteration is required. Again, the Jacobian is calculated using the new corrected values of current and bus voltages. The general equation A.3-3 for the k^{th} iterate is

$$\underline{\Delta S}^k = \underline{J}^k \cdot \underline{\Delta E}^{k+1} \quad A.3-5$$

The numerical solution technique is again used to obtain a new set of Δe 's and Δf 's. The general equation for the solution at the k^{th} iterate is

$$\underline{J}^{k-1} \cdot \underline{\Delta S}^k = \underline{\Delta E}^{k+1} \quad A.3-6$$

This process is repeated until all convergence criteria are satisfied. When convergence is obtained, the program transfers to the line flow section and then to the output section to record the results. The output section of this program is very lengthy and will not be explained.

By partitioning the matrix of partial derivatives which is called the "Jacobian" and noting that the left-hand side of A.2-4 is the vector $\underline{\Delta S^\circ}_{\text{Bus}}$ obtained in Section A.1. A.2-4 can now be written in simplified form as follows

$$\begin{bmatrix} \Delta P^\circ \\ \cdot\cdot \\ \Delta Q^\circ \end{bmatrix} = \begin{bmatrix} J_1^\circ & : & J_2^\circ \\ \cdot\cdot & & \cdot\cdot \\ J_3^\circ & : & J_4^\circ \end{bmatrix} \begin{bmatrix} \Delta e^\circ \\ \cdot\cdot \\ \Delta f^\circ \end{bmatrix} \quad \text{A.2-5}$$

Now the elements of the Jacobian must be determined. The partial derivatives can be determined from equation A.1-9 and A.1-4. The derivation of the partial derivatives will not be included in this paper. These derivations can be found in Elgerd (3) or Stagg and El-Abiad (1). Table VI shows the results of these derivations for all sections of the Jacobian.

The Jacobian represented in A.2-5 does not take the effect of voltage control buses into account. At these buses the ΔQ term is allowed to vary, while $\Delta|E|^2$ must converge to a specific value. So, the Jacobian must be modified somewhat in order to accommodate these voltage control buses. Since only the ΔQ equation will be modified, J_1 and J_2 sections of equation A.2-5 are not affected. At the rows which correspond to voltage control buses in J_3 and J_4 , the ΔQ is replaced by $\Delta|E|^2$. The actual row of the Jacobian is zeroed out, and two diagonal terms are inserted. In the J_3 section the diagonal term is $2 \cdot e_p^\circ$. In the J_4 section the diagonal term is $2 \cdot f_p^\circ$.

With these modifications of the lower part of the $\underline{\Delta S^\circ}_{\text{Bus}}$ vector now contains as many $\Delta|E|^2$ as there are voltage control buses. These terms could be grouped at the bottom of the vector, but it was found to be more efficient to maintain the order of subscripts to allow ease in

TABLE VI

ELEMENTS OF THE JACOBIAN

	Diagonal Terms	Off-Diagonal Terms
J_1	$e_p^\circ G_{pp} - B_{pp} f_p^\circ + c_p^\circ$	$e_p^\circ \cdot G_{pq} - f_p^\circ \cdot B_{pq}$
J_2	$e_p^\circ B_{pp} + f_p^\circ \cdot G_{pp} + d_p^\circ$	$e_p^\circ \cdot B_{pq} + f_p^\circ \cdot G_{pq}$
J_3	$e_p^\circ \cdot B_{pp} + f_p^\circ \cdot G_{pp} - d_p^\circ$	$e_p^\circ \cdot B_{pq} + f_p^\circ \cdot G_{pq}$
J_4	$- e_p^\circ \cdot G_{pp} + f_p^\circ \cdot B_{pq} + c_p^\circ$	$- e_p^\circ \cdot G_{pq} + f_p^\circ \cdot B_{pq}$

A.4 Calculation of Line Flows

In a load-flow analysis bus voltages and slack bus power are not the only information that can be calculated. Once bus voltages and slack bus power are calculated, the actual power flowing in each transmission line can be calculated. This information determines whether the capacity of the line has been exceeded. Also, transmission line losses are important to show the relative efficiency of each line. This section of the load flow problem is called line flow. To calculate line flow and line losses, the following representation is used. Power flowing into one end of the line is calculated along with the power flowing into the other end of the line. The algebraic sum of these two quantities is the line loss.

In Figure A.5, a π equivalent of a transmission line between bus i and j is shown. First calculate the current flowing into the line at bus i.

$$\bar{I}_i = \bar{I}_{ij} + \bar{I}_{SHi} \quad A.4-1$$

Series line current is

$$\bar{I}_{ij} = (\bar{V}_i - \bar{V}_j) \cdot \bar{Y}_{ij} \quad A.4-2$$

And the shunt current is

$$\bar{I}_{SHi} = \bar{V}_i \cdot \bar{Y}_{SHi} \quad A.4-3$$

By substituting A.4-2 and A.4-3 into A.4-1, the current flowing into the line at bus i is

$$\bar{I}_i = (\bar{V}_i - \bar{V}_j) \cdot \bar{Y}_{ij} + \bar{V}_i \cdot \bar{Y}_{SHi} \quad A.4-4$$

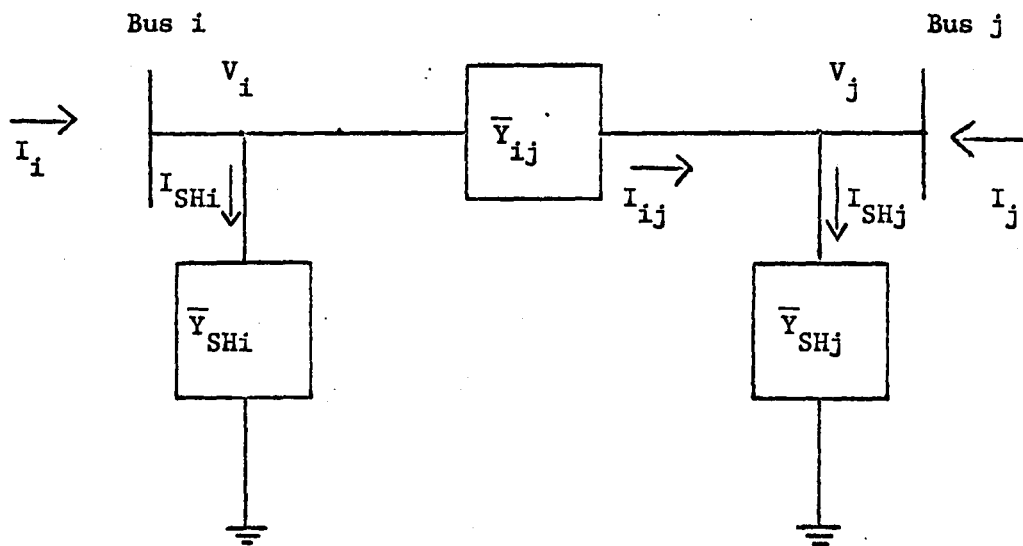


Figure A.2 - Transmission Line π Equivalent

Power flowing into the line at bus i is

$$\overline{S}_i = \overline{V}_i \cdot \overline{I}_i^* \quad \text{A.4-5}$$

Substituting A.4-4 into A.4-5, the resulting power entering the line at bus i is

$$\overline{S}_i = \overline{V}_i \cdot [(\overline{V}_i - \overline{V}_j) \cdot \overline{Y}_{ij} + \overline{V}_i \cdot \overline{Y}_{SHi}]^* \quad \text{A.4-6}$$

To obtain power flowing into the line at bus j, the subscripts of equation A.4-6 are interchanged, and the equation for power entering at bus j is

$$\overline{S}_j = \overline{V}_j \cdot [(\overline{V}_j - \overline{V}_i) \cdot \overline{Y}_{ji} + \overline{V}_j \cdot \overline{Y}_{SHj}]^* \quad \text{A.4-7}$$

To arrive at the value of the losses of this transmission line, the algebraic sum of the two powers is calculated using

$$\overline{\text{Loss}}_{ij} = \overline{S}_i + \overline{S}_j \quad \text{A.4-8}$$

A.4-6 through A.4-8 can be applied to each transmission line in the system. Summing the $\overline{\text{Loss}}_{ij}$ for each line, the total system transmission losses can be obtained.

**The vita has been removed from
the scanned document**

DIGITAL LOAD FLOW ANALYSIS
USING GENERAL MATRIX STORAGE AND
NON-ZERO ELEMENT STORAGE

by

William M. Heldreth

(ABSTRACT)

This thesis deals with the AC Load Flow Problem and the comparison of two different solution techniques. The basic solution used was the Newton Raphson method. The two methods used to solve for the corrections to the bus voltages are general matrix storage and non-zero element storage. In the general matrix all matrices are used with every element represented. In the non-zero routine, only non-zero entries in the YBUS and the Jacobian matrix are stored. Along with the non-zero routine, a form of sub-optimal ordering was used to make the inversion process faster. The execution times and storage requirements for seven examples are given. These results show the advantages of using a non-zero storage technique with respect to execution time as well as storage requirements.