Power System Analysis Suite for Windows

by

Steven Douglas Estes

Thesis submitted to the 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:

AGPhadke
Dr. A. Phadke, Chairman

Jaime de la Ree Lopez
Dr. J. De La Ree Lopez

SRahman
Dr. S. Rahman

April, 1996
Blacksburg, Virginia

Key Words: State Estimation, Load Flow, Power System

Power System Analysis Suite for Windows

by

Steven Douglas Estes

Committee Chairman: Dr. A. Phadke
Bradley Department of Electrical Engineering

(ABSTRACT)

The ability to analyze a power system is essential to power system engineers and planners. The Bus program, a Microsoft Windows-based program, helps users make these analyses. Unlike other power system analysis programs, the Bus program performs three different types of analyses (short circuit, load flow, and state estimation) and offers users a graphical interface on which to enter their system and data. This thesis presents the Bus program and discusses various aspects of it, focusing on the load flow and state estimation routines, which were the main thrust of the project. Each of these routines was written by setting up a flowchart and defining the calculations to be carried out. Vehicles were then developed so that users can enter system data and view the results of the calculations. The ability to do this graphically is one of the main features of the program. Several test cases are presented to demonstrate the program's operation, and a User's Manual is included to show users how to operate the program.

ACKNOWLEDGEMENTS

I would like to acknowledge and thank the following people: my parents, for their love and support throughout my education; to my committee members, Drs. Phadke, De La Ree, and Rahman, for their knowledge, help, and guidance; and Mr. David Todd, for his timely computer assistance.

Table of Contents

Table of Figures

Table of Tables

CHAPTER 1
INTRODUCTION

In order to maintain safe and reliable operation of a power system, it is often very important that various analyses be performed on the system. These calculations help power system operators and planners in designing the system and preventing problems in the system.

Three such analyses are short circuit, load flow, and state estimation studies. Short circuit calculations, which calculate the currents and voltages during a fault to the system, are indispensible to those designing a protection scheme for the system. A load flow analysis, where the power flows and voltages throughout the system are calculated, can show planners the performance of a system without physically building it. A state estimation is very similar to a load flow analysis except actual system data is used. Data from thoughout the system is used to determine the "state" (voltages and angles) of the system, while also serving to identify "bad" measurement units. In addition to their usefulness in actual practice, these tools can be very helpful in an academic setting. Many times students, who do not have access to a physical system, learn about power system by performing these analyses.

Many programs have been written to perform these computations. However, because many of them were written for MS-DOS or other platforms, entering the system data can be a time-consuming and/or confusing process. Many of these programs only perform one type of analysis, meaning that this same data may have to be entered more than once. The bus program, which is the topic of this thesis, seeks to alleviate these problems. By providing a graphical, Windows-based interface, users are allowed to actually "draw" their system. This makes data entry and visualization much simpler, quicker, and more organized. The program performs all three analyses mentioned above on the system entered. Because of the nature of the program, only a few minor modifications of the data are necessary between analyses. This allows for a more flexible, and user-friendly method of performing the necessary analyses.

In the program interface, users can place buses (up to 75), lines (up to 75, including transformers), and transformers wherever they please. Thus, a system can be visually recreated as on paper. The system parameters may then be entered by clicking on the respective component. Details of this process, and the data to be entered is contained in the User's Manual (Appendix B). Once the parameters and other data (for example, measurements for state estimation) has been entered, the desired calculations can be performed.

At this point, the program creates the files and arrays necessary for the algorithms, and then runs MATLAB. Within MATLAB, the user can run the analysis, view the results, and inspect any particular variable. While using MATLAB sacrifices computational speed, the ease with which algorithms can be written and the control the user has over the process are important advantages over a normal programming language. When finished with the computations, the user can must quit MATLAB, at which time, the Bus program reads in the MATLAB results for inspection by the user.

The Bus program was originally written by Chris Howard. The original version included the interface and drawing features, as well as the short circuit calculations. The task of this thesis project was to correct any errors in the short circuit portion and expand this program to perform load flow studies and state estimations calculations. This involved adding data entry processes and dialogs, as well as the calculation routines themselves.

Chapter 2 of this thesis will present the algorithms for each of the analyses as well as the calculations involved in them. Chapter 3 discusses considerations (both in the interface and MATLAB portions) taken in development of the program, while Chapter 4 will present some examples of each analysis.

CHAPTER 2

ALGORITHMS AND CALCULATIONS

The calculation routines for this program were written by first creating a flowchart for the algorithm. This helped establish what calculations were necessary and when they were to be made. From here, the routines were written using the necessary equations (or sets of equations). Once finished, the routines were tested using test examples from [1] and [3]. Here, bugs and other problems in the routines were worked out, producing what are hopefully error free algorithms. This chapter presents the flowcharts for each analysis, and more importantly, the calculations and equations used in each of them.

2.1 Short Circuit

In order to protect a power system during a fault to it, it is vital that the system engineers know the condition of the system during the fault. Short circuit calculations are carried out for this purpose. These calculations determine the voltages at each bus and currents flowing through each line during a fault. These results allow protection engineers to set the relays and circuit breakers so that the system is protected in the event of a fault.

2.1.1 Method

Because this program performs both 3-phase and phase-to-ground fault calculations, there are two different algorithms, although they are somewhat similar. Since the equations involved are linear, no iterative procedure is necessary. This leads to a rather quick computation time for both calculations since they can be carried out with matrix algebra. The calculations are outlined in the flowcharts following, and are discussed in the next section.

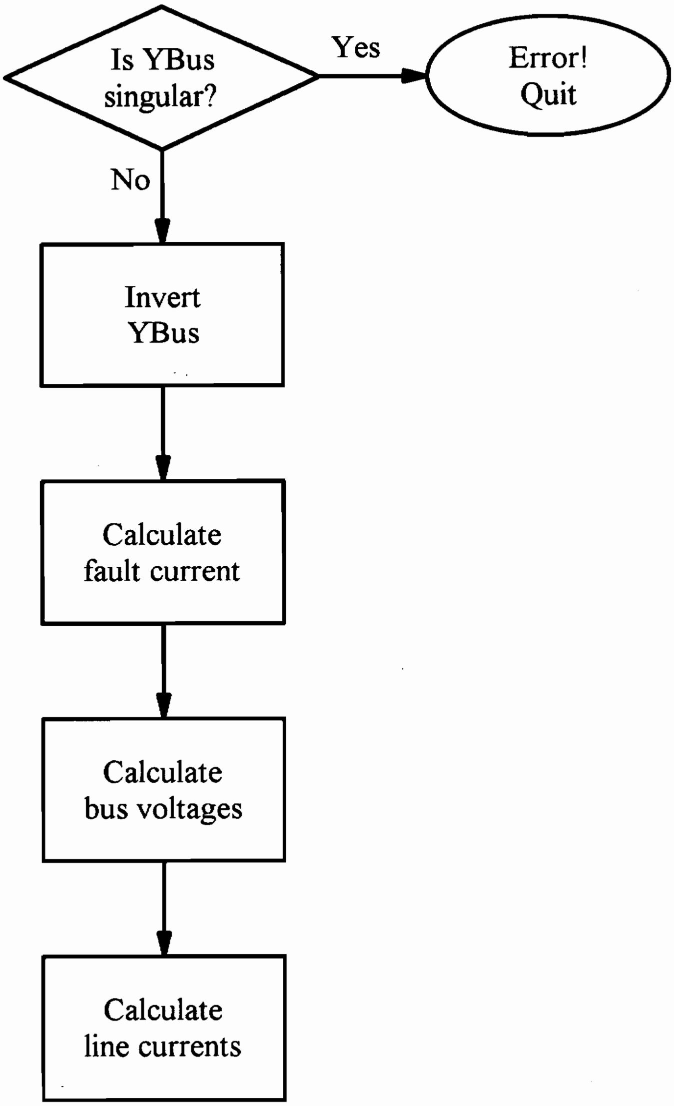


Figure 2.1 Flowchart for Three-Phase Short Circuit Calculations

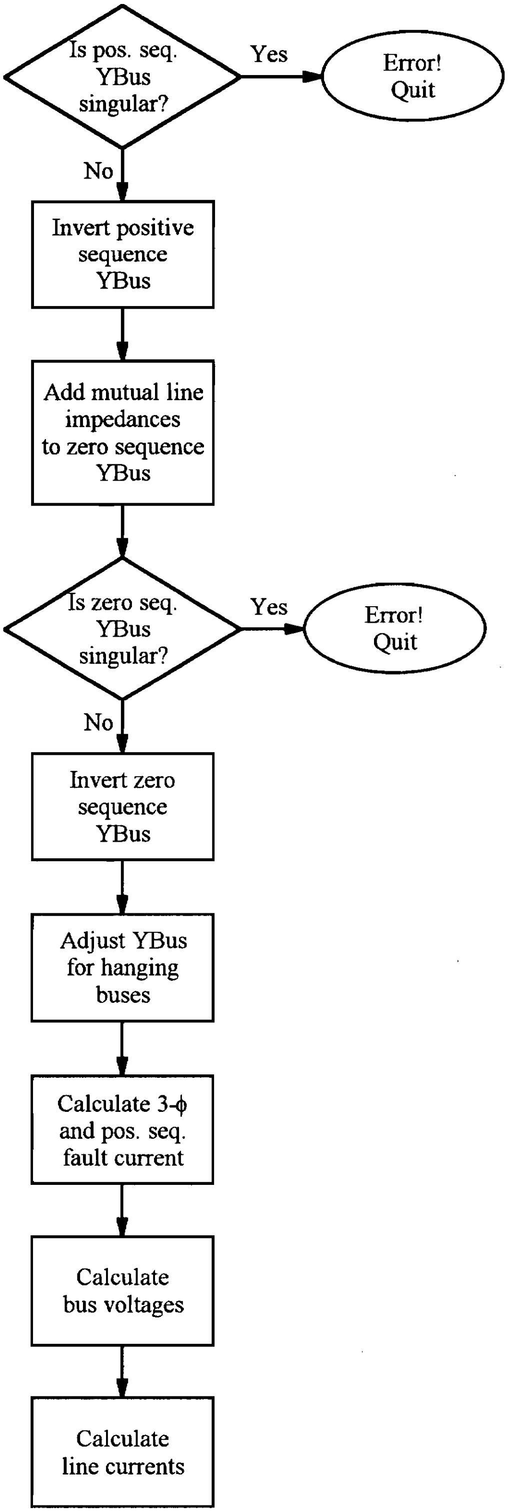


Figure 2.2 Flowchart for Phase-to-Ground Short Circuit Calculations

2.1.2 Calculations

In order to perform the short circuit calculations, the user need only have entered the system impedances and a fault bus. For phase-ground calculations, the user must also enter zero sequence data.

Both routines use the positive-sequence admittance matrix ($\mathbf{Y}_{\mathbf{busp}}$) and the phase-ground analysis uses the zero-sequence admittance matrix ($\mathbf{Y}_{\mathbf{busz}}$) as well. More details on the admittance matrices can be found in Appendix A. A derivation of the calculations used here is available in [2], as well as others.

The 3-phase short circuit calculation starts out by first checking to make sure that $\mathbf{Y}_{\mathbf{bus}}$ isn't singular. If it is, an error is posted and the routine is stopped. The next step is to invert $\mathbf{Y}_{\mathbf{bus}}$ to get $\mathbf{Z}_{\mathbf{bus}}$ ($\mathbf{Z}_{\mathbf{bus}} = \mathbf{Y}_{\mathbf{bus}}{}^{-1}$). Once this is found, the fault current is calculated using Ohm's Law as shown:

$$\mathrm{I}_f = \frac{1.0}{\mathbf{Z}_{\mathbf{bus}_{ff}}} \tag{2.1}$$

where 1.0 is the prefault voltage and $\mathbf{Z}_{\mathbf{busff}}$ is the impedance at the fault bus.

Once the fault current has been found, the voltages at each bus may be calculated using the following equation:

$$\mathrm{V}_i = 1.0 - \mathrm{Z}_{bus_{if}} * \mathrm{I}_f \tag{2.2}$$

with the fault bus voltage set to 0. This equation results from adding (subtracting) the voltage drop due to the fault current from the prefault voltage at the fault bus (1.0).

Once the bus voltages are known, the line currents can be calculated using Ohm's Law. The current in the line is calculated by simply multiplying the voltage drop across the line by the line admittance as follows:

$$I_{ij} = \left(V_i - V_j\right) * Y_{busp_{ij}} \tag{2.3}$$

This is all that is computed in the three-phase analysis.

If a phase-ground analysis has been chosen, the short circuit calculations change somewhat. After checking the singularity of $\mathbf{Y_{busp}}$, it is inverted to get $\mathbf{Z_{busp}}$. The zero-sequence admittance matrix is then computed from

$$\mathbf{Y_{busz}} = \mathbf{Y_{busz}} + \mathbf{Mutual}_{real} + \mathbf{Mutual}_{imag} \tag{2.4}$$

where **Mutual** is the matrix containing the mutual impedances between lines. This $\mathbf{Y_{busz}}$ is inverted as well, giving $\mathbf{Z_{busz}}$, the zero-sequence impedance matrix.

After all this is done, the 3-phase fault current is calculated by dividing the prefault voltage by the sums of the positive, negative, and zero-sequence impedances. Since the positive and negative-sequence impedances are equal, $\mathbf{Z_{busp}}$ is multiplied by 2. The equation used is as follows:

$$I_f = \frac{3.0}{2 * Z_{busp_{ff}} + Z_{busz_{ff}}} \tag{2.5}$$

and the positive-sequence phase a current is found by

$$I_{a1} = \frac{I_f}{3.0} \tag{2.6}$$

After the current has been found, the next step is to calculate the bus voltages. This is done by adding the voltage drop due to the fault current to the prefault voltage (which is 0 in negative and zero-sequence networks) as follows:

$$\begin{aligned} V_{a1_i} &= 1.0 - I_{a1} * Z_{busp_{i,f}} \\ V_{a2_i} &= -I_{a1} * Z_{busp_{i,f}} \\ V_{a0_i} &= -I_{a1} * Z_{busz_{i,f}} \end{aligned} \tag{2.7}$$

where i is the bus in question, I_{a1} is the positive-sequence fault current ($=I_{a0}=I_{a2}$) and f is the fault bus. From here, the phase voltages can be computed.

$$\begin{aligned} V_{A_i} &= V_{a1_i} + V_{a2_i} + V_{a0_i} \\ V_{B_i} &= a^2 V_{b1_i} + a V_{b2_i} + V_{b0_i} \\ V_{C_i} &= a V_{c1_i} + a^2 V_{c2_i} + V_{c0_i} \end{aligned} \qquad (2.8)$$

where a = -.5 + j.866 = 1∠120° and a2 = -.5 - j.866 = 1∠240°. These calculations are derived in [2].

Once the bus voltages have been found, the line currents must be calculated. This is done just like the three-phase current calculations:

$$\begin{aligned} I_{a1_{ij}} &= \left(V_{a1_i} - V_{a1_j}\right) * Y_{busp_{ij}} \\ I_{a2_{ij}} &= \left(V_{a2_i} - V_{a2_j}\right) * Y_{busp_{ij}} \\ I_{a0_{ij}} &= \left(V_{a0_i} - V_{a0_j}\right) * Y_{busz_{ij}} \\ I_{A_{ij}} &= \left(V_{A_i} - V_{A_j}\right) * Y_{busp_{ij}} \\ I_{B_{ij}} &= \left(V_{B_i} - V_{B_j}\right) * Y_{busp_{ij}} \\ I_{C_{ij}} &= \left(V_{C_i} - V_{C_j}\right) * Y_{busz_{ij}} \end{aligned} \qquad (2.9)$$

This concludes the phase-ground short circuit calculations.

After the short circuit (either 3-phase or phase-ground) analysis has been completed, the $\mathbf{Z}_{\mathbf{bus}}$ matrix, current, and voltage results are saved and imported back into the bus program. At this point they are read in and the voltage magnitudes and angles calculated. The voltages and currents can be displayed on the system.

2.2. Load Flow

A load flow study involves the determination of voltage magnitudes, angles, power generation or load at each bus, and power flows (real and reactive) on each line of a system, given a model of that system. These studies are very important, as they are usually used to help planners determine the effects of future expansion of the system. They can also be used to look at operating problems within the system, which is helpful to both planners and system operators.

2.2.1 Method

Because the power flow equations used are non-linear, obtaining a load flow solution is often a complex problem. It involves finding a solution to a set of non-linear equations. While derivation of these equations is not difficult, obtaining a closed-form solution to solve them by is [2]. Thus, iterative procedures are often used. For this project, the Newton-Raphson method was used. This method is detailed in many books, so any treatment of it will be limited to the flowchart and the calculations. While Newton-Raphson requires more calculations per iteration, it converges much quicker than other methods. This makes it a more desired routine, especially for systems larger than a few buses [2]. Figure 1 shows the flowchart for this method and the load-flow algorithm in general. The next section will go through the calculations and steps involved in solving the load flow.

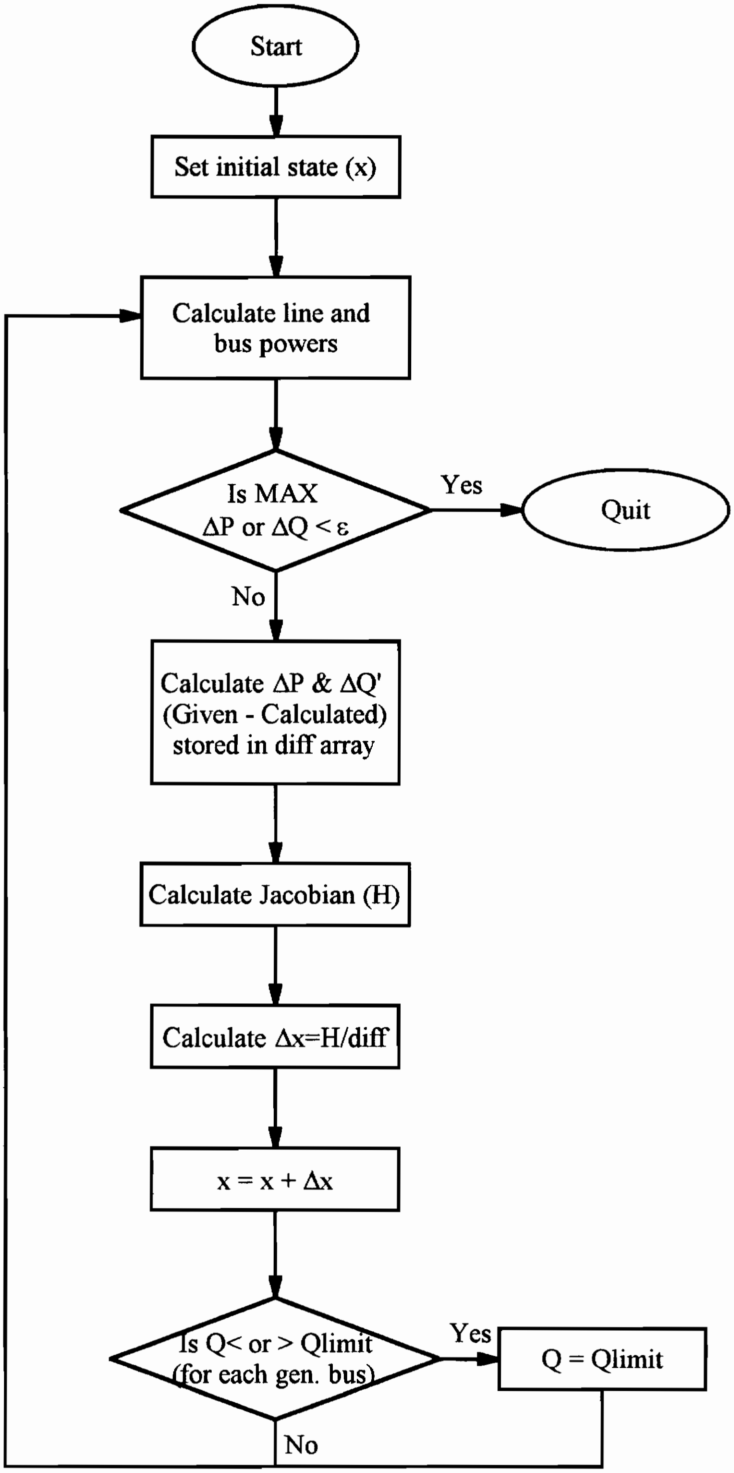


Figure 2.3 Flowchart for Load Flow Algorithm (based on Figure 4.7 from [1])

2.2.2 Calculations

In order to carry out the calculations, the following data must be entered into the program: system parameters (impedances, etc.), generated and load powers at buses, initial state (**X**), and error tolerance. The state vector contains the voltage magnitudes and angles for each bus in the system.

Because this algorithm relies on the difference between the calculated and input powers in determining when a solution has been reached, the first step in the routine is to calculate the injected powers at each bus (except the swing bus) using the state vector, **X**, and the system parameters. This is accomplished by summing the line flows at each bus.

The following formulas are used:

$$P_i = \sum_{j=1}^{n} p_{ij} + P_{shunt_i} \qquad\qquad Q_i = \sum_{j=1}^{n} q_{ij} + Q_{shunt_i} \tag{2.10}$$

where P_i and Q_i are the real and reactive bus powers, n is the number of buses in the system, and

$$p_{ij} = |E_i|^2 G_{ij} - |E_i||E_j|(\cos(\Theta_i - \Theta_j)G_{ij} + \sin(\Theta_i - \Theta_j)B_{ij}$$

$$q_{ij} = -|E_i|^2 \left(B_{cap_{ij}} + B_{ij}\right) - |E_i||E_j|\left(\sin(\Theta_i - \Theta_j)G_{ij} - \cos(\Theta_i - \Theta_j)B_{ij}\right) \tag{2.11}$$

$$P_{shunt_i} = -|E_i|^2 G_{shunt}$$

$$Q_{shunt_i} = |E_i|^2 B_{shunt} \tag{2.12}$$

In these equations, p_{ij} and q_{ij} are the line flows and P_{shunt_i} and Q_{shunt_i} are the power contributions from any shunts connected to the bus. $|E_i|$ and Θ_i are the voltage magnitude and voltage angles, respectively, at the buses (the state variables). G_{ij} and B_{ij} are the real and imaginary parts of the line admittance, while $B_{cap_{ij}}$ is the charging capacitance of the line divided by 2 [1]. G_{shunt_i} and B_{shunt_i} are the real and imaginary parts of the shunt admittance at the bus. For more information on these parameters, see Appendix A.

These calculated powers are subtracted from the given bus powers (entered into the program) and stored in an array (**diff**) to be used later. These are the ΔP's and ΔQ's. For generator buses that are voltage controlled (this is assumed), the reactive power (Q) calculation is omitted.

At this point, the largest ΔP or ΔQ is compared to the error tolerance entered by the user. If it is less, then the iterations have converged. The routine is exited and the final results (powers) are computed. Otherwise, the routine continues with the calculation of the Jacobian.

The Jacobian matrix (**H**) relates the **ΔX** (change in state) vector to the ΔP's and ΔQ's as follows:

$$\underbrace{\begin{bmatrix} \Delta P_1 \\ \Delta Q_1 \\ \Delta P_2 \\ \Delta Q_2 \\ \vdots \end{bmatrix}}_{\text{diff}} = \underbrace{\begin{bmatrix} \frac{\partial P_1}{\partial \Theta_1} & \frac{\partial P_1}{\partial E_1} & \frac{\partial P_1}{\partial \Theta_2} & \frac{\partial P_1}{\partial E_2} & \cdots \\ \frac{\partial Q_1}{\partial \Theta_1} & \frac{\partial Q_1}{\partial E_1} & \frac{\partial Q_1}{\partial \Theta_2} & \frac{\partial Q_1}{\partial E_2} & \cdots \\ \frac{\partial P_2}{\partial \Theta_1} & \frac{\partial P_2}{\partial E_1} & \frac{\partial P_2}{\partial \Theta_2} & \frac{\partial P_2}{\partial E_2} & \cdots \\ \frac{\partial Q_2}{\partial \Theta_1} & \frac{\partial Q_2}{\partial E_1} & \frac{\partial Q_2}{\partial \Theta_2} & \frac{\partial Q_2}{\partial E_2} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}}_{H} \underbrace{\begin{bmatrix} \Delta \Theta_1 \\ \Delta |E_1| \\ \Delta \Theta_2 \\ \Delta E_2 \\ \vdots \end{bmatrix}}_{\Delta X} \tag{2.13}$$

This relationship can be derived by taking a set of equations with $x_i + \Delta x_i$ as the solution and expanding them into a Taylor's Series [2].

The rows and columns corresponding to the swing bus are omitted from the Jacobian (**H**). For generator buses that are voltage controlled, the row corresponding to the reactive power (Q) and the column corresponding to the voltage magnitude |E| of the bus are also omitted [2]. The elements of the Jacobian are computed as follows:

$$
\begin{aligned}
\frac{\partial \mathrm{P}_i}{\partial \mathrm{E}_j} &= \sum_{k=1}^{n} \frac{\partial \mathrm{p}_{ik}}{\partial \mathrm{E}_j} + \frac{\partial \mathrm{P}_{shunt_i}}{\partial \mathrm{E}_j} &\qquad \frac{\partial \mathrm{P}_i}{\partial \Theta_j} &= \sum_{k=1}^{n} \frac{\partial \mathrm{p}_{ik}}{\partial \Theta_j} \\
\frac{\partial \mathrm{Q}_i}{\partial \mathrm{E}_j} &= \sum_{k=1}^{n} \frac{\partial \mathrm{q}_{ik}}{\partial \mathrm{E}_j} + \frac{\partial \mathrm{Q}_{shunt_i}}{\partial \mathrm{E}_j} &\qquad \frac{\partial \mathrm{Q}_i}{\partial \Theta_j} &= \sum_{k=1}^{n} \frac{\partial \mathrm{q}_{ik}}{\partial \Theta_j}
\end{aligned}
\tag{2.14}
$$

where the partial derivatives for the line and shunt flows are as follows:

$$
\begin{aligned}
\frac{\partial \mathrm{p}_{ik}}{\partial \mathrm{E}_j} &= 2|\mathrm{E}_i|\mathrm{G}_{ik} - |\mathrm{E}_k|\left(\cos(\Theta_i - \Theta_k)\mathrm{G}_{ik} + \sin(\Theta_i - \Theta_k)\mathrm{B}_{ik}\right) && \text{if } \mathrm{i} = \mathrm{j} \\
&= -|\mathrm{E}_k|\left(\cos(\Theta_i - \Theta_k)\mathrm{G}_{ik} + \sin(\Theta_i - \Theta_k)\mathrm{B}_{ik}\right) && \text{if } \mathrm{k} = \mathrm{j} \\[2ex]
\frac{\partial \mathrm{p}_{ik}}{\partial \Theta_j} &= -|\mathrm{E}_i||\mathrm{E}_k|\left(-\sin(\Theta_i - \Theta_k)\mathrm{G}_{ik} + \cos(\Theta_i - \Theta_k)\mathrm{B}_{ik}\right) && \text{if } \mathrm{i} = \mathrm{j} \\
&= -|\mathrm{E}_i||\mathrm{E}_k|\left(\sin(\Theta_i - \Theta_k)\mathrm{G}_{ik} - \cos(\Theta_i - \Theta_k)\mathrm{B}_{ik}\right) && \text{if } \mathrm{k} = \mathrm{j} \\[2ex]
\frac{\partial \mathrm{q}_{ik}}{\partial \mathrm{E}_j} &= -2|\mathrm{E}_i|\left(\mathrm{B}_{cap_{ik}} + \mathrm{B}_{ik}\right) - |\mathrm{E}_k|\left(\sin(\Theta_i - \Theta_k)\mathrm{G}_{ik} - \cos(\Theta_i - \Theta_k)\mathrm{B}_{ik}\right) && \text{if } \mathrm{i} = \mathrm{j} \\
&= -|\mathrm{E}_i|\left(\sin(\Theta_i - \Theta_k)\mathrm{G}_{ik} - \cos(\Theta_i - \Theta_k)\mathrm{B}_{ik}\right) && \text{if } \mathrm{k} = \mathrm{j} \\[2ex]
\frac{\partial \mathrm{q}_{ik}}{\partial \Theta_j} &= -|\mathrm{E}_i||\mathrm{E}_k|\left(\cos(\Theta_i - \Theta_k)\mathrm{G}_{ik} + \sin(\Theta_i - \Theta_k)\mathrm{B}_{ik}\right) && \text{if } \mathrm{i} = \mathrm{j} \\
&= -|\mathrm{E}_i||\mathrm{E}_k|\left(-\cos(\Theta_i - \Theta_k)\mathrm{G}_{ik} - \sin(\Theta_i - \Theta_k)\mathrm{B}_{ik}\right) && \text{if } \mathrm{k} = \mathrm{j}
\end{aligned}
\tag{2.15}
$$

$$
\begin{aligned}
\frac{\partial \mathrm{P}_{shunt_i}}{\partial \mathrm{E}_j} &= -2|\mathrm{E}_i|\mathrm{G}_{shunt_i} && \text{if } \mathrm{i} = \mathrm{j} \\
&= 0 && \text{if } \mathrm{i} \neq \mathrm{j}
\end{aligned}
\qquad
\frac{\partial \mathrm{P}_{shunt_i}}{\partial \Theta_i} = 0 \qquad \text{for all j}
$$

$$\frac{\partial Q_{shunt_i}}{\partial E_j} = 2|E_i|B_{shunt_i} \qquad \text{if } i = j$$

$$= 0 \qquad \text{if } i \neq j$$

$$\frac{\partial Q_{shunt_i}}{\partial \Theta_i} = 0 \qquad \text{for all } j$$

$$\frac{\partial p_{ik}}{\partial E_j} = \frac{\partial p_{ik}}{\partial \Theta_j} = \frac{\partial q_{ik}}{\partial E_j} = \frac{\partial q_{ik}}{\partial \Theta_j} = 0 \qquad \text{if } j \neq i \text{ or } k \tag{2.16}$$

These equations can be derived by hand from (2.10) through (2.12). The same parameters are applied here as well.

Once the Jacobian has been calculated the state difference **ΔX** is calculated from

$$\mathbf{\Delta X} = \mathbf{H} / \mathbf{diff} \tag{2.17}$$

which is simply solving a set of equations. Remember that this does not include the swing bus and the reactive power (Q) at generator buses. These must be set to 0 in the **ΔX** vector. This **ΔX** vector is then added to the current state to get the new state vector (**X** = **X** + **ΔX**).

Following this, the reactive power at each generator bus is computed, and compared to the generator limits set by the user. If this calculated Q is within the limits, nothing is done at this stage. If not, the closest limit is set as the generated reactive power and the bus is treated as a regular bus (i.e. NOT voltage controlled).

After this step, the routine loops back to the calculation of the bus powers from the new state vector and continues on.

NOTE: The procedures listed above are found in [1], [2], and various other references.

2.3 State Estimation

A state estimation solution involves finding the state (voltage at each bus and angle at all except the swing bus) of a system given voltage and power measurements made on the system. Often, this estimation provides a more accurate representation of the system conditions than the measurements do, since measurement errors are taken into consideration and "bad" measurements are identified and removed during the process. For these reasons, and the fact that the entire system can be viewed from a few measurements, state estimation can be a very valuable tool to system operators.

2.3.1 Method

The method used to perform the state estimation is similar to that of the load flow. Not only are the power flow equations used the same, but the iterative procedures used to reach a solution are similar. Both routines use Newton's method to converge to a solution, although state estimation uses a different convergence criterion.

Since the measurements made on the system have an error associated with them, the estimate will also (it will not reach the "true" value). Thus, it is possible to arrive at several different estimates using the given data. In order to determine which is the best one, a statistical criterion must be used [1]. Once the criterion has been set, the estimator equations are derived so that it will always be followed. The criterion used to develop this estimator is the maximum likelihood criterion, which maximizes the probability that the state estimate equals the "true" value of the state. [1]

Since the measurement errors are normally-distributed random variables, this results in a weighted least-squares estimator. In fact, with this condition, all of the criterion result in this particular estimator. This estimator algorithm attempts to minimize the sum of the squares of the measurement residuals (differences between estimated and actual values). These differences are weighted with respect to the error variance of the measurement. This ensures that measurements made by more accurate instruments are weighted heavier than others. A derivation of the estimator can be found in [1] among other books. [1]

Because many more calculations are involved than with load flow studies, computation time will be somewhat longer. However, the use of Newton's Method gives a quick convergence to the routine. Figure 3 shows the flowchart for the state estimator. The calculations and equations involved are discussed in the next section.

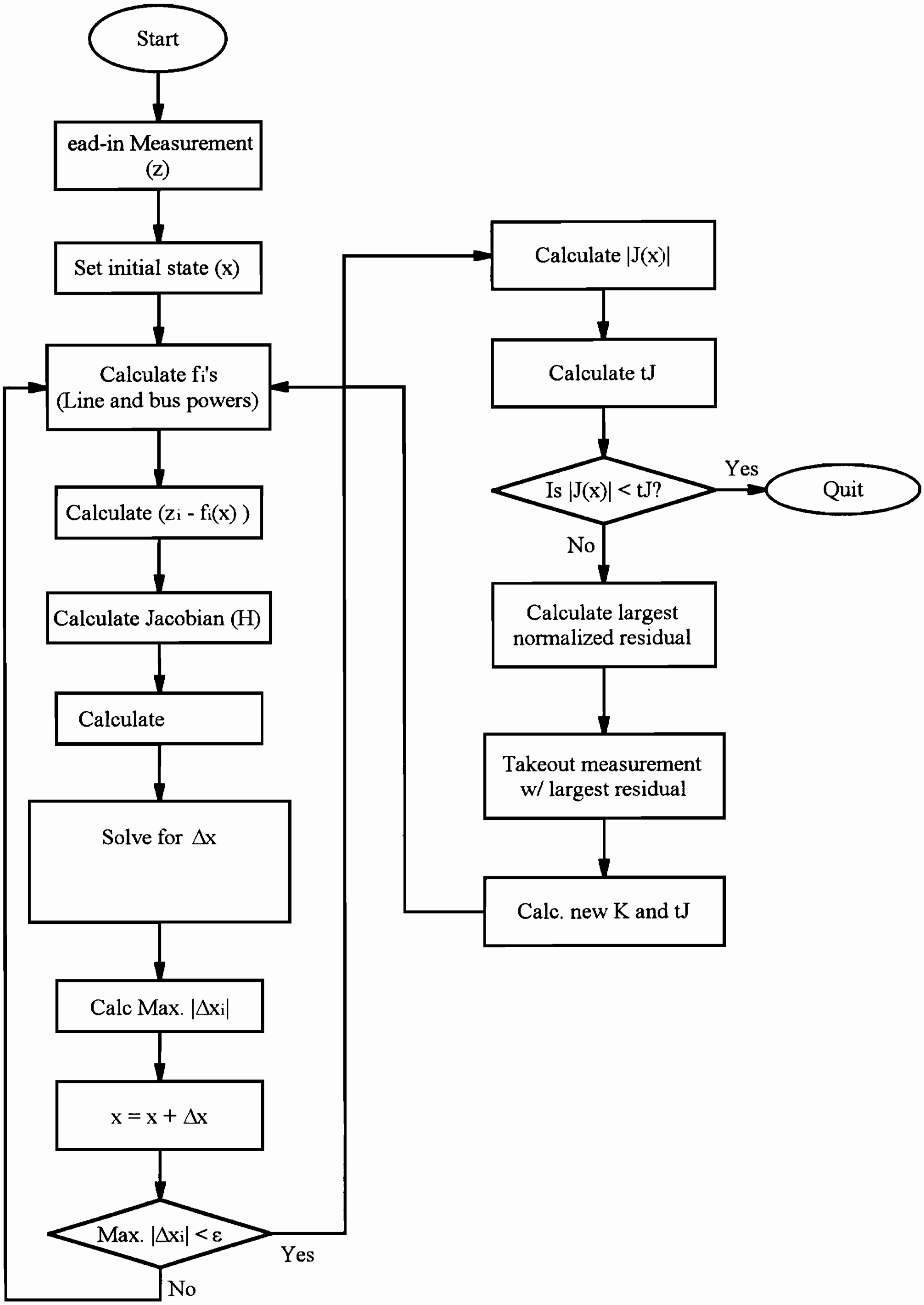


Figure 2.3 Flowchart for State Estimation Algorithm (based on Fig. 12.11 of [1])

2.3.2 Calculations

In order to perform the state estimation, the user must have entered the following data: system parameters, initial state (**X**), error tolerance, measurements (**z**), and measurement error variances.

Any combination of line power flows, generator or load powers, or voltage magnitude measurements may be made, as long as there are enough to observe the system. An error variance is required for each measurement.

After reading in the data, the first step in the routine is to calculate (using the state vector) each of the values being measured (f_i's). The equations used follow. They are the same as the load flow equations, although now the equation used for f_i depends on the type of measurement.

For bus power measurements (either real or reactive):

$$P_i = \sum_{j=1}^{n} p_{ij} + P_{shunt_i}$$

$$Q_i = \sum_{j=1}^{n} q_{ij} + Q_{shunt_i} \qquad (2.18)$$

where n is the number of buses in the system and

$$p_{ij} = |E_i|^2 G_{ij} - |E_i||E_j|\left(\cos(\Theta_i - \Theta_j)G_{ij} + \sin(\Theta_i - \Theta_j)B_{ij}\right.$$

$$q_{ij} = -|E_i|^2\left(B_{cap_{ij}} + B_{ij}\right) - |E_i||E_j|\left(\sin(\Theta_i - \Theta_j)G_{ij} - \cos(\Theta_i - \Theta_j)B_{ij}\right) \qquad (2.19)$$

$$P_{shunt_i} = -|E_i|G_{shunt}$$

$$Q_{shunt_i} = |E_i|B_{shunt} \qquad (2.20)$$

For line power flow measurements, the equations for either p_{ij} or q_{ij} would be used. For a voltage magnitude measurement, the calculated f_i is simply the respective voltage magnitude from the state vector.

Each f_i is then subtracted from its corresponding measurement (z_i). This forms an array of non-weighted measurement residuals $[z_i - f_i(x)]$. After this, the Jacobian (**H**) is calculated. It is similar to the load flow Jacobian except that there is a row for each measurement and a column for each state variable. This means that **H** will not be square. The Jacobian is defined as:

$$H = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots \\ \vdots & \vdots & \ddots \\ & & \end{bmatrix} \tag{2.21}$$

where the f_i's correspond to each measurement and x_i's are the state variables. [1] Depending on the type of measurement and the specific state variable, the elements of the Jacobian are computed from one of the following equations.

For buses:

$$\frac{\partial P_i}{\partial E_j} = \sum_{k=1}^{n} \frac{\partial p_{ik}}{\partial E_j} + \frac{\partial P_{shunt_i}}{\partial E_j} \qquad \frac{\partial P_i}{\partial \Theta_j} = \sum_{k=1}^{n} \frac{\partial p_{ik}}{\partial \Theta_j}$$

$$\frac{\partial Q_i}{\partial E_j} = \sum_{k=1}^{n} \frac{\partial q_{ik}}{\partial E_j} + \frac{\partial Q_{shunt_i}}{\partial E_j} \qquad \frac{\partial Q_i}{\partial \Theta_j} = \sum_{k=1}^{n} \frac{\partial q_{ik}}{\partial \Theta_j} \tag{2.22}$$

The partial derivatives for the line flows are as follows:

$$
\begin{aligned}
\frac{\partial p_{ik}}{\partial E_j} &= 2|E_i|G_{ik} - |E_k|\left(\cos(\Theta_i - \Theta_k)G_{ik} + \sin(\Theta_i - \Theta_k)B_{ik}\right) && \text{if } i = j \\
&= -|E_k|\left(\cos(\Theta_i - \Theta_k)G_{ik} + \sin(\Theta_i - \Theta_k)B_{ik}\right) && \text{if } k = j
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial p_{ik}}{\partial \Theta_j} &= -|E_i||E_k|\left(-\sin(\Theta_i - \Theta_k)G_{ik} + \cos(\Theta_i - \Theta_k)B_{ik}\right) && \text{if } i = j \\
&= -|E_i||E_k|\left(\sin(\Theta_i - \Theta_k)G_{ik} - \cos(\Theta_i - \Theta_k)B_{ik}\right) && \text{if } k = j
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial q_{ik}}{\partial E_j} &= -2|E_i|\left(B_{cap_{ik}} + B_{ik}\right) - |E_k|\left(\sin(\Theta_i - \Theta_k)G_{ik} - \cos(\Theta_i - \Theta_k)B_{ik}\right) && \text{if } i = j \\
&= -|E_i|\left(\sin(\Theta_i - \Theta_k)G_{ik} - \cos(\Theta_i - \Theta_k)B_{ik}\right) && \text{if } k = j
\end{aligned}
$$

$$
\begin{aligned}
\frac{\partial q_{ik}}{\partial \Theta_j} &= -|E_i||E_k|\left(\cos(\Theta_i - \Theta_k)G_{ik} + \sin(\Theta_i - \Theta_k)B_{ik}\right) && \text{if } i = j \\
&= -|E_i||E_k|\left(-\cos(\Theta_i - \Theta_k)G_{ik} - \sin(\Theta_i - \Theta_k)B_{ik}\right) && \text{if } k = j
\end{aligned}
$$

$$
\frac{\partial p_{ik}}{\partial E_j} = \frac{\partial p_{ik}}{\partial \Theta_j} = \frac{\partial q_{ik}}{\partial E_j} = \frac{\partial q_{ik}}{\partial \Theta_j} = 0 \qquad \text{if } j \neq i \text{ or } k \tag{2.23}
$$

The following are used only in bus power calculations:

$$
\begin{aligned}
\frac{\partial P_{shunt_i}}{\partial E_j} &= -2|E_i|G_{shunt_i} && \text{if } i = j \qquad & \frac{\partial P_{shunt_i}}{\partial \Theta_i} &= 0 && \text{for all } j \\
&= 0 && \text{if } i \neq j \\
\frac{\partial Q_{shunt_i}}{\partial E_j} &= 2|E_i|B_{shunt_i} && \text{if } i = j \qquad & \frac{\partial Q_{shunt_i}}{\partial \Theta_i} &= 0 && \text{for all } j \\
&= 0 && \text{if } i \neq j
\end{aligned} \tag{2.24}
$$

Note that these are identical to the load flow equations. Again, they can be verified by hand calculations.

Once the Jacobian has been found, $\Delta\mathbf{X}$, can be computed from

$$
\Delta\mathbf{X} = \left[\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\right]^{-1}\mathbf{H}^T\mathbf{R}^{-1}\left[z_i - f_i(x)\right] \tag{2.25}
$$

where **H** and $[z_i - f_i(x)]$ are defined as before and R is the measurement error variance matrix,

$$\mathbf{R} = \begin{bmatrix} \sigma_1^2 & 0 & 0 & \cdots \\ 0 & \sigma_2^2 & 0 & \cdots \\ 0 & 0 & \sigma_3^2 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{2.26}$$

where σ_i^2 is the measurement variance of the i^{th} measurement. The derivation of (2.25) is extensive and can be found in [1]. It results from a Taylor expansion of the solution set [1].

From here, the new state is computed ($\mathbf{X} = \mathbf{X} + \Delta\mathbf{X}$). At this point, the maximum value of the $\Delta\mathbf{X}$ vector is compared to the error tolerance entered by the user. If it is higher, the program loops back to the calculation of the f_i's. If less, the routine goes to the error detection stage.

The purpose of error detection is to determine whether or not "bad" measurements are present. If they are not identified and removed, an erroneous estimate will be obtained. The estimate residual J(x) and normalized measurement residuals are used to help detect errors. J(x) is a residual of the whole estimate and is computed from

$$\mathrm{J}(\mathbf{x}) = \sum_{i=1}^{N_m} \frac{[\mathrm{z}_i - \mathrm{f}_i(\mathbf{x})]^2}{\sigma_i^2} \tag{2.27}$$

This should be recognized as a weighted least-squares residual. This residual is assumed to be a random number (since the measurements have random errors) and follows a chi-squared distribution. Along with this quantity, the normalized measurement residuals also assist in error detection. They are calculated as follows:

$$\mathrm{y}_i^{norm} = \frac{\mathrm{z}_i - \mathrm{f}_i^{est}}{\sigma_{y_i}} \tag{2.28}$$

where σ_{y_i} is the i^{th} diagonal element of

$$\mathbf{R}-\mathbf{H}\left[\mathbf{H}^{T}\mathbf{R}^{-1}\mathbf{H}\right]^{-1}\mathbf{H}^{T} \tag{2.29}$$

This is the estimate variance as opposed to the measurement variance. [4]

After these quantities are calculated, the error testing takes place. By setting a significance level α for the chi-squared distribution (this sets the probably of a "false alarm") of J(x), a threshold t_J can be found above which it can be assumed that there is a bad measurement present [1]. (The degrees of freedom, which is the number of measurements minus the number of buses, is also used to find t_J.) This is based on the fact that a bad measurement will cause J(x) to be high. The t_J threshold simply uses the chi-squared property of J(x) to determine when J(x) is too high. [1]

If there is a normalized measurement residual with a magnitude greater than 3, it is also assumed that a bad measurement is present. Three standard deviations is an accepted limit for values in a normal distributions. It is assumed that any measurements deviating further are probably bad [1]. If there are no bad measurements, then the routine is exited and the final results are computed. If there are, then the measurement with the highest normalized residual is identified and removed. The estimation is then re-computed with one less measurement. This is repeated until no more errors are detected.

CHAPTER 3

PROGRAM DESIGN CONSIDERATIONS

The object of this project was to add state estimation and load flow capabilities to the Bus program, as well as to correct errors in the short circuit calculations. Since the program already existed, the interface (window) and basic features did not have to be re-written, although many additions had to be made. Since the program interfaces with MATLAB to perform the calculations, two new MATLAB routines (state estimation and load flow) had to be written and set up to read and write the appropriate data.

In making additions to the program, many considerations had to be made, especially with data entry and interfacing with the MATLAB routines. The changes made to the program (including MATLAB routines) and the factors considered in making them are the subject of this chapter. It is strongly urged that the reader refer to the User's Manual (App. B) while reading this chapter.

3.1 Program Interface Changes

3.1.1 Window and Drawing Features

The program window and drawing features created with the original program were entirely sufficient. All of the elements needed for the analyses (buses, lines, transformers) could be drawn, so nothing had to be changed.

3.1.2 Menu

Although all of the menu items were needed for either basic operation or short circuit calculations, most of them are applicable to the new analyses. Thus, only two changes needed to be made to the program's menu. These were the **View Line Flow** and **View Bus Power** features added to display short circuit and load flow results. These features are discussed in Section 3.5.

3.2 Data Entry

One area of the program which did require quite a few additions was data entry. The previous version of the program provided for entry of line and transformer parameters and bus data (load and generator powers and shunt impedances). This was sufficient for short circuit analysis, and provided all the needed data for load flow studies, with the exception of the line charging capacitance. This is also needed for the state estimation studies, so the line impedance dialog was modified to include this quantity.

State estimation, however, required many more changes than the load flow calculation. Because most of the data (aside from the aforementioned parameters) is in the form of measurements, accommodations had to be made for entering them. New dialog boxes were created for this purpose.

Since data for both bus and line measurements needed to be entered, the main issue was how to incorporate all of this into dialog boxes. The boxes could not be confusing or cumbersome and the process needed to be simple.

Since the box for line parameters was still needed, it was decided that all the data would be entered by double-clicking on a bus. The bus data would be entered first, followed by data for each line in a separate dialog. This not only allows all data to be entered conveniently, but also helps the user keep the line flow direction straight.

For the bus measurements, one large dialog box is presented where the power and voltage measurements can be entered. (App. B, Fig. 4.5) This replaces the dialog box used in short circuit and load flow studies. The user is presented with groups of check boxes and text entries. These were placed in such a way as to not only be easy to look at but easy to navigate as well.

The two check boxes asking if a generator or load is present serve two purposes. They not only specify what type of bus is present, but whether all of the necessary measurements have been made (For power measurements, if a generator and load is present, a measurement must be entered for each.).

The other set of check boxes lets the user specify which measurements have been entered. This not only helps the program in compiling the data, but lets the user see which measurements have been made, so additions or deletions can be made.

Associated with each check box are two text boxes. In the first one, the measurement is entered, and in the second one, the measurement variance. If a measurement has been made, both boxes must be filled in. There are many check boxes and data boxes in the dialog, but the user will find the box very straightforward and easy to use.

The line measurement dialog is similar to that for buses in that a check box and two text boxes are presented for each measurement (P and Q). The flow direction on the line is listed in the dialog box. Only one P and/or Q measurement is made on the line in the given direction. One dialog is presented for each line connected to the selected bus. As mentioned previously, these are opened after the bus dialog.

With all data entry taken care of, the only other issue was entering the error tolerance and initial values for the load flow and short circuit routines. To make sure these parameters were entered, it was decided that a dialog asking for these quantities should appear as soon as the type of analysis was chosen. In this dialog, the user is given a text entry for the error tolerance and a set of buttons, from which the initial state may be set. The user is presented the option of a flat start, initial state entered from keyboard, or from an m-file. If the keyboard option is chosen, a series of dialogs will appear, one for each bus. The initial voltage and angle for each bus may be entered in these. The file option opens a box asking for the file name. The file must be in MATLAB format. (See Appendix B.)

3.3 MATLAB

Since the short circuit calculations were previously written, they will not be discussed here. The state estimation and load flow routines, which were written from the ground up, will be discussed. Various aspects of their structure will be discussed in this section. Since the state estimation routine was written first, it will be discussed first. For details on the algorithms and specific calculations involved, see Chapter 2.

The flowchart presented in Section 2.3 was the model for the MATLAB routine. The calculations were carried out in the same order. Since the equations to be used were already defined, the two main issues were passing the necessary data to MATLAB and making sure the correct quantities were calculated at the correct time in the routine.

The first data to be considered was the impedances. With the equations in their present form, only the line impedances were needed, along with any shunt admittances connected to the buses. Since the data was entered as resistance and reactance (R and X), R and X matrices were compiled and passed to MATLAB from the program. A MATLAB routine converts these to conductance and susceptance (G and B) matrices. The charging capacitances are compiled into another array and used "as is", as are the shunts. All of these arrays are called from the routine.

The measurements and variances were taken and assembled into separate arrays. To designate the type of measurement being calculated which is which and to make sure the correct quantities are computed, a couple of additional arrays are created. One contains a type number for each measurement (mtype) so the correct equations may be used, and the other contains the node(s) associated so the correct variables may be used in the calculation. If measurements are added or subtracted, these vectors change correspondingly. This ensures that the correct quantities are always calculated. This appeared to be the best way to provide information on the types and locations of measurements to MATLAB.

With the calculations and variables set, the routine was tested. Once it was working, modifications were made to speed up the algorithm. Some suggestions from [X] were followed. All variables were pre-defined. Instead of calling the previously mentioned arrays in each function, they are called once and passed as arguments to each function. And instead of calling these functions each time through a loop to calculate the f_i's and derivatives, the function is called once, and then the loop is instituted. With these changes made, computation time (for both routines) was decreased by 80-90%.

The load flow routine was adopted from the state estimation routine so it was structured similarly. Both routines use functions to calculate most of the functions and derivatives. There are, however, a couple of major differences. Instead of measurements, the bus powers entered by the user are used (in the same capacity). Similar type and mode arrays like before are also used with this. Another difference is with the reactive power (Q) limits on the generator. The limits are passed as a matrix so the calculated Q can be compared to the limits. The flowchart of Figure 4.4 was followed in this routine. No error rejection feature is needed in this routine.

This routine was also modified to speed it up, using the same techniques as before.

3.4 Display of Results

The short circuit analysis displayed all results on the system (currents and voltages). The load flow and state estimation studies display information differently. Since more information is involved (e.g., line flows each way), placing it on the diagram could be crowded and very difficult to read. Therefore, two dialog boxes were created to display results, one for bus generator or load powers and the other for line flows.

The net generation or load at a bus (real and reactive) is displayed in the bus dialog. The line flow dialog contains the real and reactive power flows in each direction on the line. This provides an easily accessible, simple way to view the results. Voltages can still be displayed directly on the diagram, however.

In addition to this, the results are displayed in MATLAB in a tabular format. This provides an easy-to-read and printable summary of the results. For load flow studies, more detailed information is included as well.

CHAPTER 4

TEST RESULTS

Several test cases were set up to test the operation of the calculation algorithms. These test cases and the results are presented in this chapter to show that the program operates correctly.

4.1 Short Circuit Analysis

The test case used to present the short circuit calculations is a 6 bus, 11 line system similar to Figure 4.8 of [1]. However, instead of that system's impedances, reactances of .1 pu and resistances of .01 pu were used. The transformers connected to each generator are delta-wye transformers with a grounding impedance on the wye side (system side). A one-line diagram is shown in Figure 4.1.

Both 3-phase and phase-ground analyses were performed on the system, with the fault bus arbitrarily placed at bus 6. The results of these analyses are presented in Tables 6.1 and 6.2.

Inspection of these results will show them to be correct. The currents and voltages calculated for both analyses are reasonable. Since the short circuit routine was written prior to this project (a few bugs were corrected), this system was used only to show the operation of this analysis (NOT for testing). Several other small systems were tested, although they are not presented here, and the results were found to be correct as well.

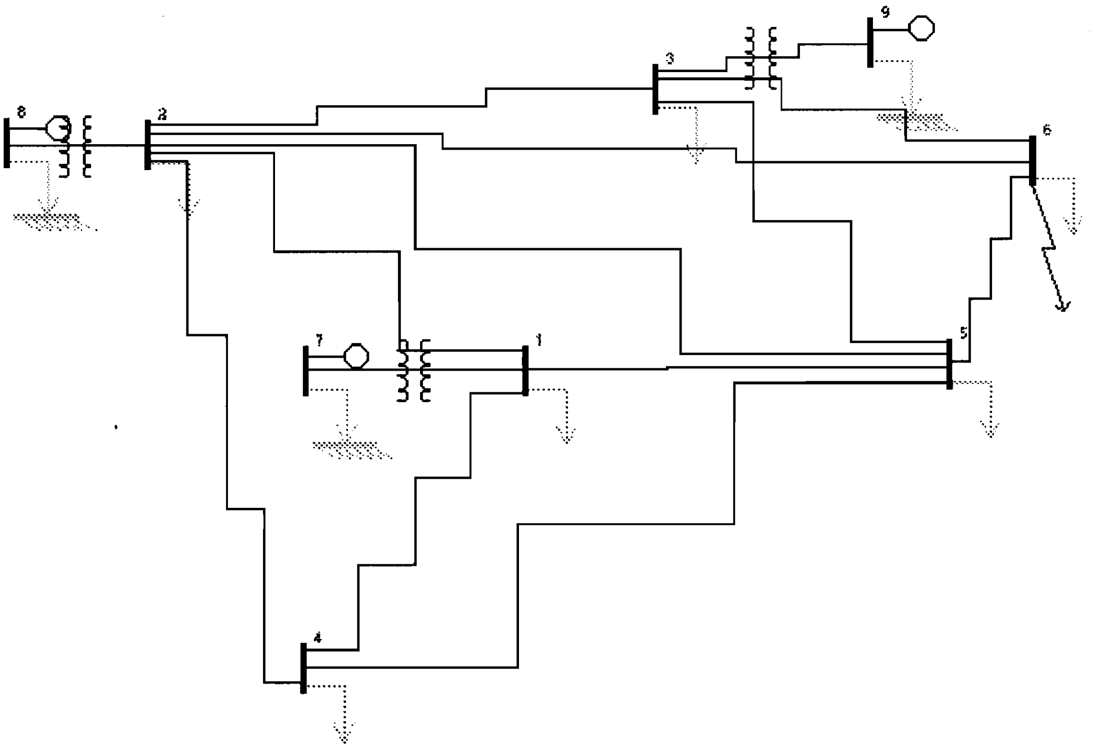


Figure 4.1 One-Line Diagram for Short Circuit Test Case

Table 4.1 Current Flows for Short Circuit Test Case

Line	3-Phase Fault	Phase-ground Fault	
		Pos. Seq.	Zero Seq.
1-2	.080 - j .799	.067 - j .674	.023 - j .235
1-4	.071 - j .713	.060 - j .597	.021 - j .205
1-5	.134 - j 1.341	.112 - j 1.119	.038 - j .382
1-7	-.285 + j 2.852	-.485 + j 4.856	0
2-3	.020 - j .196	.017 - j .170	.006 - j .062
2-4	-.009 + j .086	-.008 + j .076	-.003 + j .029
2-5	.054 - j .542	.045 - j .445	.015 - j .147
2-6	.340 - j 3.398	.281 - j 2.803	.094 - j .936
2-8	-.325 + j 3.252	-.531 + j 5.310	0
3-5	.035 - j .345	.027 - j .275	.008 - j .085
3-6	.320 - j 3.201	.264 - j 2.633	.087 - j .874
3-9	-.335 + 3.350	-.542 + j 5.426	0
4-5	.063 - j .628	.052 - j .521	.018 - j .176
5-6	.286 - j 2.856	.236 - j 2.359	.079 - j .789

Table 4.2 Bus Voltages and Angles for Short Circuit Test Case

Bus	3-Phase Fault		Phase-ground Fault			
			Positive Sequence		Zero Sequence	
	V	Ang.	V	Ang.	V	Ang.
1	0.42	0	0.35	0	0.33	3.14
2	0.34	0	0.28	0	0.36	3.14
3	0.32	0	0.27	0	0.36	3.14
4	0.35	0	0.29	0	0.35	3.14
5	0.29	0	0.24	0	0.37	3.14
6	0.00	0	0.00	0	0.45	3.14
7	0.71	0	0.84	0	0.00	0
8	0.67	0	0.82	0	0.00	0
9	0.66	0	0.81	0	0.00	0

4.2 Load Flow

Two test cases will be presented to demonstrate the load flow calculations. The first case is the 6 bus, 11 line system of Figure 4.8 in [1]. This system was used to test and debug the routine, using the given solution as a guideline. The test system has parameters identical to those specified in Chapter 4, Appendix A of [1]. These are shown in Table 4.3. A one-line diagram of the system (same topology as the short circuit case, minus the transformers) is shown in Figure 4.2, and the results are presented in Tables 4.4, 4.5, and 4.6. In these tables, the results from the bus program are compared to the results from [1]. Although the program gives per unit results, they are displayed in MW or MVAR here for easier reading.

It can be seen from the tabulated results that the Bus program results are correct. There are only a few small deviations from the given results.

Another test case was set up to double-check the routine on a different system. The 5 bus, 6 line system of Example 8.3 in [3] was used. A schematic of this system is shown in figure 4.3. Tables 4.7 through 4.9 present the results from the Bus program compared to the given solution in per unit. This time the results were basically identical. This showed that the load flow routine is working correctly.

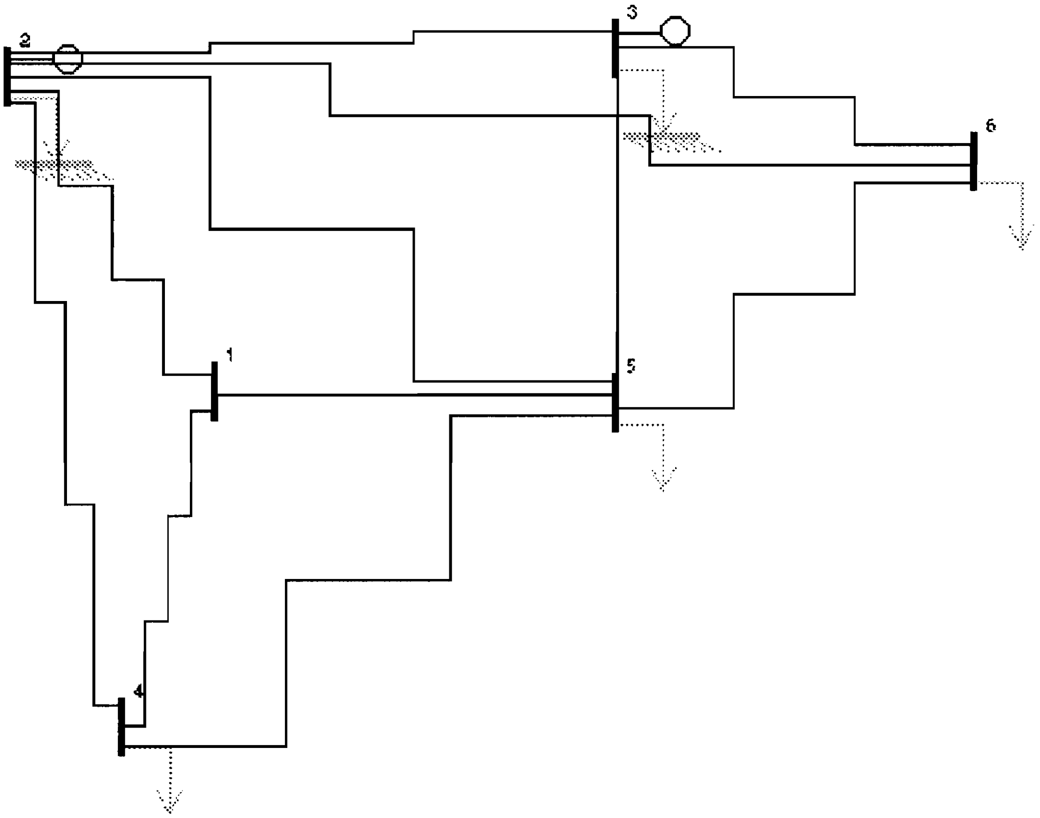


Figure 4.2 Load Flow Test Case from [1] (Test Case 1)

Table 4.3 System Parameters for Test Case 1

Line	R	X	Bcap
1-2	.1	.2	.02
1-4	.05	.2	.02
1-5	.08	.3	.03
2-3	.05	.25	.03
2-4	.05	.1	.01
2-5	.1	.3	.02
2-6	.07	.2	.025
3-5	.12	.26	.025
3-6	.02	.1	.01
4-5	.2	.4	.04
5-6	.1	.3	.03

Table 4.4 Line Power Flows for Test Case 1

Line	Reference 1 Solution		Bus program Solution	
	P	Q	P	Q
1-2	28.7	-15.4	**28.7**	**-15.4**
1-4	43.6	20.1	**43.6**	**20.1**
1-5	35.6	11.3	**35.6**	**11.3**
2-1	-27.8	12.8	**-27.8**	**12.8**
2-3	2.9	-12.3	**2.9**	**-12.3**
2-4	33.1	46.1	**33.1**	**46.1**
2-5	15.5	15.4	**15.5**	**15.4**
2-6	26.2	12.4	**26.2**	**12.4**
3-2	-2.9	5.7	**-2.9**	**5.7**
3-5	19.1	23.2	**19.1**	**23.2**
3-6	43.8	60.7	**43.8**	**60.7**
4-1	-42.5	-19.9	**-42.5**	**-19.9**
4-2	-31.6	-45.1	**-31.6**	**-45.1**
4-5	4.1	-4.9	**4.1**	**-4.9**
5-1	-34.5	-13.5	**-34.5**	**-13.4**
5-2	-15.0	-18.0	**-15.0**	**-18.0**
5-3	-18.0	-26.1	**-18.0**	**-26.1**
5-4	-4.0	-2.8	**-4.0**	**-2.8**
5-6	1.6	-9.7	**1.6**	**-9.7**
6-2	-25.7	-16.0	**-25.7**	**-16.0**
6-3	-42.8	-57.9	**-42.8**	**-57.9**
6-5	-1.6	3.9	**-1.6**	**3.9**

Table 4.5 Bus Powers for Test Case 1

Bus	Reference 1 Solution		Bus program Solution	
	P	Q	P	Q
1	107.9	16.0	**107.9**	**16.0**
2	50.0	74.4	**50.0**	**74.4**
3	60.0	89.6	**60.0**	**89.6**
4	-70.0	-70.0	**-70.0**	**-70.0**
5	-70.0	-70.0	**-70.0**	**-70.0**
6	-70.0	-70.0	**-70.0**	**-70.0**

Table 4.6 Bus Voltages and Angles for Test Case 1

Bus	Reference 1 Solution		Bus program Solution	
	V	**Ang.**	**V**	**Ang.**
1	241.5	0	241.5	0
2	241.5	-3.7	241.5	-3.7
3	246.1	-4.3	246.1	-4.3
4	227.6	-4.2	227.5	-4.2
5	226.7	-5.3	226.6	-5.3
6	231.0	-5.9	230.9	-6.0

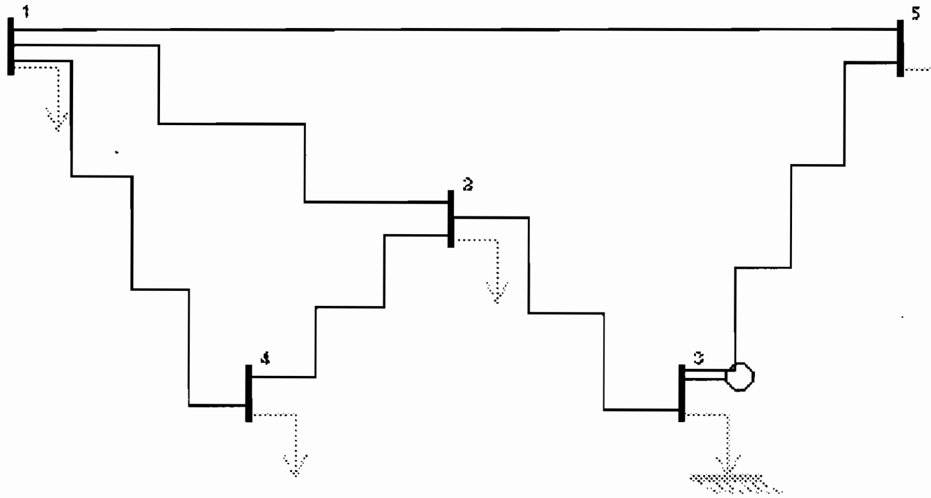


Figure 4.3 Load Flow Test Case from [3] (Test Case 2)

Table 4.7 Line Power Flows for Test Case 2

Line	**Reference X Solution**		**Bus program Solution**	
	P	**Q**	**P**	**Q**
1-2	.198	.123	**.198**	**.123**
1-4	.248	.117	**.248**	**.117**
1-5	.205	.089	**.205**	**.089**
2-1	-.193	-.102	**-.193**	**-.102**
2-3	-.573	-.237	**-.573**	**-.237**
2-4	.166	.039	**.166**	**.039**
3-2	.594	.321	**.594**	**.321**
3-5	.406	.155	**.406**	**.155**
4-1	-.237	-.074	**-.237**	**-.074**
4-2	-.163	**-.206	**-.163**	**-.026**
5-1	-.203	-.079	**-.203**	**-.079**
5-3	-.397	-.121	**-.397**	**-.121**

** - Typographical Error

Table 4.8 Bus Powers for Test Case 2

Bus	Reference 1 Solution		Bus program Solution	
	P	Q	P	Q
1	.652	.329	**.651**	**.329**
2	-.600	-.300	**-.600**	**-.300**
3	1.00	.477	**1.000**	**.477**
4	-.4	-.1	**-.4**	**-.1**
5	-.6	-.2	**-.6**	**-.200**

Table 4.9 Bus Voltages and Angles for Test Case 2

Bus	Reference 1 Solution		Bus program Solution	
	V	**Ang.**	**V**	**Ang.**
1	1.02	0	1.02	0
2	.955	-.069	.955	-.069
3	1.04	.035	1.04	.035
4	.923	-.140	.923	-.140
5	.993	-.036	.993	-.036

4.3 State Estimation

Three different test cases were run to test the state estimation routine, all on the same 6 bus system used for the load flow study. The system parameters were the same, with the measurements given in Chapter 12 of [1] as the only added data.

The three different cases run on this system were: one with all measurements present, one with two of these measurements made "bad" by making them the negative of the actual values, and one with measurements from only two buses present. The first and last cases tested the estimator and the second tested the error rejection feature. Tables 4.10 through 4.18 compare the results from the Bus program with the base case (load flow) values, the measurements, and the given estimate. These tables display results in MW, MVAR, or Volts instead of per unit for easier reading. Review of these tables will show that the routine correctly computed the estimate in all three cases. There are some deviations between the given and program estimates, with the Bus program estimate usually more accurate. It should be mentioned that for the case with the line 1-2 measurements multiplied by -1, the estimate from [1] is before the error rejection. So for this case, the program's results is compared to the other estimate only to show the effects of error rejection.

Thus, it was shown that the program could come up with acceptable estimate given many or a few measurements and could also identify and reject "bad" measurements. This fulfills the expectations for the routine.

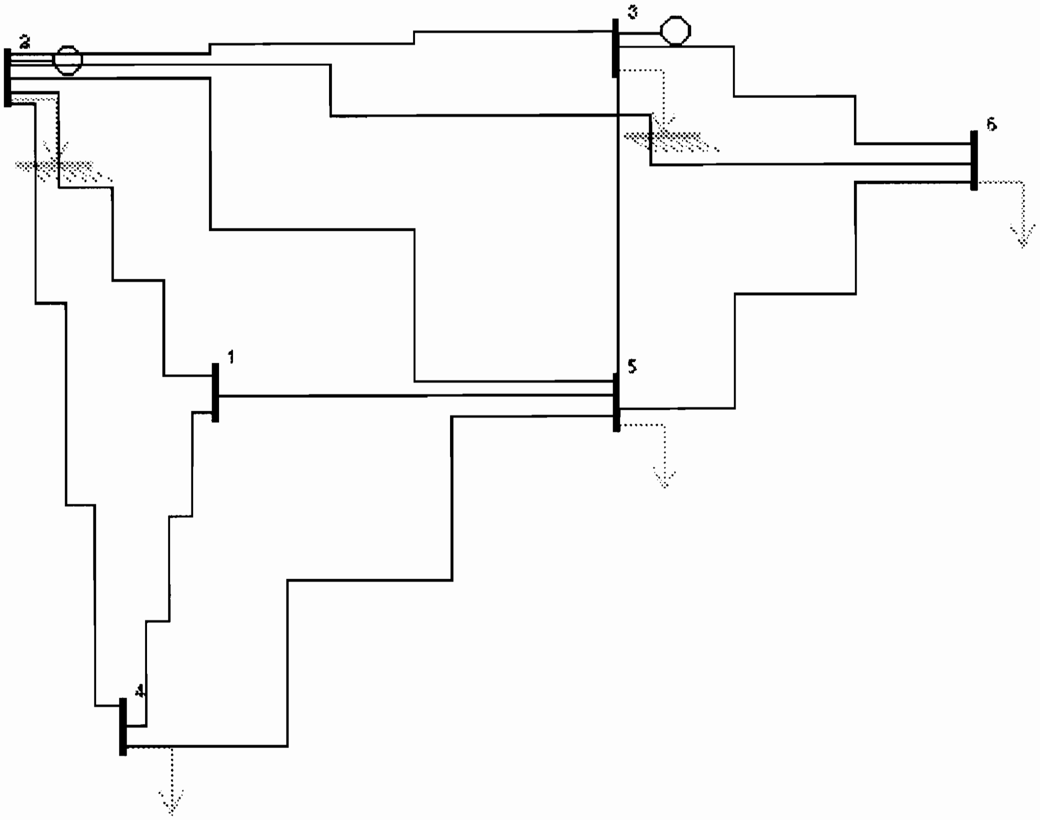


Figure 4.4 One-Line Diagram for State Estimation Test Cases

Table 4.10 Line Power Flows for State Estimation w/ All Measurements

Line	Base-case value		Measured value		Reference 1 estimate		Bus program estimate	
	P	Q	P	Q	P	Q	P	Q
1-2	28.7	-15.4	31.5	-13.2	30.4	-14.4	**30.4**	**-14.3**
1-4	43.6	20.1	38.9	21.2	44.8	21.2	**44.8**	**21.2**
1-5	35.6	11.3	35.7	9.4	36.8	11.8	**36.8**	**11.8**
2-1	-27.8	12.8	-34.9	9.7	-29.4	11.9	**-29.4**	**11.9**
2-3	2.9	-12.3	8.6	-11.9	3.0	-12.6	**3.0**	**-12.6**
2-4	33.1	46.1	32.8	38.3	32.4	45.3	**32.3**	**45.2**
2-5	15.5	15.4	17.4	22.0	15.6	14.8	**15.6**	**14.8**
2-6	26.2	12.4	22.3	15.0	25.9	10.8	**25.9**	**10.9**
3-2	-2.9	5.7	-2.1	10.2	-3.0	6.2	**-3.0**	**6.2**
3-5	19.1	23.2	17.7	23.9	19.2	22.9	**19.2**	**23.0**
3-6	43.8	60.7	43.3	58.3	43.3	58.3	**43.3**	**58.5**
4-1	-42.5	-19.9	-40.1	-14.3	-43.6	-20.7	**-43.6**	**-20.7**
4-2	-31.6	-45.1	-29.8	-44.3	-30.9	-44.4	**-30.9**	**-44.3**
4-5	4.1	-4.9	0.7	-17.4	4.3	-5.1	**4.3**	**-5.1**
5-1	-34.5	-13.5	-36.6	-17.5	-35.6	-13.6	**-35.6**	**-13.7**
5-2	-15.0	-18.0	-11.7	-22.2	-15.1	-17.4	**-15.1**	**-17.4**
5-3	-18.0	-26.1	-25.1	-29.9	-18.1	-25.8	**-18.1**	**-25.8**
5-4	-4.0	-2.8	-2.1	-1.5	-4.2	-2.5	**-4.2**	**-2.6**
5-6	1.6	-9.7	-2.1	-0.8	1.3	-10.1	**1.3**	**-10.1**
6-2	-25.7	-16.0	-19.6	-22.3	-25.4	-14.5	**-25.4**	**-14.5**
6-3	-42.8	-57.9	-46.8	-51.1	-42.3	-55.7	**-42.3**	**-55.8**
6-5	-1.6	3.9	1.0	2.9	-1.2	4.4	**-1.2**	**4.4**

Table 4.11 Bus Powers for State Estimation w/ All Measurements

Bus	Base-case value		Measured value		Reference 1 estimate		Bus program estimate	
	P	Q	P	Q	P	Q	P	Q
1	107.9	16.0	113.1	20.2	111.9	18.7	**111.9**	**18.8**
2	50.0	74.4	48.4	71.9	47.5	70.3	**47.6**	**70.2**
3	60.0	89.6	55.1	90.6	59.5	87.4	**59.5**	**87.6**
4	-70.0	-70.0	-71.8	-71.9	70.2	-70.2	**-70.2**	**-70.0**
5	-70.0	-70.0	-72.0	-67.7	71.8	-69.4	**-71.8**	**-69.5**
6	-70.0	-70.0	-72.3	-60.9	68.9	-65.8	**-69.0**	**-66.0**

Table 4.12 Bus Voltages and Angles for State Estimation w/ All Measurements

Bus	Base-case value	Measured value	Reference 1 estimate	Bus prog. estimate
	V	V	V	V
1	241.5	238.4	240.6	240.6
2	241.5	237.8	239.9	239.7
3	246.1	250.7	244.7	244.5
4	227.6	225.7	226.1	226.1
5	226.7	225.2	225.3	225.2
6	231.0	228.9	230.1	230.0

Table 4.13 Line Power Flows w/ Line 1-2 Measurements "Bad"

Line	Base-case value		Measured value		Bus program estimate	
	P	Q	P	Q	P	Q
1-2	28.7	-15.4	-31.5	13.2	**30.3**	**-14.4**
1-4	43.6	20.1	38.9	21.2	**44.7**	**21.1**
1-5	35.6	11.3	35.7	9.4	**36.7**	**11.8**
2-1	-27.8	12.8	-34.9	9.7	**-29.3**	**12.0**
2-3	2.9	-12.3	8.6	-11.9	**3.0**	**-12.6**
2-4	33.1	46.1	32.8	38.3	**32.4**	**45.2**
2-5	15.5	15.4	17.4	22.0	**15.6**	**14.8**
2-6	26.2	12.4	22.3	15.0	**25.9**	**10.9**
3-2	-2.9	5.7	-2.1	10.2	**-3.0**	**6.2**
3-5	19.1	23.2	17.7	23.9	**19.2**	**23.0**
3-6	43.8	60.7	43.3	58.3	**43.3**	**58.5**
4-1	-42.5	-19.9	-40.1	-14.3	**-43.6**	**-20.6**
4-2	-31.6	-45.1	-29.8	-44.3	**-30.9**	**-44.3**
4-5	4.1	-4.9	0.7	-17.4	**4.3**	**-5.1**
5-1	-34.5	-13.5	-36.6	-17.5	**-35.6**	**-13.6**
5-2	-15.0	-18.0	-11.7	-22.2	**-15.1**	**-17.5**
5-3	-18.0	-26.1	-25.1	-29.9	**-18.1**	**-25.8**
5-4	-4.0	-2.8	-2.1	-1.5	**-4.2**	**-2.6**
5-6	1.6	-9.7	-2.1	-0.8	**1.3**	**-10.1**
6-2	-25.7	-16.0	-19.6	-22.3	**-25.4**	**-14.5**
6-3	-42.8	-57.9	-46.8	-51.1	**-42.3**	**-55.8**
6-5	-1.6	3.9	1.0	2.9	**-1.2**	**4.4**

Table 4.14 Bus Powers w/ Line 1-2 Measurements "Bad"

Bus	Base-case value		Measured value		Bus program estimate	
	P	Q	P	Q	P	Q
1	107.9	16.0	113.1	20.2	**111.7**	**18.5**
2	50.0	74.4	48.4	71.9	**47.7**	**70.3**
3	60.0	89.6	55.1	90.6	**59.5**	**87.6**
4	-70.0	-70.0	-71.8	-71.9	**-70.2**	**-70.0**
5	-70.0	-70.0	-72.0	-67.7	**-71.8**	**-69.5**
6	-70.0	-70.0	-72.3	-60.9	**-68.9**	**-65.9**

Table 4.15 Bus Voltages and Angles w/ Line 1-2 Measurements "Bad"

Bus	Base-case value	Measured value	Bus prog. estimate
	V	V	V
1	241.5	238.4	240.6
2	241.5	237.8	239.7
3	246.1	250.7	244.7
4	227.6	225.7	226.1
5	226.7	225.2	225.2
6	231.0	228.9	230.0

Table 4.16 Line Power Flows w/ Bus 1&2 Measurements Only

Line	Base-case value		Measured value		Reference 1 estimate		Bus program estimate	
	P	Q	P	Q	P	Q	**P**	**Q**
1-2	28.7	-15.4	31.5	-13.2	30.6	-13.4	**30.6**	**-13.5**
1-4	43.6	20.1	38.9	21.2	44.7	19.4	**44.7**	**19.3**
1-5	35.6	11.3	35.7	9.4	37.1	14.6	**37.1**	**14.6**
2-1	-27.8	12.8	-34.9	9.7	-29.6	11.1	**-29.6**	**11.1**
2-3	2.9	-12.3	8.6	-11.9	8.8	-11.7	**8.8**	**-11.7**
2-4	33.1	46.1	32.8	38.3	30.5	40.2	**30.5**	**40.2**
2-5	15.5	15.4	17.4	22.0	16.1	16.8	**16.1**	**16.8**
2-6	26.2	12.4	22.3	15.0	22.4	15.2	**22.5**	**15.2**
3-2	-2.9	5.7			-8.7	5.5	**-8.7**	**5.5**
3-5	19.1	23.2			15.1	25.3	**15.1**	**25.3**
3-6	43.8	60.7			20.9	64.0	**21.0**	**63.9**
4-1	-42.5	-19.9			-43.6	-18.9	**-43.5**	**-18.8**
4-2	-31.6	-45.1			-29.3	-39.7	**-29.3**	**-39.7**
4-5	4.1	-4.9			5.3	-2.6	**5.3**	**-2.6**
5-1	-34.5	-13.5			-35.9	-15.9	**-35.8**	**-15.9**
5-2	-15.0	-18.0			-15.5	-19.0	**-15.5**	**-19.0**
5-3	-18.0	-26.1			-14.0	-28.0	**-28.0**	**-28.0**
5-4	-4.0	-2.8			-5.2	-4.8	**-5.2**	**-4.8**
5-6	1.6	-9.7			-1.4	-9.0	**-1.4**	**-9.0**
6-2	-25.7	-16.0			-21.9	-18.8	**-21.9**	**-18.8**
6-3	-42.8	-57.9			-20.0	-61.8	**-20.2**	**-61.7**
6-5	-1.6	3.9			1.4	3.4	**1.4**	**3.5**

Table 4.17 Bus Powers w/ Bus 1&2 Measurements Only

Bus	**Base-case value**		**Measured value**		**Reference 1 estimate**		**Bus program estimate**	
	P	**Q**	**P**	**Q**	**P**	**Q**	**P**	**Q**
1	107.9	16.0	113.1	20.2	112.4	20.5	**112.4**	**20.5**
2	50.0	74.4	48.4	71.9	48.2	71.7	**48.2**	**71.7**
3	60.0	89.6	55.1	90.6	27.2	94.9	**27.4**	**94.7**
4	-70.0	-70.0	-71.8	-71.9	-67.6	-61.2	**-67.6**	**-61.2**
5	-70.0	-70.0	-72.0	-67.7	-71.9	-76.7	**-71.9**	**-76.7**
6	-70.0	-70.0	-72.3	-60.9	-40.5	-77.2	**-40.7**	**-77.1**

Table 4.18 Bus Voltages and Angles w/ Bus 1&2 Measurements Only

Bus	Base-case value	Measured value	Reference 1 estimate	Bus prog. estimate
	V	V	V	V
1	241.5	238.4	238.8	238.7
2	241.5	237.8	237.6	237.6
3	246.1	250.7	241.4	241.3
4	227.6	225.7	225.0	224.9
5	226.7	225.2	221.4	221.3
6	231.0	228.9	226.2	226.1

CHAPTER 5
CONCLUSIONS

This project involved the addition of load flow and state estimation to the Bus program, as well as any correction needed on the original program. This involved making changes to the interface, as needed, and writing the two new calculation routines for MATLAB.

Most of the modifications to the program involved adding or modifying dialogs for users to enter the necessary data. This was most pertinent to the state estimation, where many measurements must be entered. All changes and additions were made in such a way as to be as user-friendly as possible.

The MATLAB calculation routines were written using flowcharts and calculations taken from the references. Both routines use the Newton-Raphson method to iteratively find a solution. They attempt to minimize the difference between the calculated and given quantities (powers or measurements). In addition to finding an estimate, the state estimation routine contains a procedure to identify and remove "bad" measurements from the estimate.

Several test cases were set up and solved to demonstrate the program's operation. It was found that the program performed all of the analyses correctly. In addition, the state estimation error rejection feature worked correctly. Thus, the program operates correctly and the project was successful.

CHAPTER 6

REFERENCES

1) A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*, New York: John Wiley & Sons, 1984.

2) W. D. Stevenson, Jr., *Elements of Power Systems Analysis, Fourth Ed.*, New York: McGraw-Hill, Inc., 1982.

3) T. Gönen, *Modern Power Systems Analysis*, New York: John Wiley & Sons, 1988.

4) J. F. Dopazo, O. A. Klitin, and A. M. Sasson, "State Estimation for Power Systems: Detection and Identification of Gross Measurement Errors," Proceedings of the Eighth PICA Conference, Minneapolis, June 1973, pp. 313-318.

5) A. H. Seidman, H. Mahrous, and T. G. Hicks, Eds., *Handbook of Electric Power Calculations*, New York: McGraw-Hill, 1984.

6) The MathWorks, Inc., *The Student Edition of MATLAB* (Manual), Englewood Cliffs, NJ: Prentice-Hall, Inc., 1992.

7) C. H. Pappas, W. H. Murray III, *Borland C++ Handbook, Fourth Ed.*, Berkeley, CA: McGraw-Hill, 1994.

8) Programmer's Guide, Turbo C++ 3.0 for Windows, Scotts Valley, CA: Borland International, 1991.

APPENDIX A
IMPEDANCE MATRICES

This appendix briefly describes the admittance and impedance matrices used in the calculations. This should help the reader better understand the calculations.

A.1 Short Circuit

The short circuit calculations create and utilize positive and negative-sequence admittance matrices. These matrices are constructed in the standard way (outlined in many books). For the positive sequence admittance matrix ($\mathbf{Y}_{\mathbf{busp}}$), the off-diagonal entries contain the line or transformer impedances, while the diagonal entries contain the sum of all impedances (lines and synchronous impedances) connected to the respective bus.

The zero-sequence matrix is similar except that the zero-sequence line and synchronous impedances replace those of the positive sequence. Only those lines present in the zero-sequence network are included. In addition, the diagonal terms include any transformer or generator grounding impedances connected to the bus multiplied by three.

Because the process involved in creating the matrices of the mutual inductances between lines is extensive and was written before this project, it will not be discussed.

A.2 Load Flow and Short Circuit

These two routines use the same impedance matrices and arrays, so they are discussed together. Several more arrays are involved here than in the short circuit calculations, although construction of them is simpler.

Because of the way the equations are constructed, the node impedances (diagonal entries) are not needed, so the resistance and reactance matrices contain only line and transformer impedances (off-diag. entries). The resistance and reactance of each line is placed in separate matrices as entered. A MATLAB routine later converts these elements to admittances. The only calculation needed in compiling these matrices is if a phase-shifting transformer is included. If so, the impedances for these elements are computed as follows:

$$y_{12_{real}} = y_{21_{real}} = \frac{y_{real} * mag * \cos(\Theta) + y_{imag} * mag * \sin}{mag^2}$$

$$R = y_{12_{real}} * \left(R^2 + X^2\right)$$

$$y_{12_{imag}} = y_{21_{imag}} = \frac{y_{imag} * mag * \cos(\Theta) - y_{real} * mag * \sin(\Theta)}{mag^2}$$

$$X = y_{12_{imag}} * \left(R^2 + X^2\right)$$

$$B_{cap_{12}} = y_{imag} - y_{12_{imag}}$$

$$B_{cap_{21}} = \frac{y_{imag}}{mag^2} - y_{21_{imag}} \qquad (A.1)$$

where mag is the phase-shifting transformer magnitude, Θ is the transformer angle, and y_{real} and y_{imag} are the resistance and reactance converted to admittances. These impedances are derived using the model and equations in Chapter 8 of [2]. If a magnitude-shifting transformer with a ratio other than unity is used, then more calculations are needed:

$$B_{cap_{12}} = y_{imag} * \left(\frac{a-1}{a} \right)$$
$$B_{cap_{21}} = y_{imag} * \left(\frac{1-a}{a^2} \right) \tag{A.2}$$

where a is the ratio of the transformer. The resistances and reactances of the transformer are simply divided by this ratio to get the new values.

The Bcap matrix contains the charging capacitances as entered by the user, except for the transformer cases outlined above. The values to be used in those cases are shown with the other calculations above. Since the charging capacitance applies only to lines, this matrix contains only off-diagonal entries as well.

The other arrays used are the shunt arrays. These contain the sums of any shunt impedances connected to the buses. Arrays are made for both the real and imaginary parts of the shunts.

Calculation and compilation of these matrices is fairly simple, as has been described.

Appendix B

Bus Program
User's Manual

Steven D. Estes

CENTER FOR POWER ENGINEERING
BRADLEY DEPT. OF ELECTRICAL ENGINEERING
VIRGINIA TECH
BLACKSBURG, VA 24061-0111
(703) 231-7029 Fax (703) 231-4303

Table of Contents

List of Figures

Disclaimer

Permission to copy and distribute this software is granted, provided that this notice is included. Although every effort has been made to prevent problems, the author makes no warranty of any kind, expressed or implied, with regard to the suitability or performance of this software.

Registered Trademarks

Borland C++ for Windows is a registered trademark of Borland International, Inc.

Microsoft Windows is a registered trademark or The Microsoft Corporation.

MATLAB is a registered trademark of The Math Works, Inc.

Introduction

A variety of programs have been written to perform calculations and analyses on power systems. However, since most of these programs are MS-DOS based or written on another platform, data entry (and resulting information) can be confusing or tiresome. The Bus program was written to not only make this process less confusing, but combine the ability to do more than one type of analysis from the same program. This program may be especially useful to students wishing to learn more about power systems.

The program provides users with a graphical interface onto which a one-line diagram of the system can be drawn. This allows users to actually "look" at the topology of their system while performing calculations on it. Data can be entered by simply clicking on the desired component. The results from the calculations are also easier to interpret since the calculated currents and voltages are displayed on the system (for short circuit calculations; state estimation and load flow results can be viewed in MATLAB or using boxes available from the menu).

The Bus program can handle up to a 75 bus, 75 line system in a wide variety of configurations. This gives users a flexible and valuable tool for analyzing power systems.

This manual acquaints users with the program and informs them about its operation. Chapter 2 gives brief descriptions of the menu items and related dialogs. Chapter 3 shows how to draw a sample system, while Chapter 4 covers the process involved in entering the system parameters.

Section 1: Installation and Setup

System Requirements

This program was written for a computer running Microsoft Windows 3.1. In order to perform the calculations, the system must also have MATLAB for Windows installed. The hardware requirements are as follows:

- 80386 Processor (80486 recommended)
- 4 MB minimum (8MB recommended)
- At least 1 MB Hard Disk space

Installing Bus on a Hard Drive

The distribution disk should contain the file BUS.EXE (the program) and other necessary files (in the directory \BUS). The program file may be placed in any directory on any hard drive. No decompression is necessary; just copy the file from the floppy disk to a hard drive. The files in the \BUS directory **MUST** go in the C:\BUS directory. If this instruction is not followed, the calculations cannot be carried out.

Running Bus.exe

There are two ways to run the Bus program: from the file manager or from the Program Manager. To run the program from the file manager, go to the directory the program is in and double-click the file name. The other way is to set up an icon in a program group in Program Manager. To do this, select the group you wish to add the program to. Then select **FILE | NEW** from the menu. Choose to create a new Program Item. Then enter the following information:

Description:	type "BUS" or the name you want the program to be called		
Command Line:	type "BUS"		
Working Directory:	type the directory name that BUS.EXE is in		
Short-cut Key:	N/A	Run Minimized:	N/A

Once the icon has been created, double-click it to run the Bus program.

Section 2: Program Interface

File Menu

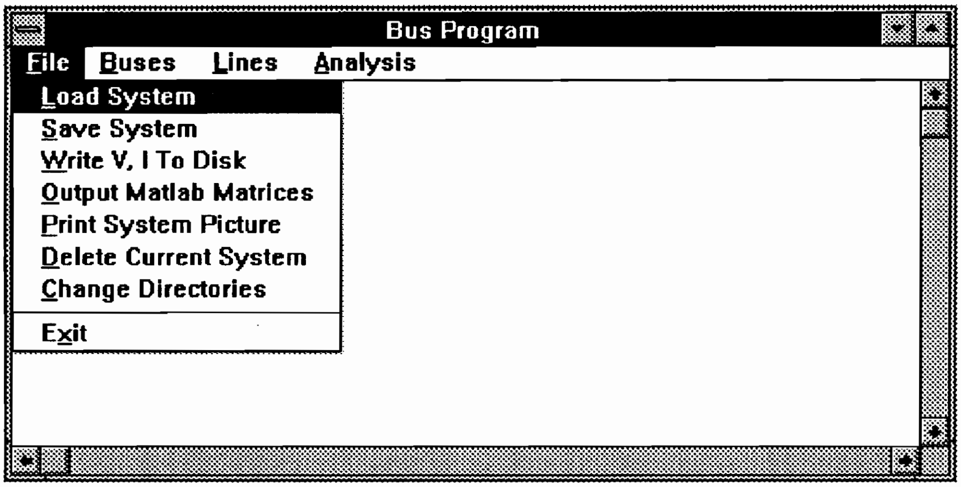
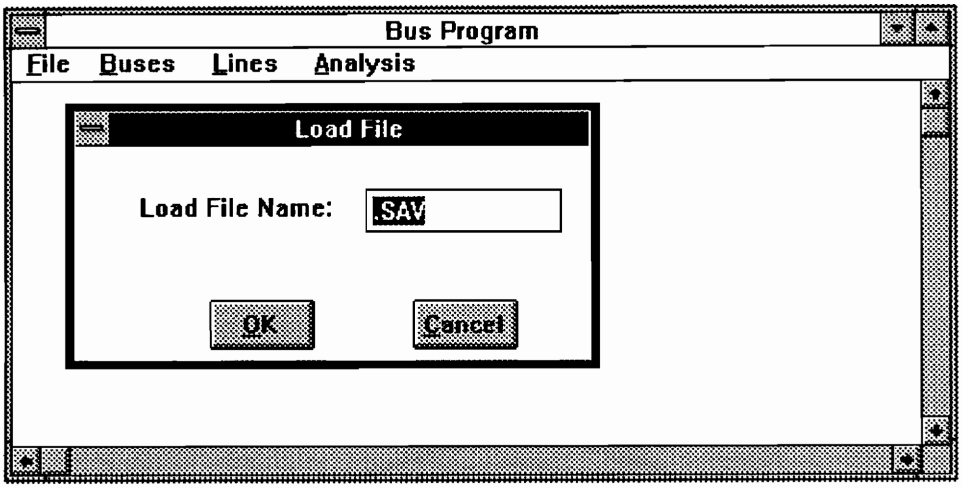


Figure 2.1 File Menu

Load System

Any system that was previously entered in the Bus program and saved can be retrieved by choosing this command. Enter the desired file name in the resulting dialog box (Figure 2.2). The file must be in the C:\BUS directory.



Load File Name: .SAV

Figure 2.2 Load File Dialog

Save System

Choosing this option will ask for a file name to save the current system in. The physical topology and all data from the system will be saved. The filename will be entered in a box like the one above. The file is saved in C:\BUS.

Write V, I to disk

Saves system and writes voltages and currents to C:\BUS. The voltage and current data files have the same names as the system except with the extensions ".cur" and ".vol".

Output MATLAB Matrices

Creates and writes the matrices necessary for MATLAB computations to the C:\BUS directory. The files written depend on the analysis selected. Refer to the listing in the back of the manual for file names.

Print System Picture

This option allows the user to print an image of the current system. If the system is large, all of it may not appear on the page. (NOTE: There may be a problem printing in Windows 95. This is because of an incompatibility with the print manager.)

Delete Current System

This deletes everything on the screen. A new system can then be entered.

Change Directories

If MATLAB is not installed in the C:\MATLAB directory, this option allows the user to specify which directory MATLAB is in. The box in which to enter the new directory looks like the file dialog already shown.

Exit

Exits the program.

Buses Menu

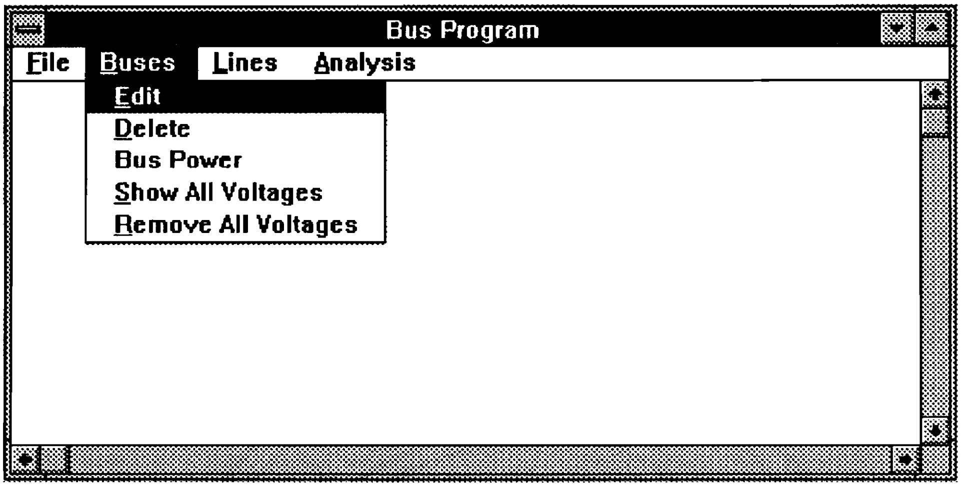


Figure 2.3 Buses Menu

Edit

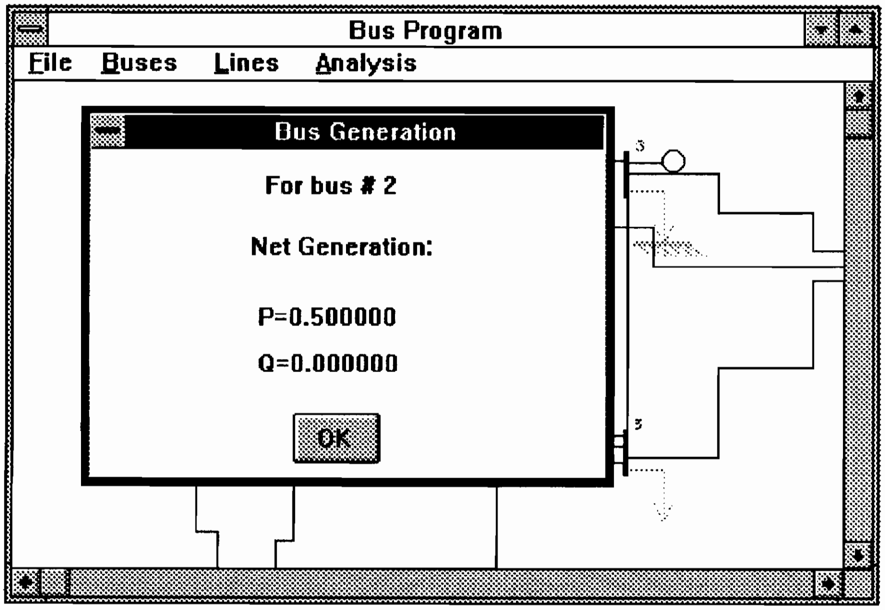
This allows you to enter or modify the highlighted bus data. The same thing can be accomplished by double-clicking the bus itself. See Section 4, Entering Data, for more details.

Delete

Deletes the highlighted bus from the current system. There must be no lines attached to the bus.

Bus Power

After load flow and state estimation calculations, this displays the net generation at the highlighted bus (Figure 2.4).



For bus # 2

Net Generation:

P=0.500000

Q=0.000000

Figure 2.4 Bus Generation Dialog

Show All Voltages

This displays the voltages and angles on all buses.

Remove All Voltages

Removes the previous option.

Lines Menu

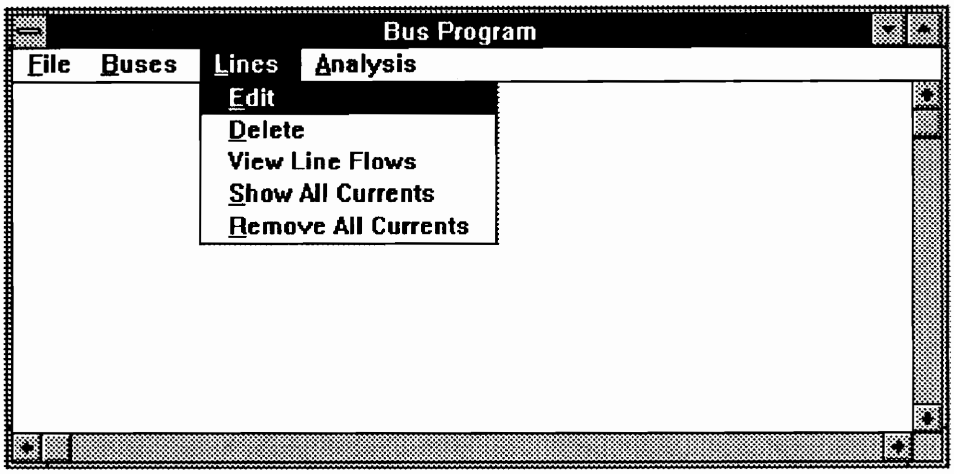


Figure 2.5 Lines Menu

Edit

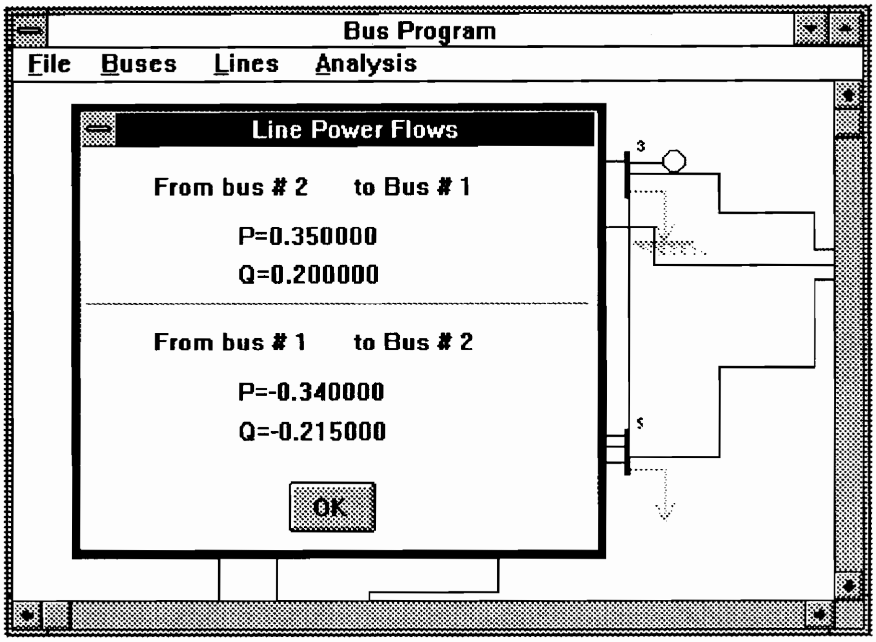
This allows you to enter or modify the highlighted line data. The same thing can be accomplished by double-clicking the line itself. See Section 4, Entering Data, for more details.

Delete

Deletes the highlighted line from the current system.

View Line Flows

After load flow and state estimation calculations, this displays the power flows in both directions on the highlighted line (Figure 2.6).



From bus # 2 to Bus # 1

P=0.350000

Q=0.200000

From bus # 1 to Bus # 2

P=-0.340000

Q=-0.215000

Figure 2.6 Line Flow Dialog

Show All Currents

This displays the currents on all lines. Note that this option is not available on the state estimation and load flow options.

Remove All Currents

Removes the previous option.

Analysis Menu

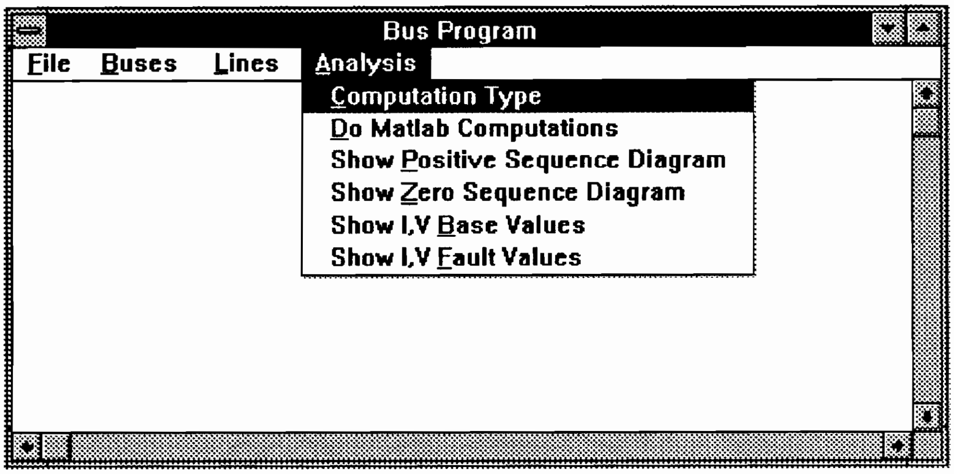


Figure 2.7 Analysis Menu

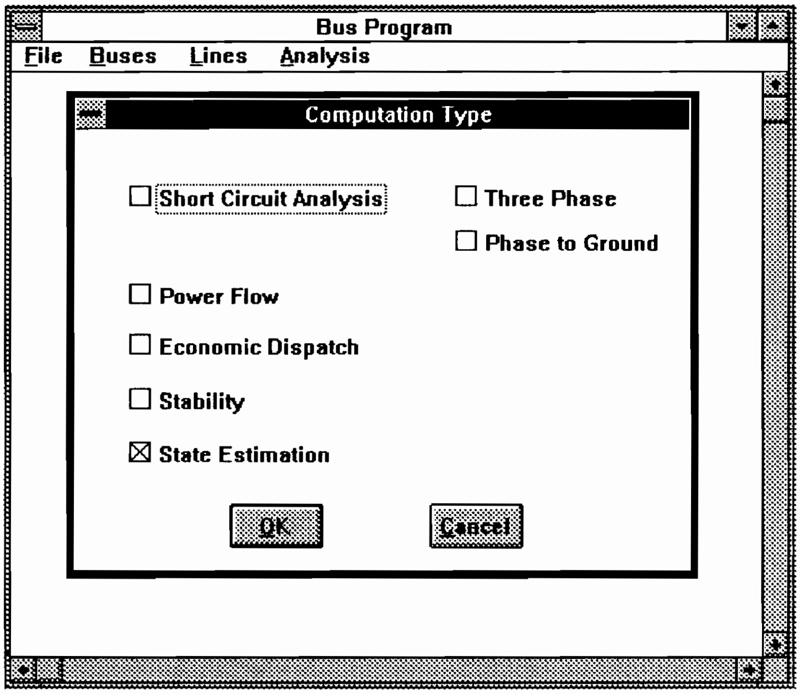
Computation Type

This allows the user to select which computation should be performed. The dialog in Figure 2.8 will appear for this purpose. The three types of computations now performed are short circuit, load flow, and state estimation. If load flow or state estimation is selected, another dialog box opens asking for the error tolerance and the initial state (Figure 2.9).

The initial state to be used in computations may be entered three ways. The first is to specify a flat start. This means that each voltage will be one, and each angle will be zero. The second is to enter the initial state from a keyboard. This will bring up a dialog for each bus asking for the initial voltage and angle. The third way is to specify a file which contains the initial state in MATLAB form. The file should be in the following format:

$$\begin{array}{ll} x=[& E_1 \\ & \Theta_1 \\ & | \\ & | \\ & E_n \\ & \Theta_n]; \end{array}$$

where E_i are the bus voltages and Θ_i are the bus voltage angles (in radians). For state estimation, the voltage angle at the swing bus should be omitted. A state vector saved in MATLAB (see MATLAB help for info. on saving a variable) is acceptable, but the `x=` and brackets will need to be added.



☐ Short Circuit Analysis

☐ Three Phase

☐ Phase to Ground

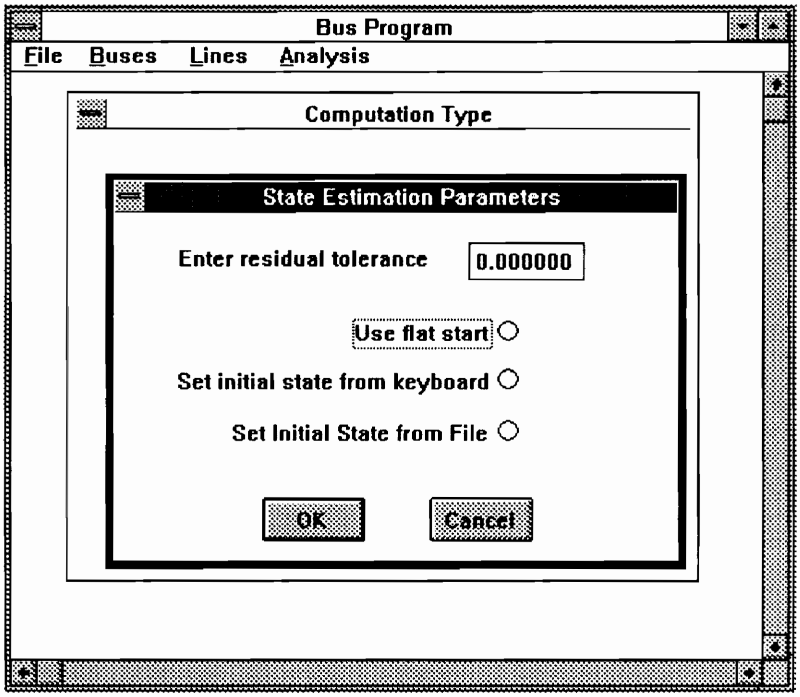
☐ Power Flow

☐ Economic Dispatch

☐ Stability

☒ State Estimation

Figure 2.8 Computation Type Dialog



Enter residual tolerance 0.000000

Use flat start ○

Set initial state from keyboard ○

Set Initial State from File ○

Figure 2.9 State Estimation (or Load Flow) Parameters Dialog

Do MATLAB Computation

This option compiles the necessary arrays and file for the calculations and opens MATLAB. The user can then enter MATLAB and run the desired calculation. While any calculation could be run (technically), only the one specified in the program should be run, since some data may not be present for other analyses. For short circuit analysis, type "scc". For load flow analysis, type "loadflow", and to run a state estimation, type "state". **After the routine has run, the user should quit MATLAB, at which time the data will be loaded into BUS for display.**

NOTE: The remaining options are valid during short circuit calculations only.

Show Positive Sequence Diagram

If the zero sequence diagram is currently on the screen, this puts the positive sequence diagram back on the screen.

Show Zero Sequence Diagram

If the positive sequence diagram is currently on the screen, this puts the zero sequence diagram back on the screen.

Show I, V Base Values

If the screen currently contains fault values, this puts the base values for currents and voltages back on the screen.

Show I, V Fault Values

If the screen currently contains base values, this puts the fault values for currents and voltages back on the screen.

Section 3: Creating a System

When the Bus program is run, the user is presented with a blank window with the previously-mentioned menu at the top. At this point, a saved system may be loaded or a new system may be entered. This section will focus on the latter, showing how to enter a simple system. The process is simple.

To add a bus to the system, simply click on any blank portion of the window. The new bus will appear with the bus number. The new bus will be a load bus by default. The sample system to be drawn here has four buses, which will be entered all at once. **A bus can be moved at any time by positioning the cursor and holding the right mouse button down while moving the cursor.** This will move the bus and any lines attached to it.

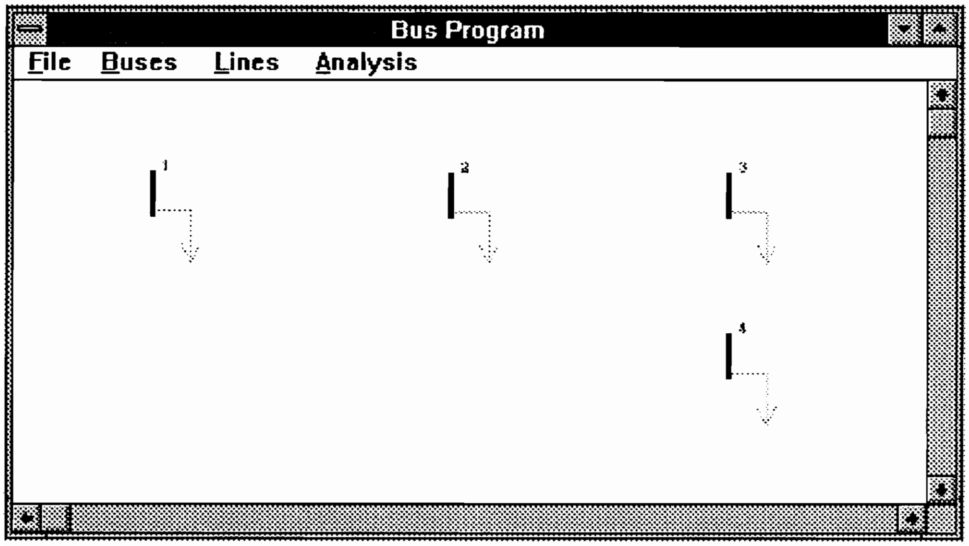


Figure 3.1 Sample System w/Buses Only

As mentioned, the buses are load buses by default. This can be changed by double-clicking on the bus and setting the bus type. This procedure will be covered in detail in the next section. For this system, bus #1 will be a generator; all the others will be load buses.

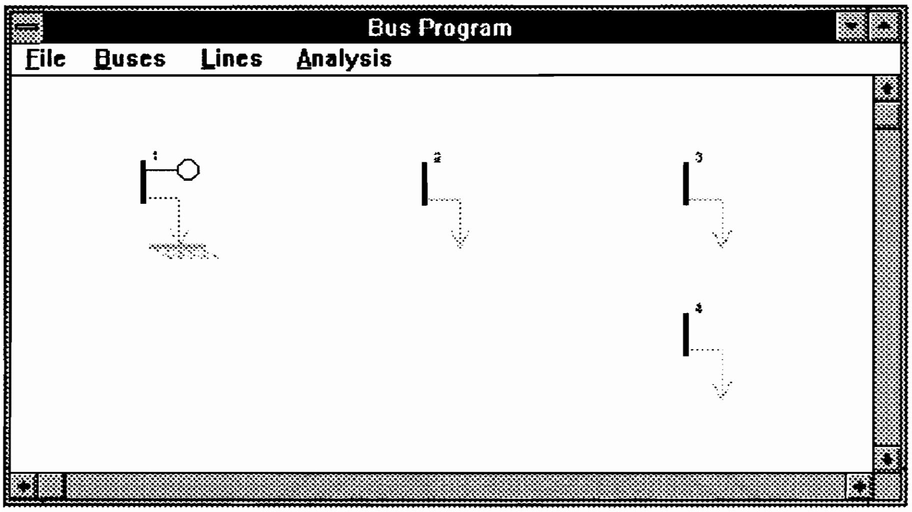


Figure 3.2 Sample System w/Generator

Once the buses have been entered, the lines can be drawn, although it is not required to draw all buses before lines are drawn. To draw a line, highlight one bus of the line by clicking on it and then click on the bus the line is to be drawn to. The termination points of the line are determined by where the user clicks on the buses.

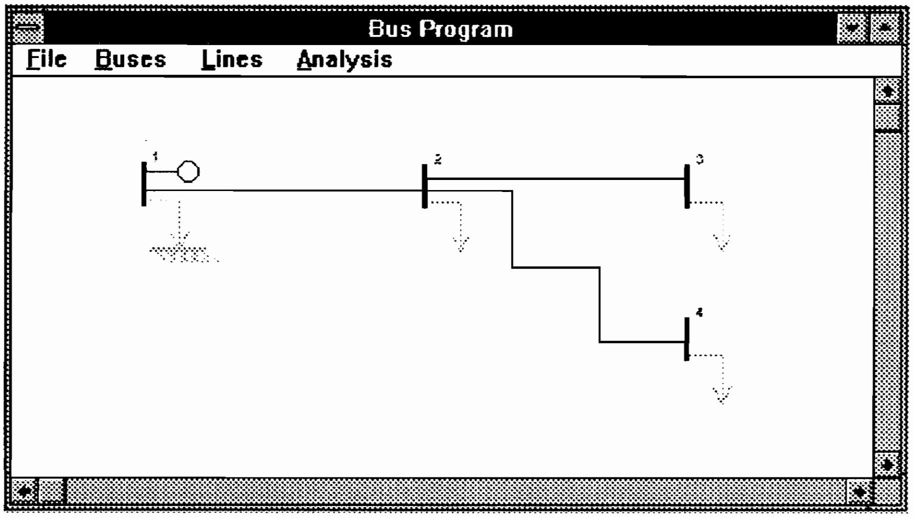


Figure 3.3 Sample System w/Buses & Lines

To add a transformer to the system, draw a line (just like above) where the transformer is to go (or an existing line can be used). Then highlight the line by clicking on it, and press 't'. This will create a transformer, and the original line no longer exists. In this system, a transformer is going to be added between buses #1 and #2. Using the procedure above, the transformer is drawn.

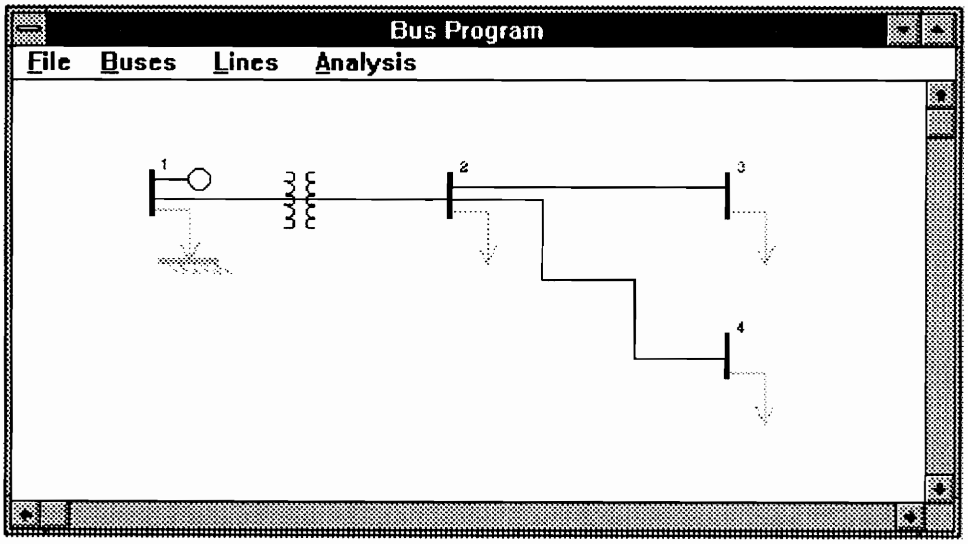


Figure 3.4 Finished Sample System

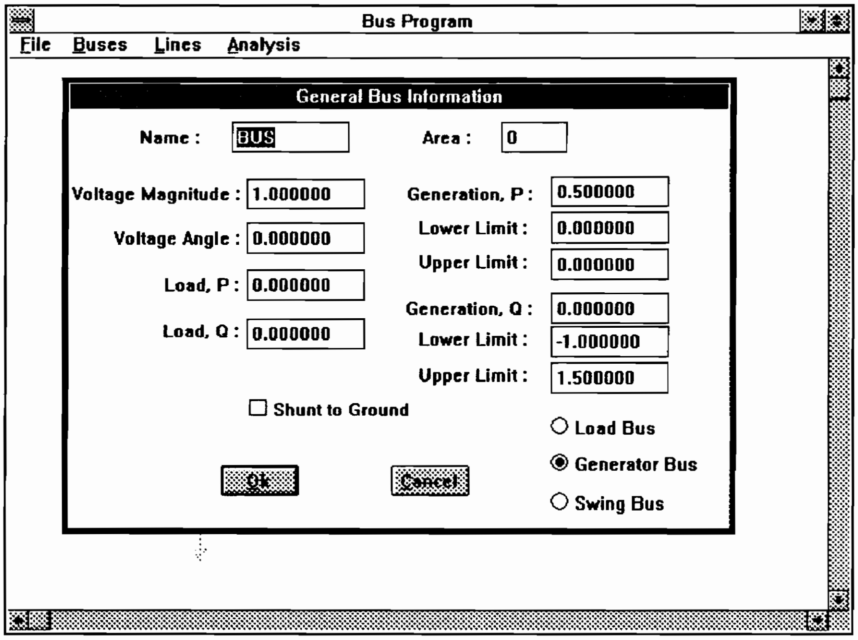
This is basically all there is to drawing a system. These same procedures can be used for a small system like the one above or for large systems with a number of buses, lines, and transformers. Once the system has been drawn, the system data must be entered. That will be the topic of the next section.

Section 4: Entering System Data

Once the system topology has been entered, the parameters must be entered as well. This can be accomplished by double-clicking on the buses and lines. All of the data may be entered this way, although the necessary parameters vary depending on the type of analysis chosen. Thus, it is important that the correct item in the computation type dialog be chosen before data is entered, as some data may have been overwritten if another analysis has been done.

Entering Bus Data

To enter the parameters for a bus, double-click on it. A dialog box will appear. If short circuit analysis or load flow is chosen, the following box will appear.



Name : BUS
Area : 0
Voltage Magnitude : 1.000000
Voltage Angle : 0.000000
Load, P : 0.000000
Load, Q : 0.000000
Generation, P : 0.500000
Lower Limit : 0.000000
Upper Limit : 0.000000
Generation, Q : 0.000000
Lower Limit : -1.000000
Upper Limit : 1.500000
☐ Shunt to Ground
○ Load Bus
● Generator Bus
○ Swing Bus

Figure 4.1 Bus Data Dialog

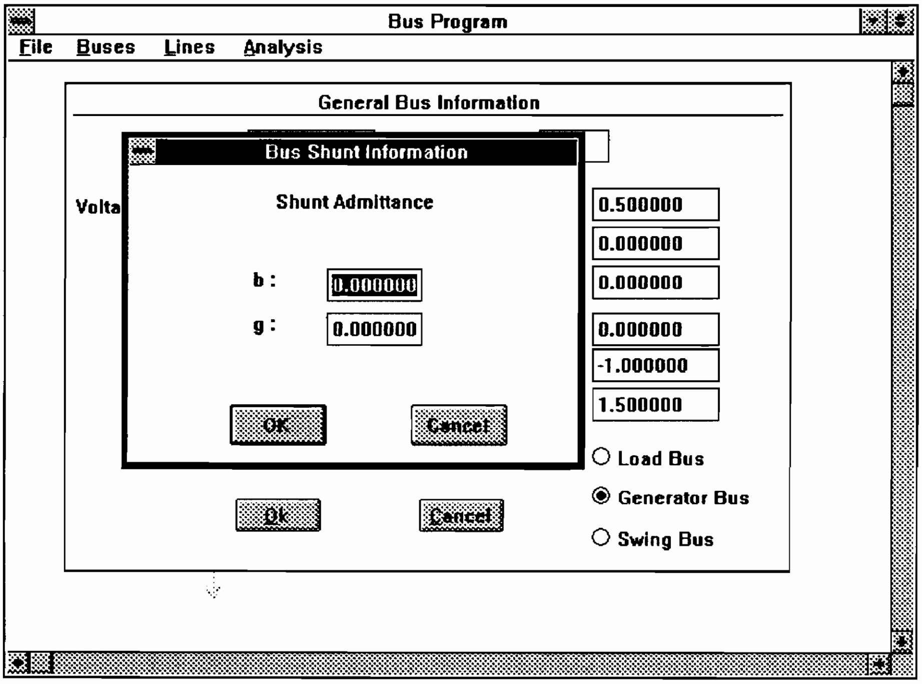
The following data may be entered in this box:

Name:	The bus may be given a name although one is not required.
Area:	If the system is divided into areas, the area # for the bus may be entered here. This is also not required.

All of the following data should be entered in per unit:

Voltage Magnitude:	The voltage magnitude at the bus should be entered here.
Voltage Angle:	Enter the voltage angle at the bus here.
Load P & Q:	If there is a load at the bus, the real and reactive powers can be entered in these two boxes. This data should be given even for Swing buses.
Generation P & Q:	If the bus is a generator bus, the generated real and reactive powers may be entered here. For load flow analysis, the Generated Q should be left zero. Both quantities will be taken as zero for the swing bus.
Lower Limit: Upper Limit:	The generation limits for the bus should be entered here. These parameters must be entered for reactive power in load flow studies.
Shunt to Ground:	If the bus has a shunt to ground, this box should be checked.
Bus Type Buttons:	The bus type (load, generator, swing) should be selected here. Load is the default.

If the Shunt to Ground box is checked, another dialog box will follow:



Shunt Admittance

b : 0.000000

g : 0.000000

0.500000

0.000000

0.000000

0.000000

-1.000000

1.500000

○ Load Bus

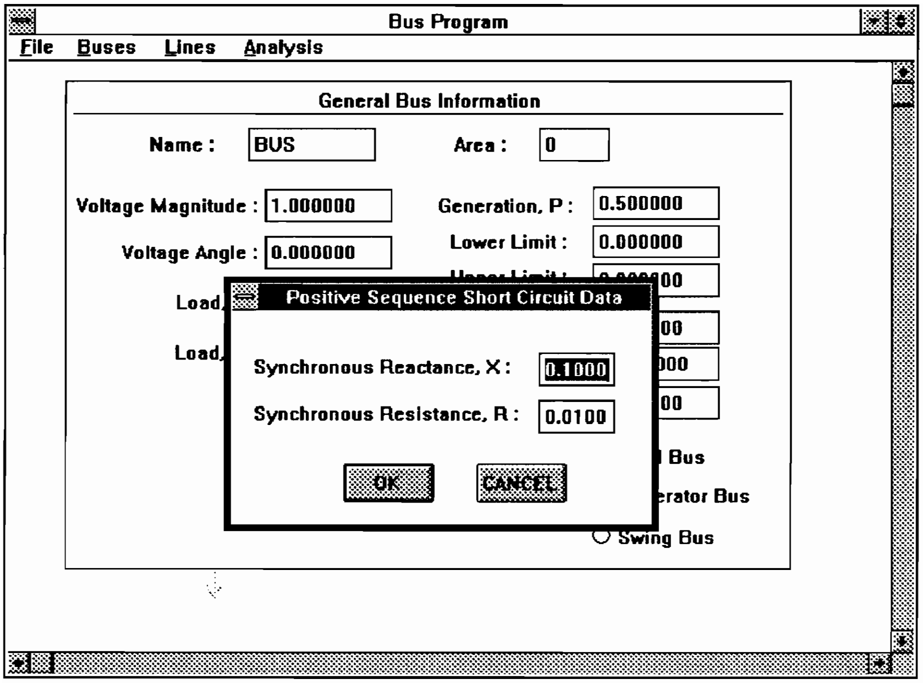
● Generator Bus

○ Swing Bus

Figure 4.2 Shunt Admittance Dialog

The real and reactive parts of the bus shunt admittance should be entered in this dialog.

If the bus has a generator and short circuit analysis is selected, the following dialog box will follow the original dialog (Figure 4.1):



General Bus Information

Name : BUS
Area : 0
Voltage Magnitude : 1.000000
Generation, P : 0.500000
Voltage Angle : 0.000000
Lower Limit : 0.000000

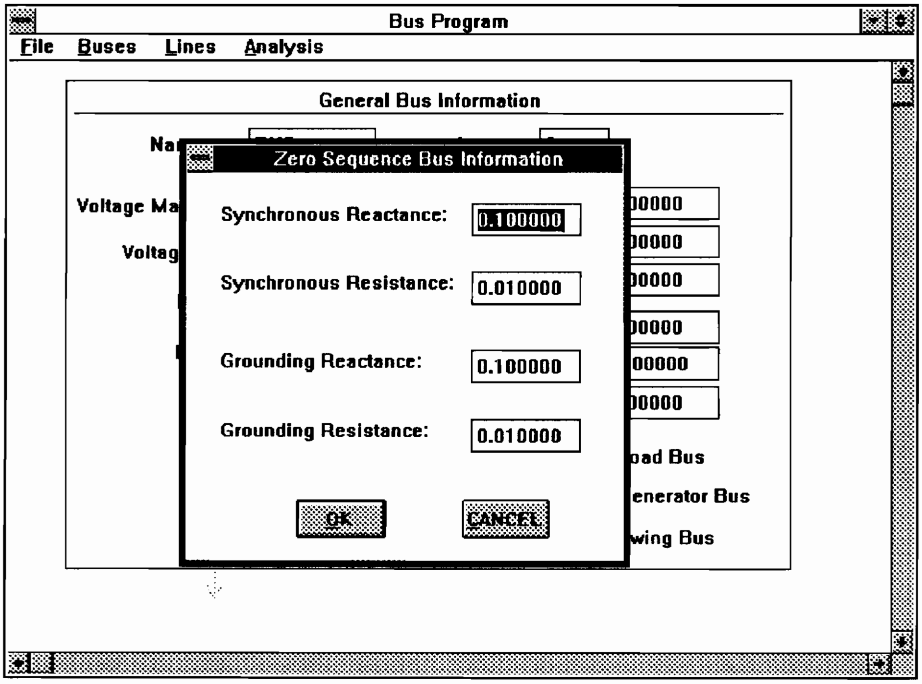
Positive Sequence Short Circuit Data

Synchronous Reactance, X : 0.1000
Synchronous Resistance, R : 0.0100

Figure 4.3 Generator Synchronous Impedance Dialog

In this dialog box, the synchronous reactance and resistance of the generator should be entered for short circuit purposes.

If phase-ground analysis is selected, one more dialog will follow (Figure 4.4). This one will ask for the zero sequence generator data.



Synchronous Reactance: 0.100000

Synchronous Resistance: 0.010000

Grounding Reactance: 0.100000

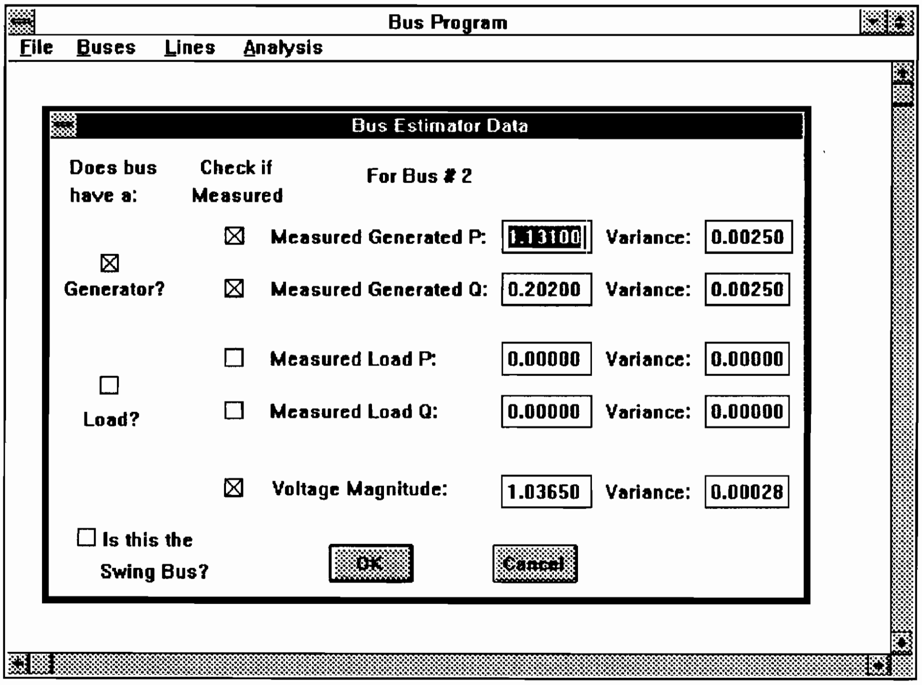
Grounding Resistance: 0.010000

Figure 4.4 Zero Sequence Bus Data Dialog

The parameters to be entered are:

Synchronous Reactance and Resistance:	The per unit zero sequence reactance and resistance of the generator should be entered here.
Grounding Reactance and Resistance:	The grounding reactance and resistance of the generator should be entered here.

If state estimation is the selected analysis, the following dialog box will appear when a bus is double-clicked (instead of Figure 4.1):



Does bus have a:

Check if Measured

For Bus # 2

☒ Measured Generated P: 1.13100 Variance: 0.00250

☒ Generator? ☒ Measured Generated Q: 0.20200 Variance: 0.00250

☐ Measured Load P: 0.00000 Variance: 0.00000

☐ Load? ☐ Measured Load Q: 0.00000 Variance: 0.00000

☒ Voltage Magnitude: 1.03650 Variance: 0.00028

☐ Is this the Swing Bus?

Figure 4.5 Bus Measurement Dialog

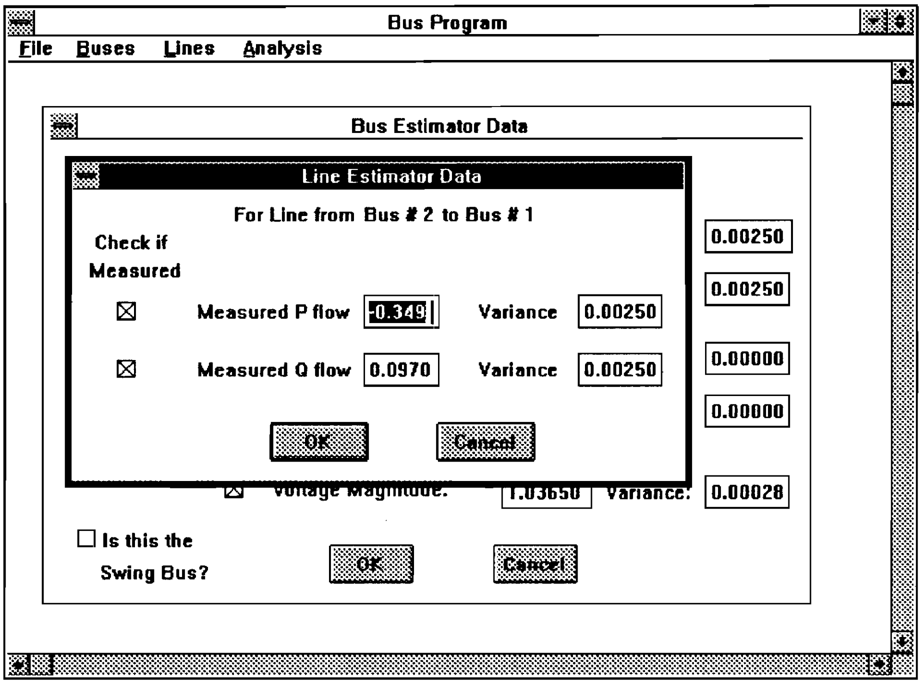
On the left side of the dialog appears a set of check boxes. These are used to make sure that the measurement arrays are built correctly. If the bus has a generator or a load (even if not measured), the appropriate boxes must be checked. If the bus is the swing bus, check the box at the bottom.

The second column of boxes should be checked for each quantity that is measured. The five quantities are:

Generated P and Q
Load P and Q
Voltage Mag.

Each quantity has a box beside it in which the measurement should be entered. To the right of this is a box where the variance of the measurement should be entered. All data should be entered in per unit. These need not be filled in if the measurement box is not checked.

When the previous dialog is cleared, another series of dialogs appears. These are for the line measurements. Consecutive dialogs are opened for each line connected to the selected bus.



Line Estimator Data

For Line from Bus # 2 to Bus # 1

Check if Measured

☒ Measured P flow -0.349 Variance 0.00250

☒ Measured Q flow 0.0970 Variance 0.00250

0.00250

0.00250

0.00000

0.00000

☒ Voltage Magnitude: 1.03650 Variance: 0.00028

☐ Is this the Swing Bus?

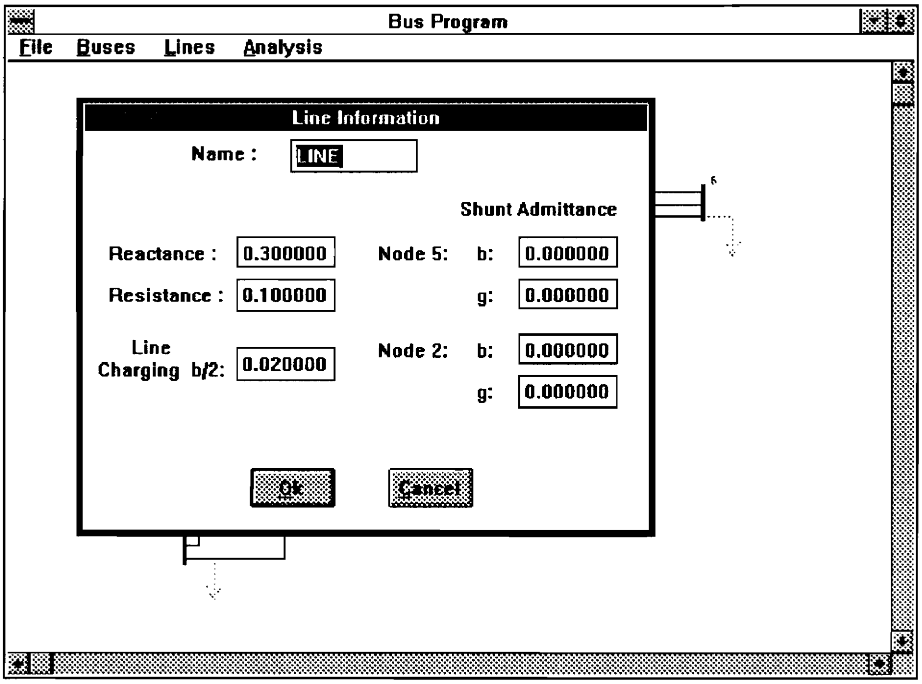
Figure 4.6 Line Measurement Dialog

Again, the following measurements should be entered in per unit.

Measured P flow:	The real power flow along the line in the direction specified.
Measured Q flow:	The reactive power flow along the line in the direction specified.
Variance:	The variance of each specified measurement.

Entering Line Data

To enter line parameters, double-click on a line. The following dialog box will appear. Remember that line measurements for state estimation are entered by double-clicking on a bus, although impedances are still entered by this procedure.



Name : LINE

Reactance : 0.300000

Resistance : 0.100000

Line Charging b/2: 0.020000

Shunt Admittance

Node 5: b: 0.000000

g: 0.000000

Node 2: b: 0.000000

g: 0.000000

Figure 4.7 Line Data Dialog

The following data may be entered (all in p.u.):

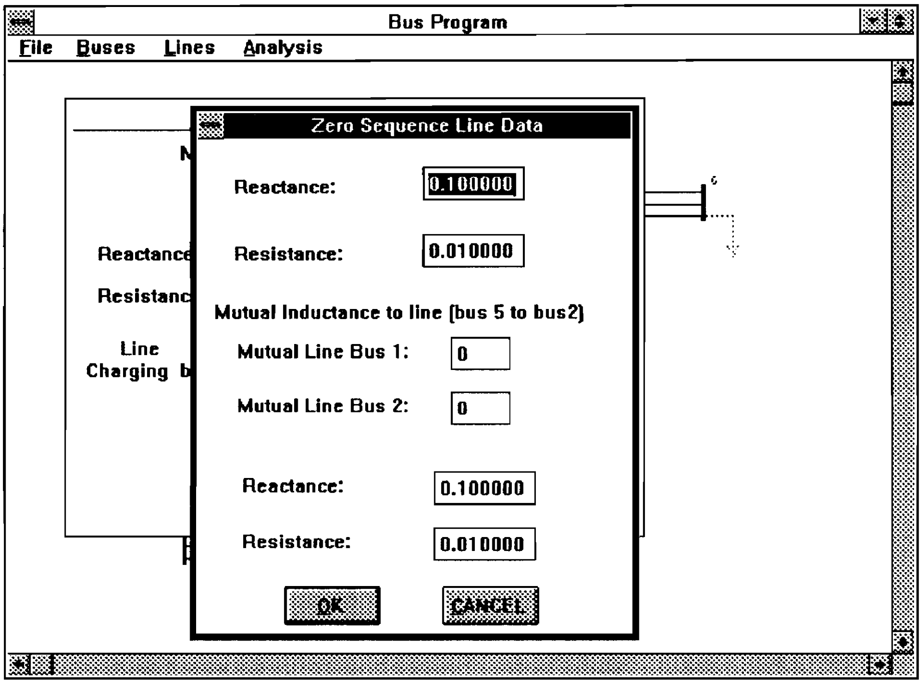
Reactance: The line reactance should be entered here.

Resistance: The line resistance should be entered here.

Line Charging:The charging capacitance (1/2 of line total) should be entered here.

Shunt Admittances: The shunt admittances for each node of the line should be entered here. A box for the conductance and _______ is given for each node.

If phase-ground short circuit analysis is chosen, the following dialog will also appear. This dialog asks for zero-sequence line data.



Reactance: 0.100000

Resistance: 0.010000

Mutual Inductance to line (bus 5 to bus2)

Mutual Line Bus 1: 0

Mutual Line Bus 2: 0

Reactance: 0.100000

Resistance: 0.010000

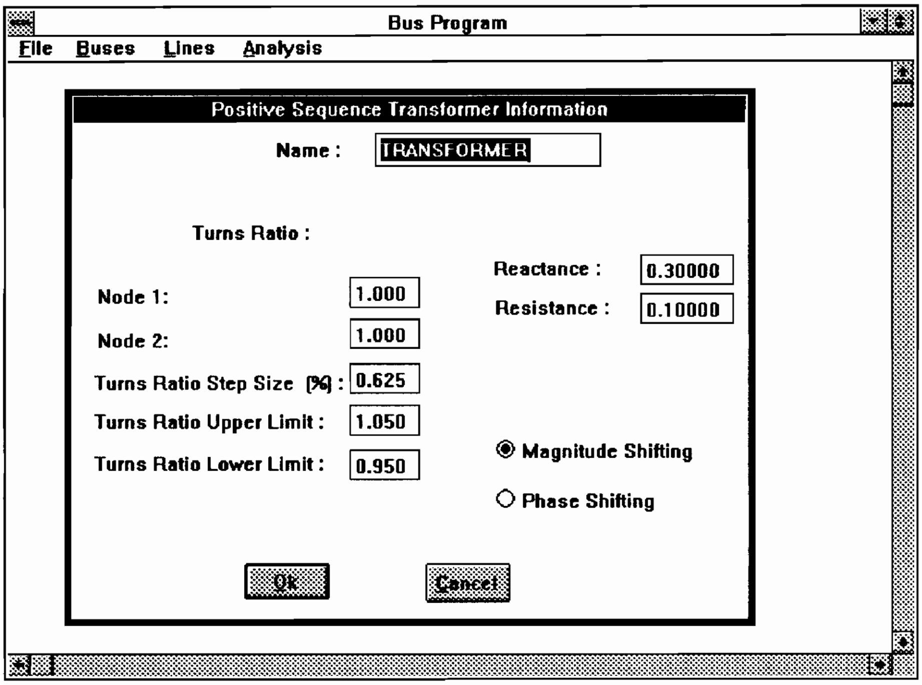
Figure 4.8 Zero Sequence Line Data Dialog

The following may be entered (in p.u.):

Reactance:	The zero-sequence reactance of the line may be entered here.
Resistance:	The zero-sequence resistance of the line may be entered here.
Mutual Line Bus 1:	Node 1 of the coupled line (must be same as node 1 of selected line).
Mutual Line Bus 2:	Node 2 of the coupled line.
Reactance:	The mutual reactance between the two lines.
Resistance:	The mutual resistance between the two lines.

Entering Transformer Data

Transformer parameters may be entered by double-clicking on the transformer just like other elements. This dialog box will appear (for all analyses):



Name : TRANSFORMER

Turns Ratio :

Node 1: 1.000

Node 2: 1.000

Turns Ratio Step Size (%) : 0.625

Turns Ratio Upper Limit : 1.050

Turns Ratio Lower Limit : 0.950

Reactance : 0.30000

Resistance : 0.10000

● Magnitude Shifting

○ Phase Shifting

Figure 4.9 Transformer Data Dialog

The following data may be entered (in p.u.):

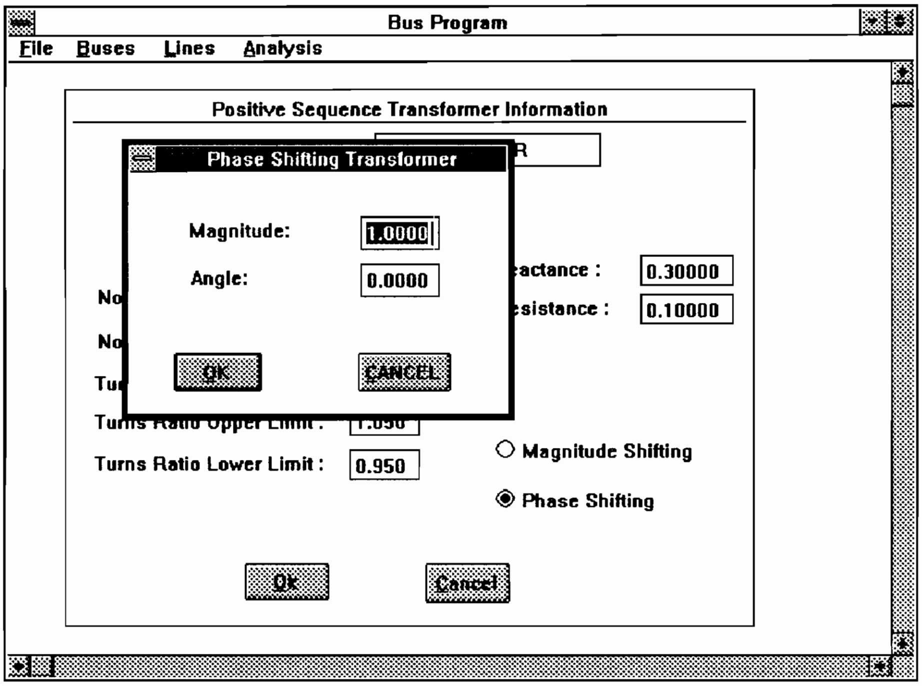
Turns Ratio (for magnitude shifting transformers)

Node 1:	This is the turns at node 1 of the transformer.
Node 2:	This is the turns at node 2 of the transformer. The ratio of the node 1 and node 2 values will be taken.
Turns Ratio Step Size (%):	The step size of the transformer's turns ratio can be entered here.
Upper Limit: Lower Limit:	The limits on the turns ratio of the transformer should be entered here.

Reactance:	The reactance of the transformer should be entered here.
Resistance:	The resistance of the transformer should be entered here.

There are two radio buttons on which the type of transformer can be specified. For a normal transformer, specify magnitude shifting and use the default turns ratio values.

If the transformer is phase-shifting, another dialog appears asking for the phase-shifting parameters.



Magnitude: 1.0000

Angle: 0.0000

actance : 0.30000

sistance : 0.10000

Turns Ratio Lower Limit : 0.950

○ Magnitude Shifting

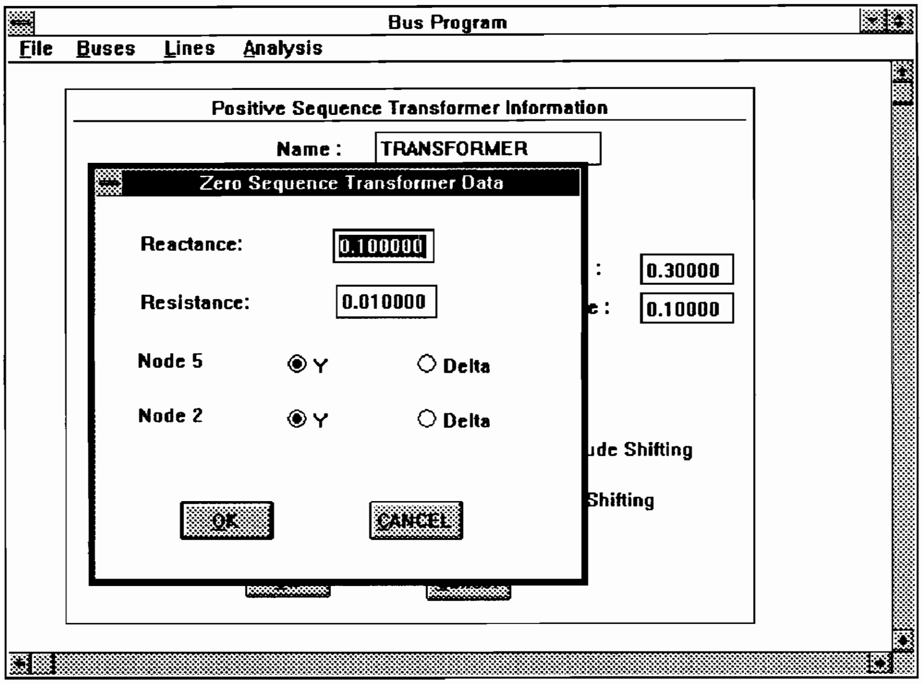
● Phase Shifting

Figure 4.10 Phase-Shifting Transformer Dialog

Two quantities may be entered in this box:

Phase-Shift Magnitude:	The phase-shifting magnitude of the transformer should be entered here.
Phase-Shift Angle:	The angle by which the phase is shifted in the transformer should be entered here.

If phase-ground short circuit analysis is chosen, this dialog will follow (for all transformers):



Name : TRANSFORMER

Reactance: 0.100000

Resistance: 0.010000

Node 5 ● Y ○ Delta

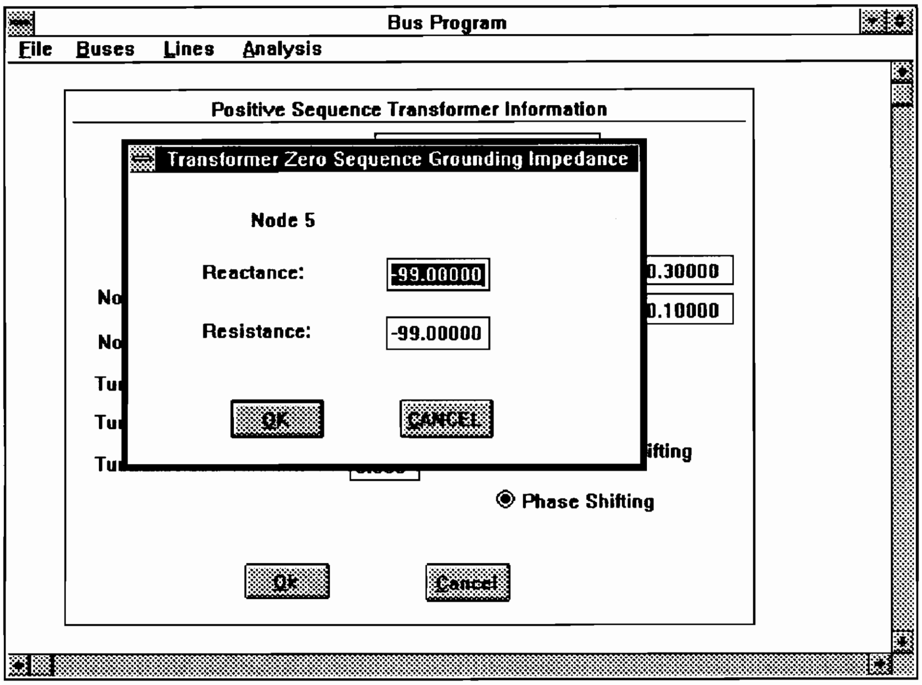
Node 2 ● Y ○ Delta

Figure 4.11 Zero Sequence Transformer Data Dialog

This box asks for the zero-sequence parameters of the transformer. Enter the following data:

Reactance:	The zero-sequence p.u. reactance of the transformer is entered here.
Resistance:	The zero-sequence p.u. resistance of the transformer is entered here.

There are two sets of radio buttons. These should be used to specify the transformer connections at each node. A wye or delta connection may be chosen. For every wye node, one more dialog will be brought up.



Node 5

Reactance: -99.00000

Resistance: -99.00000

● Phase Shifting

Figure 4.12 Zero Sequence Transformer Grounding Impedence Dialog

In this box the p.u. zero-sequence grounding reactance and resistance for the node in question should be entered.

This concludes the data entry process for the system.

Section 5: Running the Analysis

Once the system and data have been entered, the analysis may be run. This section will explain the procedures involved and how to view the results from the analysis.

Short Circuit

To run the short circuit analysis, first make sure that either a three phase or phase to ground analysis is selected in the Computation Type dialog box (Figure 2.8). Then select **Analysis | Do MATLAB Computation**. The program will run MATLAB.

The user must go to the MATLAB window and run the analysis. The command to enter is "scc." The analysis will run and the user will be presented with the prompt when finished. At this time, the user may view one of the variables (type "who" for a list) or return to the program. To do this, type "quit". MATLAB will quit and the results will be loaded in the Bus program.

To view the results, select **Buses | Show All Voltages** and/or **Lines | Show All Currents**. The voltages and currents will be displayed on the one-line diagram. These may be removed from the diagram by selecting the appropriate menu items. For phase to ground faults, the zero sequence diagram may be viewed by selecting **Analysis | Show Zero Sequence Diagram**.

Load Flow

To run a load flow study, make sure that Power Flow Analysis has been selected in Figure 2.8 and the tolerance and initial state data have been entered (Figure 2.9). At this point, select **Analysis | Do MATLAB Computation** and MATLAB will be started.

The user must go into MATLAB and type "loadflow". This will start the computations. When concluded, the results will be shown on screen in table format. This data may be viewed by scrolling the screen and may also be printed by highlighting the tables and selecting **File | Print Selected**. The prompt will then be offered. The user may view a variable or type "quit" to return to the Bus program. This will quit MATLAB and return control to the Bus program. The results may then be viewed.

The bus voltages and angles may be viewed on the diagram by selecting **Buses | Show All Voltages**. The bus powers and line power flows may be viewed by selecting **Buses | Bus Power** and **Lines | View Line Flows**. This will bring up the dialog boxes shown in Figures 2.4 (Bus Power) and 2.6 (Line Flow).

State Estimation

The procedures to run a state estimation are very similar to those for running a load flow study. Follow those procedures until MATLAB is entered. At this point, type "state" to run the state estimation routine.When concluded, the results will be shown on screen in table format. There will be a few differences from the Load Flow tables. This data may also be viewed by scrolling the screen and printed by the same procedure. At the prompt, the variables may be viewed or the program exited. The results may be viewed in the Bus program in exactly the same way as the load flow results. The commands and dialogs for this are the same for both analyses.

Descriptions of MATLAB Matrices

The following is a list of the important variables used in the MATLAB calculations. A list of all variables can be brought up in MATLAB by typing "who" at the prompt. A brief description of each variable is given. Only important variables are listed. All others are for internal use.

Short Circuit

E	Matrix of voltages with real part in first column, imaginary part in second. One row for each bus.
V	Vector of same voltages in form Re + j Im
fault	Fault bus number
ia1	Phase a positive sequence current.
ifault	Fault current, I_f
LINE	Matrix containing line nodes and currents. First two columns are the line nodes. Third column is real part of current, with the imaginary part in the fourth column. One row for each line.
pbuses	Number of positive sequence buses.
rMutual, *iMutual*	Matrices containing real and imaginary parts, respectively, of mutual admittance.
YBusp	Positive sequence admittance matrix for the system.
YBusz	Zero sequence admittance matrix for the system.
ZBus, *ZBusp*	Positive sequence impedance matrix for the system.
ZBusz	Zero sequence impedance matrix for the system.

zbuses	Number of zero sequence buses.

Load Flow

ang	Vector with angles for each bus (in radians).
B	Matrix containing line and transformer susceptances.
Bcap	Matrix containing line charging capacitances (in form of B/2).
buspq	Bus powers entered by the user in the program.
diff	Vector containing the differences between the specified and calculated bus powers.
dx	Change in state vector.
deltax	Change in state vector resulting from calculations (may not contain all state variables).
E	Vector with voltages for each bus.
func	Calculated bus powers.
G	Matrix containing line and transformer conductances.
H	Jacobian matrix.
lflow	Line flows sorted for display.
limit	Generator Q limits.
loadpq	Vector with load powers at each bus.
p, q	Load P and Q at each bus.
pgen, qgen	Generated P and Q at each bus.
pmis, qmis	P and Q mismatch at each bus.
React	Matrix containing line and transformer reactances.

Resist	Matrix containing line and transformer resistances.
swing	Swing bus number.
tolerance	Error tolerance
x	State vector.

State Estimation

ang	Vector with angles for each bus (in radians).
B	Matrix containing line and transformer susceptances.
Bcap	Matrix containing line charging capacitances (in form of B/2).
cov	Estimate variance matrix.
dx	Change in state vector.
E	Vector with voltages for each bus.
err	Difference between measurements and calculated measurements.
func	Calculated measurements.
G	Matrix containing line and transformer conductances.
H	Jacobian matrix.
J	Measurement residual.
K	Number of degrees of freedom in the chi-squared distribution.
lflow	Line flows sorted for display.
mtype	Vector of measurement types (1 for line P flow, 2 for line Q flow, 3 for bus P, 4 for bus Q, 5 for voltage magnitude).
norm	Normalized residuals.
numbus	Number of buses.

p, q	Total P and Q at each bus.
r	Vector of variances.
R	Variance matrix
React	Matrix containing line and transformer reactances.
Resist	Matrix containing line and transformer resistances.
swing	Swing bus number.
tJ	Limit on J set by chi-squared distribution.
tolerance	Error tolerance
x	State vector.
z	Vector of measurements.

Listing of MATLAB Files

The following is a list of the files created for use in MATLAB. The files generated depend on the analysis that has been selected. The contents of these files can be viewed in any text editor.

General Files

File	Contents
genbus.m	Generator bus info.
Hbus.m	Inertia data for generator buses.
linedat.m	Line nodes and admittances.
linemat.m	Line matrix in 2x2 blocks.
shunt.m	Shunt admittances.
swingbus.m	Swing bus info.
xbus.m	Synchronous reactances for stability analysis (not used).

Short Circuit

mutualb.m	Mutual admittance matrices.
ybus3p.m	Ybus matrix for three phase analysis.
ybuspg.m	Ybus matrix for phase to ground analysis.

Load Flow

bustype.m	Array with bus quantities to be computed. 3 represents real power,

	4 is for reactive power.
busnode.m	Matrix with bus numbers for the quantities in bustype.m.
linetype.m	Array with quantities to be computed for line flows. 1 represents real power flow, 2 represents reactive power flow.
lnode.m	Matrix with node numbers for the quantities above.
loads.m	Array with loads at the buses.
power.m	Array containing bus powers entered by the user.
qlim.m	Reactive generation limits at the buses.
xinit.m	Initial state vector.
x0.m	Directs MATLAB to use xinit.m or other file with initial state.

State Estimation

linetype.m	Same as linetype.m file above.
lnode.m	Same as lnode.m file above.
meas.m	Measurement array.
nodes.m	Node (bus) numbers for measurements.
types.m	Array of measurement types. 1 is for real line flow, 2 is reactive line flow, 3 represents real bus power, 4 is for reactive bus power, while 5 designates a voltage measurement.
var.m	Array of measurement variances.
xinit.m	Same as load flow xinit.m, although state vector is different.
x0.m	Same as previous x0.m.

Impedance files for Load Flow and State Estimation

charg.m	Line charging matrix.
react.m	Reactance matrix.
res.m	Resistance matrix.
yshunt.m	Arrays of shunt admittances.

Vita

Steven D. Estes was born in 1971 in Hickory, North Carolina. He received a Bachelor of Science in Electrical Engineering degree from UNC Charlotte in May of 1994 and entered the Master's degree program in electrical engineering at Virginia Tech in August of 1994. Steve is a member of Tau Beta Pi and IEEE.