Distributed Multi-Phase Distribution Power Flow: Modeling, Solution Algorithm, and Simulation Results

A Thesis

Submitted to the Faculty

of

Drexel University

by

Michael R. Kleinberg

in partial fulfillment of the

requirements for the degree

of

Master of Science in Electrical Engineering

January 2007

DEDICATIONS

This thesis is dedicated to my family:

Robert, Mary Kate, Christopher, Nikki,

Colleen, and Bridget Kleinberg.

I love you all so much and thank God

everyday for blessing me with you.

ACKNOWLEDGMENTS

I would like to acknowledge and thank my advisor, Dr. Karen Miu, for all of her guidance in performing this research and in the completion of this thesis. Your constant and continued support and encouragement has made this work possible. Thank you for believing in me.

I would also like to thank Dr. Nwankpa and Dr. Fischl for serving on my thesis committee. Thank you for bestowing upon me your expertise and passion for the field of electric power engineering and for your time dedicated to reviewing and commenting on this work.

Lastly, I would like to thank all of my fellow students at the Center for Electric Power Engineering. Thank you for sharing with me your excitement and thirst for knowledge and for always being willing to assist me in working through a tough problem or providing me with the occasional much needed distraction.

TABLE OF CONTENTS

LIST OF TABLES vii

LIST OF FIGURES viii

ABSTRACT x

1 INTRODUCTION 1

1.1 MOTIVATION 2

1.2 BACKGROUND 4

1.3 PROBLEM STATEMENT 5

1.4 ORGANIZATION OF THESIS 6

2 REVIEW OF MULTI-PHASE DISTRIBUTION SYSTEM COMPONENT MODELS AND NETWORK EQUIVALENCING TECHNIQUES 7

2.1 REVIEW OF DISTRIBUTION SYSTEM COMPONENT MODELS 7

2.1.1 LINE MODELS 7

2.1.2 NETWORK SWITCH MODEL 9

2.1.3 TRANSFORMER MODELS 9

2.1.4 LOAD MODELS 10

2.1.5 SHUNT CAPACITOR MODEL 12

2.1.6 DISTRIBUTED GENERATOR MODEL 13

2.2 REVIEW OF NETWORK EQUIVALENCING TECHNIQUES 14

2.2.1 WARD INJECTION METHOD 15

2.2.2 REI EQUIVALENTS 17

2.2.3 APPLICATION TO DISTRIBUTED, DISTRIBUTION SYSTEM ANALYSIS 18

3 COMPONENT AND SYSTEM MODELS FOR DISTRIBUTED ANALYSIS......20

3.1 NETWORK PATITIONS..20

3.2 EQUIVALENT SOURCE MODEL..23

3.3 EQUIVALENT LOAD MODELS...25

3.3.1 EQUIVALENT LOAD MODEL FOR PARTITIONS WITH A GROUNDED EQUIVALENT SOURCE BUS..26

3.3.2 EQUIVALENT LOAD MODELS FOR PARTITIONS WITH AN UN-GROUNDED EQUIVALENT SOURCE BUS..30

3.3.3 REPRESENTING EQUIVALENT LOADS ON THE EQUIVALENT LOAD BUS 31

3.4 DISTRIBUTED SYSTEM MODEL...32

3.5 EXTENSIONS FOR MESHED SYSTEMS 37

4 DISTRIBUTED MULTI-PHASE POWER FLOW ALGORITHIM........................42

4.1 SOLUTION ALGORITHM 42

4.1.1 OPTIMAL PARTITION CALCULATION ORDER 42

4.1.2 SOLUTION ALGORITHM 45

4.1.3 TIMING CONSIDERATIONS 48

4.1.4 CONVERGENCE CRITERIA 50

4.2 IMPLEMENTATION 52

4.2.1 SOFTWARE ARCHITECTURE 53

4.2.2 USER INTERFACE 55

5 CONVERGENCE PROPERTIES 57

5.1 IMPLICIT Z-BUS GAUSS POWER FLOW 57

5.2 CONVERGENCE PROPERTIES 59
6 SIMULATION RESULTS 65
6.1 SIMULATION SET-UP 66
6.2 UN-PARTITIONED ORIGINAL SYSTEM 67
6.3 THREE PARTITION SYSTEM 68
6.4 FIVE PARTITION SYSTEM 73
6.5 OBSERVATIONS ON SIMULATION RESULTS 74
7 CONCLUSIONS 77
7.1 CONCLUSIONS 77
7.2 SUMMARY OF RESEARCH CONTRIBUTIONS 78
7.3 FUTURE WORK 79
LIST OF REFERENCES 81
A APPENDIX A: LIST OF SYMBOLS 84
B APPENDIX B: TRANSFER CONTROL PROTOCOL 87

LIST OF TABLES

Table 2.1: Nominal load value calculations [7] 12

Table 2.2: Capacitor/Load admittance matrices [7] 13

Table 6.1: Overview of simulations 66

Table 6.2: 97 bus test system – component count 68

Table 6.3: Three partition system results summary 72

Table 6.4: Five partition system results summary 74

Table 6.5: Summary of simulation results 75

LIST OF FIGURES

Figure 2.1: Representation of internal/external networks 15

Figure 3.1: Diagram of a 97 bus, 249 node partitioned distribution system 23

Figure 3.2: Example of equivalent source buses in the 97 bus test system.......................24

Figure 3.3: Example of an equivalent load bus in the 97 bus test system.........................26

Figure 3.4: Equivalent current injection example 28

Figure 3.5: Single-phase 5 bus system 33

Figure 3.6: Partitioned single-phase 5 bus system – equivalent admittance 34

Figure 3.7: Partitioned single-phase 5 bus system – equivalent power injection.............35

Figure 3.8: Partitioned single-phase 5 bus system – equivalent current injection 35

Figure 3.9: General n_b bus, n_n node, n_p partition radial system 37

Figure 3.10: Meshed system which can be partitioned in a radial structure 38

Figure 3.11: Meshed system with radial structure of partitions 38

Figure 3.12: Meshed system which cannot be partition into a radial structure.................39

Figure 3.13: Meshed system with meshed structure of partitions 39

Figure 3.14: Meshed partition with multiple equivalent sources 41

Figure 4.1: Partition indexing scheme applied to the 97 bus test system..........................45

Figure 4.2: Flow chart of distributed power flow algorithm 47

Figure 4.3: Computation and communication times for 5 partition system 49

Figure 4.4: Example software architecture between three partitions 53

Figure 4.5: Matlab script node implementation..54

Figure 4.6: Graphical user interface at each location 55

Figure 5.1: Single-phase 5 bus system 62

Figure 5.2: Partitioned single-phase 5 bus system 62

Figure 6.1: 97 Bus, 249 node test distribution system..67

Figure 6.2: Test system partitioned into three partitions...69

Figure 6.3: Three partition simulation - constant impedance load models 70

Figure 6.4: Three partition simulation - constant current load models 71

Figure 6.5: Three partition simulation - constant power load models...............................72

Figure 6.6: Test system partitioned into five partitions...73

Figure B.1: TCP/IP segment[32] 87

ABSTRACT

Distributed Multi-Phase Distribution Power Flow: Modeling, Solution Algorithm, and Simulation Results

Michael R. Kleinberg
Karen Miu, Ph.D.

With the increasing presence of distributed intelligence throughout power distribution systems, the possibilities for distributed control and operation schemes are becoming progressively more attractive and feasible. Distributed operations will require tools to properly assess and predict the present and future status of the system in order to make proper control decisions. Multi-phase distribution power flow is a basic tool which calculates the operating state of the distribution system and is used to support all other applications. Therefore, this thesis will present a new method for calculating distribution power flow using physically remote distributed processors.

The proposed power flow requires the distribution system to be partitioned with each partition distributed to a remote processor. Each processor then only requires detailed information about the portion of the network it will represent. Distributed analysis component models for multi-phase distribution systems have been developed to model the remaining network not explicitly retained in each partition. These models are embedded in a new distributed algorithm for multi-phase distribution power flow. Properties of the converged solution of this algorithm have been investigated and will be reported. A distributed processor test bed was designed to emulate the distribution of intelligent devices throughout power distribution networks and simulations were conducted to asses the proposed algorithm. Results have shown the proposed method

will converge to the same solution as that of an un-partitioned traditional power flow validating the accuracy of the proposed models and algorithm.

1 INTRODUCTION

Power distribution systems serve as the direct link to electric power for the majority of residential, industrial, institutional, and agricultural power consumers world wide. The operation of these systems in a manner which avoids equipment damage while maintaining low real power losses and high levels of reliable service has become a necessity for both economic and social stability in modern society. As such, system operators must find ways to meet high levels of reliability under ever increasing demands.

To achieve these goals, modern distribution systems have begun to implement distribution automation schemes. The Institute of Electrical and Electronics Engineers (IEEE) defines distribution automation as a system that enables an electric utility to monitor, coordinate, and operate the distribution system in real-time from a remote location [1]. These systems allow for remote system managers as well as local automatic controllers to change system parameters and reconfigure network topology. The objective of these initiatives is to operate the system in the most efficient and reliable manner possible which requires minimizing real power losses while maximizing the total amount of load served. In order for remote system managers and local controllers to make proper control decisions, they will require information on the current state of the system.

Distribution power flow is a method for determining the operating state of the distribution system. This thesis aims to present a new method for calculating distribution power flow using remotely distributed processors. The distribution of the processors is meant to mimic the increasing presence of distributed intelligent devices throughout the distribution system. By distributing the power flow problem in this way, shared local information such as measurements, component status, and computation results will allow for each device in the network to attain a global view of the current operating state of the system. Coordination between the local and system wide controllers operating the network can then be facilitated. For these reasons, the proposed application of this work is in the field of distribution system automation and control.

1.1 MOTIVATION

The multi-phase distribution power flow problem is the process of calculating the steady state operating point of a multi-phase unbalanced power system given the network topology, generation profile, and load profile. The power flow problem forms the basis of all distribution system planning, operation, and optimization schemes.

Traditionally, system operations are performed at a centralized distribution control center on a powerful computer optimized for such calculations. In modern distribution systems, intelligent devices are distributed throughout the network which are capable of performing monitoring, computation, communication, and control operations. By utilizing the computation and communication capabilities of these devices, the burden of solving the computationally intensive distribution power flow problem may be ceded to the devices involved in actually controlling the network. This will in turn facilitate coordination between these controllers and allow each to acquire knowledge of the whole system in addition to their local area. With each local controller having a global view of the system, distributed control and monitoring schemes for distribution management and automation become increasingly attractive and feasible.

A large body of literature is available which supports the implementation of distribution system management applications such as capacitor control, load balancing, and network reconfiguration. These applications reduce real power losses, improve reliability, and maximize the amount of real power served by the network [2, 3]. Despite their proven advantages, full implementation of these schemes has been limited in number. Some obstacles have been a lack of sufficient computational capability and sophisticated electric power hardware required to compute and implement proper control decisions. These are problems which can be overcome by utilizing the processing power now located throughout the network in intelligent hardware.

Some examples of this intelligent hardware found in modern distribution systems are feeder terminal units (FTU), distributed generators (DG), and network protection relays. These devices have onboard computer processing units (CPU) and are capable of local monitoring, actuation, and communication. The local control actions of these devices base decisions on local measurements and provide actuation to improve local operating conditions. These types of actions however may not lead to globally desirable operating points or procedures. For example, control actions happening at different locations without coordination can lead to hunting of control devices. Increased information about the state of the whole system will then allow for coordination between these devices and result in more effective control decisions. To achieve this each device must be in communication with the others in the network. A way to facilitate this required need for increased system wide information at each location is through a distributed power flow analysis.

1.2 BACKGROUND

The distributed analysis of distribution systems has been investigated in the past. The first attempts were aimed at decreasing computation time of the power flow problem by calculating the solution at a central location on multiple processors. Later methods, while not solving the distribution power flow problem, investigated the distributed analysis of distribution systems for control and state estimation applications. Existing methods relevant to this thesis will be reviewed, highlighting pertinent points and exposing differences in techniques.

In [4], a parallel radial distribution power flow was implemented in which the network was partitioned by feeders and processed on a transputer using multiple processors. The power flow was calculated as a course-grain parallel computation using the power injected into each feeder as the system variables. In the partitioning process, the optimal number of feeders to be solved by each processor was designed so as to balance the computational load each would have to bear. Results were communicated across processors on the transputer to solve the power flow. This method was a centralized scheme aimed solely at reducing computation time. In contrast to this method, the proposed distributed power flow will partition the system not based on optimizing calculations but rather on the physical distribution of intelligent devices throughout the network, hence communication between processors will take place over a communications network.

More recently, in [5], an adaptive protection scheme based on coordination and communication between multi-agents was proposed which focused on dynamically determining relay settings. The proposed method relies, much like the work in this thesis, on the capabilities of the built-in intelligence now present in distribution system components. Specifically, the work in [5] focuses on microprocessor based relays connected together through a communication system. The combination of processing power and communication capabilities allow for each distributed device to react and adapt to changing system conditions while maintaining coordination through communication with other relays in the network. As opposed to the work in [5] which presents a specific control strategy for distributed applications, this thesis will present a method by which distributed controllers may obtain a global view of the operating state of the system.

In [6], a distributed state estimator was presented which aimed to increase computational efficiency and improve estimation reliability for avionic and naval power systems. The converter coupling points in a DC zonal system seen in avionic and naval power systems were used as natural boundaries to create sub-grids by which to distribute the network. The distributed state estimator was then based on a decentralized Kalman filter technique which involved making distributed local estimations and then communicating and assimilating them together. The idea of using network device locations to create natural boundaries is employed in this thesis as well. The focus of this work will however be on large scale, three-phase distribution systems and distribution power flow analysis.

1.3 PROBLEM STATEMENT

This thesis will present models and a method to solve the distribution power flow problem using physically distributed processors. Specifically, the thesis will present new component models for distributed analysis of multi-phase distribution systems, a distributed power flow solution algorithm, and an investigation of the convergence properties of the algorithm. To verify the proposed work, simulation results from the application of the models and method on a test distribution system will be presented.

1.4 ORGANIZATION OF THESIS

The thesis will progress in the following manner. Chapter 2 will discuss existing multi-phase distribution power flow component models and traditional network equivalencing techniques. Then, Chapter 3 will present newly proposed partition models, distributed analysis component models, and a distributed system model. Chapter 4 uses the proposed models and provides the details of a solution algorithm for distributed multi-phase distribution power flow analysis and presents an overview of an implementation of the method using commercially available software. Chapter 5 will present convergence properties of the solution algorithm and Chapter 6 will present simulation results from the implementation of the proposed method on test distribution systems. Lastly, Chapter 7 will provide final conclusions on the work highlighting advances in the research topic along with suggestions for future work in this research area.

2 REVIEW OF MULTI-PHASE DISTRIBUTION SYSTEM COMPONENT MODELS AND NETWORK EQUIVALENCING TECHNIQUES

In order to perform a distributed power flow analysis of a distribution system, the network is first divided into partitions. Each partition then contains a detailed model of a portion of the network which it represents. The effects of the topology, load, and generation of the remaining network, not explicitly retained in the respective partition, will be modeled through equivalents. State-of-the-art multi-phase distribution power flow component models [7] for lines, loads, transformers, shunt capacitors, and network switches are used to model the system within each partition and will be reviewed in this chapter. Network equivalents are required to model the system not contained in each partition, as such, two existing methods which were investigated for use in this work, the REI and Ward Injection methods, will be reviewed.

2.1 REVIEW OF DISTRIBUTION SYSTEM COMPONENT MODELS

In this section, multi-phase distribution system component models [7] will be presented. For each component, a description of its main function and a brief discussion of its common local and network controls will be discussed. This will be followed by the component's power flow model used in this thesis.

2.1.1 LINE MODELS

Distribution lines are used to transmit electric power and comprise the majority of branches in a distribution system. They are for the most part passive elements of the network. While their parameters may change due to varying operating or environmental conditions, these changes are not under the jurisdiction of any local controller. Line operating conditions can however be monitored by feeder terminal units and network protection relays and are useful measures for determining proper control decisions.

The three-phase π-line model as presented in [8] and the necessary extensions for use in multi-phase and ungrounded distribution systems as presented in [7] are used in this thesis. The model for a line connecting bus *i* to bus *k* can be seen below in (2.1) where $\boldsymbol{Z}_{ik}$ is the series impedance of the line and $\boldsymbol{Y}_{ik}^{sh}$ is the shunt admittance at each end of the line modeling line charging. $\boldsymbol{Z}_{ik}$ and $\boldsymbol{Y}_{ik}^{sh}$ are (n_{ph} x n_{ph}) matrices, where n_{ph} is the number of phases of the line. Typically for distribution systems, lines are less than 50 miles long, hence line charging shunts will be neglected [9].

$$\boldsymbol{Y}_{ik}^{phase} = \begin{bmatrix} \boldsymbol{Z}_{ik}^{-1} + \frac{1}{2}\boldsymbol{Y}_{ik}^{sh} & -\boldsymbol{Z}_{ik}^{-1} \\ -\boldsymbol{Z}_{ik}^{-1} & \boldsymbol{Z}_{ik}^{-1} + \frac{1}{2}\boldsymbol{Y}_{ik}^{sh} \end{bmatrix} \tag{2.1}$$

In ungrounded portions of the network, because there is no ground reference, line-to-line voltages are chosen as the state variables. In this thesis, V^{ab} and V^{bc} are chosen as the state variables with V^{ca} being redundant and equal to $-(V^{ab}+V^{bc})$. The dimension of $\boldsymbol{Y}_{ik}$ is then reduced from a (3 x 3) to a (2 x 2) as seen in [7].

2.1.2 NETWORK SWITCH MODEL

Sectionalizing and tie switches are located throughout the network for the purpose of reconfiguration for load balancing and service restoration. As the distribution system becomes more automated, switching actions will increasingly be dictated by both remote and local controllers. Currently, in some distribution systems, switches coordinate in order to execute switching sequences based on measurements and pre-determined control actions. With a method of determining pre- and post-reconfiguration states, coordination with other network devices will allow for on-line control of network reconfiguration. This will lead to more informed automated procedures for network optimization in terms of reducing real power loss and maximizing service restored to customers.

In this thesis, network switches are modeled as short distribution lines with small series resistance and zero reactance or as ideal zero impedance connections between two buses.

2.1.3 TRANSFORMER MODELS

Transformers are used to step-up and step-down voltages within the distribution network. Typically, voltages are stepped down from transmission levels at the substation for use in the distribution system and then used throughout the network to change between distribution level voltages and lastly to step voltages down to service levels used by customers. The tap settings of the transformers are a parameter which can be controlled via a local or network controller. The majority of transformers in the distribution system are subject to off-line tap changing, although some have the capability to change taps while serving load. Changing tap settings allows system constraint violations such as branch current flow or bus voltage limits to be alleviated. Controlling these actions through a coordinated procedure allows for a system wide approach to making these decisions.

A transformer connecting bus *i* and bus *k* can be modeled using the admittance matrix

$$\boldsymbol{Y}_k^{xfmr} = \begin{bmatrix} \boldsymbol{Y}_k^{pp} & \boldsymbol{Y}_k^{ps} \\ \boldsymbol{Y}_k^{sp} & \boldsymbol{Y}_k^{ss} \end{bmatrix}. \tag{2.2}$$

The values and dimensions of the entries of the admittance matrix (2.2) will depend upon: the number of phases, the connection type, the leakage admittance, the primary tap setting, and the secondary tap setting. If one side of the transformer is ungrounded, for example, line to line voltages V^{ab} and V^{bc} are used, reducing the dimension of the corresponding diagonal elements of $\boldsymbol{Y}_k^{xfmr}$ to (2 x 2). Also, the dimension of the off-diagonal entries must be adjusted accordingly. This concept can be applied to each connection possibility and is explained in detail in [7].

2.1.4 LOAD MODELS

In modern distribution systems, load controllers can react automatically to changes in the operating point of the network. These controls, such as load shedding and motor protection, may instruct the load to vary a parameter or disconnect from the network completely. Through communication with other load and network controllers, these operations can be conducted in a more coordinated manner which may result in fewer improper load protection relay operations and improved load parameter tuning.

Static load models are used for power flow analysis in this thesis and are modeled as constant impedance, constant current, or constant power. Loads are specified for a nominal value given a nominal operating voltage. Three-phase loads may be connected as grounded wye or delta in grounded portions of the network and as delta in ungrounded portions of the network. Table 2.1 shows the equations for calculating the load parameters based upon the specified nominal values of load power $\bar{\boldsymbol{S}}_{Lk}$, current $\bar{\boldsymbol{I}}_{Lk}$, or admittance $\bar{\boldsymbol{y}}_{Lk}$. The following notation is used:

superscript *	:	complex conjugate
subscript $_{nom}$	:	nominal value
./	:	element-wise division
$\bar{\bullet}$	:	specified value

Table 2.1: Nominal load value calculations [7]

<table>
<tr><th>Load Connections</th><th>$\boldsymbol{V}_{k,nom}$</th><th>$\boldsymbol{S}_{k,nom}$</th><th>Load Type</th><th>Load Parameter Value</th></tr>
<tr><td rowspan="3">Grounded Wye</td><td rowspan="6">$\begin{bmatrix} V^a_{k,nom} \\ V^b_{k,nom} \\ V^c_{k,nom} \end{bmatrix}$</td><td rowspan="3">$\begin{bmatrix} S^a_{k,nom} \\ S^b_{k,nom} \\ S^c_{k,nom} \end{bmatrix}$</td><td>Constant S</td><td>$\bar{\boldsymbol{S}}_{Lk} = -\boldsymbol{S}_{Lk,nom}$</td></tr>
<tr><td>Constant I</td><td>$\bar{\boldsymbol{I}}_{Lk} = -(S_{Lk,nom}./\boldsymbol{V}_{k,nom})^*$</td></tr>
<tr><td>Constant Z</td><td>$\bar{\boldsymbol{y}}_{Lk} = \boldsymbol{S}^*_{Lk,nom}./\left|\boldsymbol{V}_{k,nom}\right|^2$</td></tr>
<tr><td rowspan="6">Ungrounded Delta</td><td rowspan="6">$\begin{bmatrix} S^{ab}_{k,nom} \\ S^{bc}_{k,nom} \\ S^{ca}_{k,nom} \end{bmatrix}$</td><td>Constant S</td><td>$\bar{\boldsymbol{S}}_{Lk} = -\boldsymbol{S}_{Lk,nom}$</td></tr>
<tr><td>Constant I†</td><td>$\bar{\boldsymbol{I}}_{Lk} = (\boldsymbol{S}_{Lk,nom}./(\boldsymbol{U}\boldsymbol{V}_{k,nom}))^*$</td></tr>
<tr><td>Constant Z†</td><td>$\bar{\boldsymbol{y}}_{Lk} = \boldsymbol{S}^*_{Lk,nom}./\left|\boldsymbol{U}\boldsymbol{V}_{k,nom}\right|^2$</td></tr>
<tr><td rowspan="3">$\begin{bmatrix} V^{ab}_{k,nom} \\ V^{bc}_{k,nom} \\ V^{ca}_{k,nom} \end{bmatrix}$</td><td>Constant S</td><td>$\bar{\boldsymbol{S}}_{Lk} = -\boldsymbol{S}_{Lk,nom}$</td></tr>
<tr><td>Constant I</td><td>$\bar{\boldsymbol{I}}_{Lk} = (\boldsymbol{S}_{Lk,nom}./\boldsymbol{V}_{k,nom})^*$</td></tr>
<tr><td>Constant Z</td><td>$\bar{\boldsymbol{y}}_{Lk} = \boldsymbol{S}^*_{Lk,nom}./\left|\boldsymbol{V}_{k,nom}\right|^2$</td></tr>
</table>

†: Where $\boldsymbol{U} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ -1 & 0 & 1 \end{bmatrix}$

2.1.5 SHUNT CAPACITOR MODEL

Distribution system load and line characteristics are primarily inductive. Shunt capacitors are located throughout the distribution system to offset inductive reactance and supply reactive power to the network. Capacitor banks are frequently installed such that they are capable of providing multiple discrete levels of reactive power support. Capacitor control schemes are then used to improve the system voltage profile and reduce the total system real power losses. Local capacitor controllers are forced to make these decisions based on a single location measurement such as the bus voltage. Coordination of capacitor switching will then allow for on-line system wide loss reduction and voltage control.

In this thesis, capacitors will be modeled as purely reactive constant impedance loads. In grounded portions of the network, they will be connected in a grounded wye while in ungrounded portions of the network they will be connected in a delta configuration. The admittance matrix for these capacitors connections are represented as in Table 2.2.

Table 2.2: Capacitor/Load admittance matrices [7]

Load Connections	$\boldsymbol{V}_k$	Load Admittance Matrix
Grounded Wye	$\begin{bmatrix} V_k^a \\ V_k^b \\ V_k^c \end{bmatrix}$	$\begin{bmatrix} y_{Lk}^a & 0 & 0 \\ 0 & y_{Lk}^b & 0 \\ 0 & 0 & y_{Lk}^c \end{bmatrix}$
Ungrounded Delta		$\begin{bmatrix} y_{Lk}^{ca} + y_{Lk}^{ab} & -y_{Lk}^{ab} & -y_{Lk}^{ca} \\ -y_{Lk}^{ab} & y_{Lk}^{ab} + y_{Lk}^{bc} & -y_{Lk}^{bc} \\ -y_{Lk}^{ca} & -y_{Lk}^{bc} & y_{Lk}^{bc} + y_{Lk}^{ca} \end{bmatrix}$
	$\begin{bmatrix} V_k^{ab} \\ V_k^{bc} \end{bmatrix}$	$\begin{bmatrix} y_{Lk}^{ca} + y_{Lk}^{ab} & y_{Lk}^{ca} \\ -y_{Lk}^{ab} & y_{Lk}^{bc} \end{bmatrix}$

2.1.6 DISTRIBUTED GENERATOR MODEL

Distributed generation is installed in distribution systems to provide power generation support and to reduce peak system demands, reduce system loss, and improve reliability. In most distribution system, DGs with relatively small kilowatt (kW) output may be installed without voltage or output controllers. Larger DGs however are most commonly operated using these controls to protect the equipment from network events, control the bus voltage at which they are connected, and control the amount of real power output the generator produces. From a systems viewpoint, using generator domains and distributed slack bus models as presented in [10, 11], a DGs output can be attributed to serving either network load or losses. This information along with coordination of the local DG controllers will allow for optimal generation profiles to reduce system loss and economically dispatch DGs in the system.

In this thesis, DGs are modeled as constant negative constant power loads.

2.2 REVIEW OF NETWORK EQUIVALENCING TECHNIQUES

For the distributed analysis presented in this thesis, the network will be partitioned and then each partition's computation will be performed by a remote processor. A local processor will only have detailed information about a portion of the network for which it is responsible. The rest of the network is then modeled using equivalents.

Network equivalents have traditionally been used in power flow calculations for power transmission systems. They are applied to interconnected transmission systems in which an accurate picture of only a specific portion of the system is desired. The portion of the network of interest and its corresponding complement, the rest of the network, are referred to as the internal and the external networks, respectively. For a partitioned system in the distributed analysis presented in this thesis, these concepts are similar to a local partition and the rest of the network. The local partition is modeled in detail while the rest of the network is represented through equivalents. Because of the similarity of the two concepts, two common network equivalencing techniques will be presented.

The two methods of calculating external network equivalents reviewed in this sections are the Ward Injection method and the Radial Equivalent Independent of other nodes (REI) method. A review of the derivation of each equivalent model will be presented followed by a discussion of their applications to the distributed analysis objectives presented in this thesis.

2.2.1 WARD INJECTION METHOD

The Ward Injection equivalent method [12, 13] starts with a solved load flow of the external network including the boundary buses. Boundary buses are the set of buses at which the internal and external networks meet. A graphical depiction of the internal network, external network, and boundary buses can be seen in Figure 2.1.

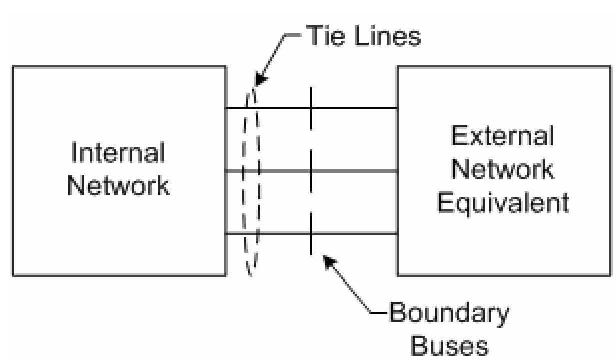

Figure 2.1: Representation of internal/external networks

The method then builds the nodal analysis equations of the external system and boundary buses at a given initial operating point, these equations are seen on the left in (2.3). Gaussian elimination is then performed to eliminate the external buses as seen on the right side of (2.3). This leaves an equivalent network which will be connected at the boundary buses and a set of equivalent boundary bus currents.

$$\begin{bmatrix} \boldsymbol{Y}_{ee} & \boldsymbol{Y}_{eb} \\ \boldsymbol{Y}_{be} & \boldsymbol{Y}_{bb} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_e \\ \boldsymbol{V}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_e \\ \boldsymbol{I}_b \end{bmatrix} \Rightarrow \begin{bmatrix} \boldsymbol{Y}_{ud,\,rem} & \boldsymbol{Y}_{rem} \\ 0 & \boldsymbol{Y'} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_e \\ \boldsymbol{V}_b \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_{rem} \\ \boldsymbol{I}' \end{bmatrix} \tag{2.3}$$

where:

$\boldsymbol{Y}_{ee}, \boldsymbol{Y}_{eb}, \boldsymbol{Y}_{be}, \boldsymbol{Y}_{bb}$: external network and boundary bus sub-matrices of the of the admittance matrix of the external and boundary buses

$\boldsymbol{I}_e, \boldsymbol{I}_b$: vector of external bus and boundary bus current injections, respectively

$\boldsymbol{V}_e, \boldsymbol{V}_b$: vector of external bus and boundary bus voltages, respectively

$\boldsymbol{Y'}, \boldsymbol{I'}$: equivalent network and equivalent boundary bus current injections, respectively

$\boldsymbol{Y}_{rem}, \boldsymbol{I}_{rem}$: remaining admittance matrix and current vector entries after Gaussian elimination

$\boldsymbol{Y}_{ud,\,rem}$: upper diagonal remaining admittance matrix entries after Gaussian elimination

Knowing the boundary bus voltages, the currents are converted to powers for use with most common industrial power flow packages. It can also be shown that when the boundary bus voltages are known, the value for $\boldsymbol{I}_e$ and $\boldsymbol{I}_b$ in (2.3) is irrelevant [14]. Whatever external and boundary bus injection values are used, after boundary matching, the equivalent boundary bus injections will be the same. The Ward Injection equivalent is therefore a function of only the topology of the external network.

2.2.2 REI EQUIVALENTS

The REI method [14, 15] starts with a solved power flow of the external system but does not include the set of boundary buses as in the Ward Injection method. The objective of the REI method is to eliminate the external network by aggregating injections of a group of buses of the external network onto a radial connected fictitious node added to the internal system. Buses of the external system are grouped according to a common parameter which could be electrical distance, geographical distance, or operator control area among others.

Given a group of buses, where each bus has voltage $|V_l|\angle\theta_l$ and current and power injections I_l and S_l, the REI method can be presented in 4 steps:

Step 1. Remove the injections from all buses

Step 2. Create fictitious REI bus *R*. The current and power injections into *R* are defined in (2.4) and (2.5), and the voltage is defined in (2.6)

$$I_R = \sum_{\forall\, l} I_l \tag{2.4}$$

$$S_R = \sum_{\forall\, l} S_l \tag{2.5}$$

$$V_R = \frac{S_R}{I_R^*} \tag{2.6}$$

Step 3. Augment the external network by connecting bus *R* to every bus through a fictitious REI network with an equidistant bus G. The REI network branch impedances are then:

$$Y_{RG} = \frac{S_R}{|V_R|^2} \tag{2.7}$$

$$Y_{RG} = -\frac{S_l}{|V_l|^2} \quad \forall\ l \tag{2.8}$$

Step 4. Eliminate all buses *l* and *G* by Gaussian elimination, similar to (2.3), leaving the equivalent network in which the REI bus has replaced the buses which have been eliminated

For both REI and Ward Injection equivalents, once the equivalent model is calculated, power flow studies on the internal network can then be performed. The derived equivalent model remains the same for each case studied.

2.2.3 APPLICATION TO DISTRIBUTED, DISTRIBUTION SYSTEM ANALYSIS

When the network is partitioned for distributed analysis, each partition will share one or more buses with other partitions in the network. Just as in the two methods above, these buses act as the boundary between the partition and the rest of the network which is not represented. When performing a distributed analysis, it is the characteristics of these boundary buses which are vital for attaining an accurate solution to the power flow.

While the traditional concepts of network equivalencing and external networks are similar to solving the problem of a distributed system, in reality a distribution system acts as only one entity. Despite the system being partitioned and distributed, an accurate picture of the state of the whole system is desired. This is unlike the external network equivalents which start with a solved power flow solution of the external network, create a static external network equivalent, and then focus only on the states of the internal system. Unlike both the Ward Injection and REI methods, in the distributed analysis, at each partition the operating state of the rest of the system will not be known. Equivalent models will instead be derived from power flow results communicated from the rest of the network and the solution to whole system will be determined.

Similar to the REI method of grouping buses of the external network together based on a shared physical characteristic or classification, the buses of the distribution system in the distributed analysis can be grouped into partitions based upon the location of the intelligent devices and controllers in the network. In addition, equivalent injections at the boundary buses which represent the characteristics of the external network as presented in the Ward Injection method will be required to represent external partitions. The calculation of these equivalent injections however will be derived iteratively from power flow results and calculated as presented in the next chapter.

3 COMPONENT AND SYSTEM MODELS FOR DISTRIBUTED ANALYSIS

For a distributed analysis of the system, the network is partitioned into subsystems, each representing a specific set of buses, branches, and loads of the original system. In each partition, the retained portion of the un-partitioned, original network will be modeled using the standard distribution system models discussed in the previous chapter. This chapter will first discuss network partitioning and then derive new network equivalents. Two types of equivalent models will be presented: equivalent sources and equivalent loads. Lastly, a model for the distributed system will be presented.

The focus of this thesis is on distribution system analysis. Distribution systems are operated primarily in a radial structure which is used for its simplicity of design and protection schemes. As such, the focus of the derivations of the following equivalent models will be presented with respect to systems of a radial topology. The equivalent models can however be applied to meshed systems. Following the presentation of the models, a discussion of their application to meshed systems will be presented.

3.1 NETWORK PATITIONS

The distribution system is divided into network partitions which are constructed based on the measurement capability, control capability, or control area of each distributed device in the network. Partitioning is done in this way to reflect the actual distribution of control, measurement, and protection devices throughout the network. The device characteristics of interest will depend upon the specific analysis to be performed. For example, the locations of sectionalizing and tie switches provide natural partition boundaries for distributed network reconfiguration or service restoration applications. Alternately, a distribution system with controllable DGs may be partitioned according to regions defining generator domains as presented in [10]. Thus, the network partitioning procedure is flexible and can be tailored to the specific type of system analysis to be performed.

In this thesis, the network is partitioned at specific buses which then form the boundaries of each partition. The system will be partitioned into n_p partitions with each partition denoted as partition *i*, where $i = 0, 1, 2, \ldots, n_p$. Once the network is partitioned it may then be viewed as a graph. In this context, each partition can be seen to have a physical relationship to each other. Partitions which share a bus in the un-partitioned, original system, subsequently referred to as the original system, will be defined to be adjacent to one another in the partitioned network. In addition, for radial systems, partitions may also be referred to as upstream or downstream from one another with respect to a specified source in the network, not necessarily the substation.

With the focus on systems with radial topology, when partitioning the original system, each partition then represents one or more systems each with the following characteristics:

- One bus will serve as the equivalent source bus of its respective partition, in these discussions selected as the bus closest to the substation. This bus will model the upstream partitions as an equivalent voltage.
- End buses of each partition will serve as equivalent load buses. These buses will model downstream partitions as an equivalent load, in addition to any load present on the corresponding bus of the original system

A bus of the original system may appear as an equivalent load bus in one partition and as an equivalent source bus in another. In addition, exceptions exist for both of the above cases. A source bus equivalent is not needed for the partition that includes the substation because the substation bus voltage is known. An equivalent load is not needed for partitions which have no further downstream adjacent partitions from themselves because these partitions contain the actual end buses of the main feeder and laterals.

Figure 3.1 is used to illustrate the partitioning concepts. Figure 3.1 presents a 97 bus, 249 node distribution system distributed into 5 partitions. In this example, partition 1 and 2 can be seen to be adjacent to partition 0. Partition 1 has an equivalent source located at Bus 4" and an equivalent load bus at Bus 69'. The models and methods for determining model parameters for the source and load equivalents are discussed in the following two sections.

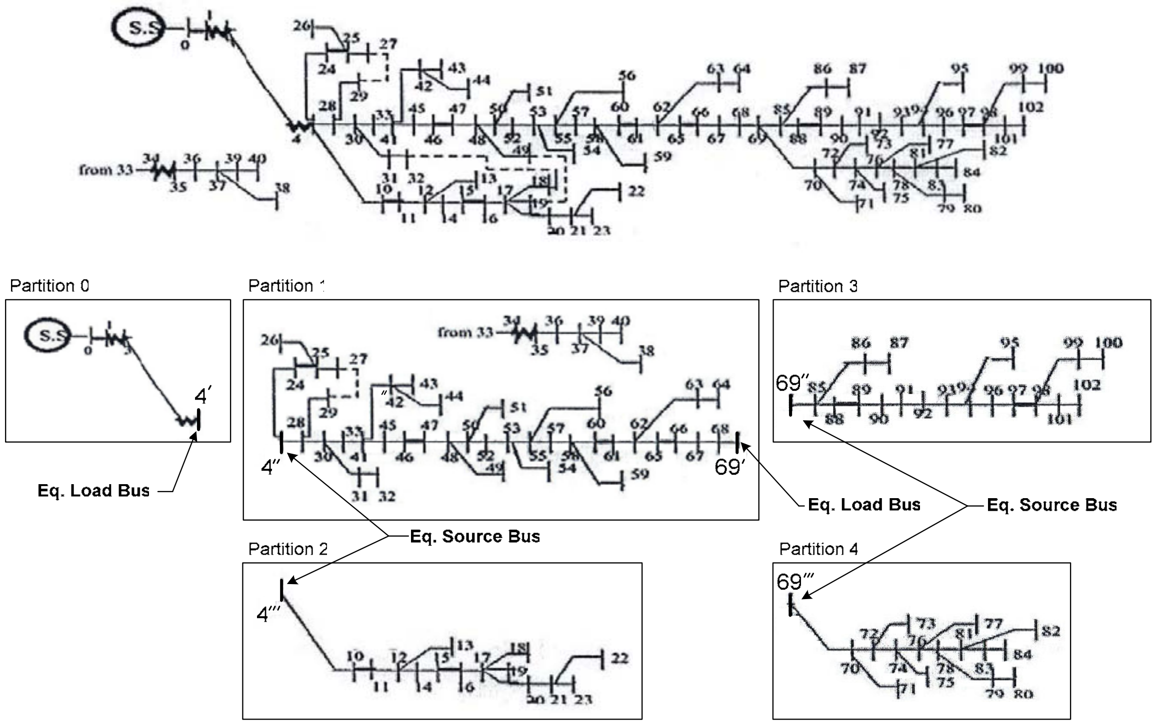

Figure 3.1: Diagram of a 97 bus, 249 node partitioned distribution system

3.2 EQUIVALENT SOURCE MODEL

An equivalent source model is required for each network partition except that which contains the substation. The equivalent source bus of a partition models the effects of the upstream network as an ideal multi-phase unbalanced voltage source. The value of each equivalent source bus voltage is set by the voltage of the equivalent load bus of the adjacent upstream partition. The adjacent upstream equivalent load bus voltage is determined through a power flow which requires information from downstream partitions. This interdependence leads to an iterative process between power flows of adjacent partitions and causes the solution algorithm, described in detail in the next chapter, to take on a backward/forward sweep nature of partition processing. For illustration, the equivalent source bus of partition 1 and partition 2 in Figure 3.1 are highlighted in Figure 3.2 below.

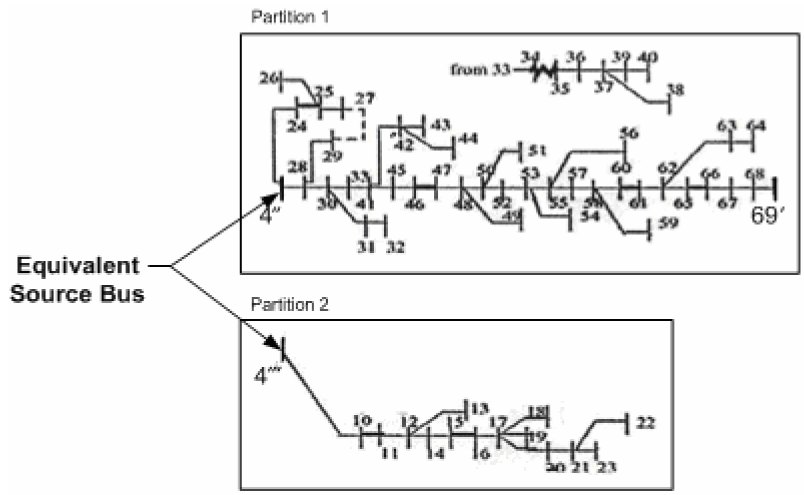

Figure 3.2: Example of equivalent source buses in the 97 bus test system

The parameters of an equivalent source bus are updated using information communicated from its adjacent upstream partition. Specifically, the equivalent source bus model of partition i is:

$$\boldsymbol{V}_{i,0}^{(k)} = \boldsymbol{V}_{i-1,\ eq.\ load}^{(k)} . \tag{3.1}$$

where:

$\boldsymbol{V}_{i,0}^{(k)}$: complex (n_{ph} x 1), multi-phase equivalent source bus voltage vector update of partition i, iteration k

$\boldsymbol{V}_{i-1,\ eq.\ load}^{(k)}$: complex (n_{ph} x 1), multi-phase equivalent load bus voltage vector of the corresponding upstream partition i-1, iteration k

k : iteration number

3.3 EQUIVALENT LOAD MODELS

Upstream partitions require an equivalent load to model the effect of the downstream network. The parameters for an equivalent load bus are updated using information communicated from its adjacent downstream partition. An equivalent load is calculated for each partition by performing a traditional power flow on the partition using the most recently updated values of the equivalent source and load parameters. The solution is then post-processed to calculate an equivalent load which will be sent to adjacent upstream partitions. The calculation will be dependent on whether the equivalent source bus of the partition is located in a grounded or un-grounded portion of the network and on the load types found in the partition to be represented by the equivalent. In this thesis, calculation of the load equivalent utilizes the power flow solution of the relevant partition. The conditions under which the power flow on each partition will converge are discussed in Chapter 5.

Depending on the load types found in a partition and the presence of a ground on the upstream equivalent load bus, the equivalent load passed to the adjacent upstream partition will be an equivalent current injection, equivalent impedance, or equivalent power injection. The decision of the equivalent load type is based on the distribution of load types in the network contained in a partition. The equivalent load value is passed through the communication system and placed on the adjacent upstream partition equivalent load bus as a constant impedance, constant current, or constant power load, respectively. For illustration, the equivalent load bus of partition 0 in Figure 3.1 is highlighted in Figure 3.3 below. The calculation of equivalent loads for grounded and un-grounded equivalent load buses is presented in the next sections followed be procedures for aggregating multiple partition equivalents together.

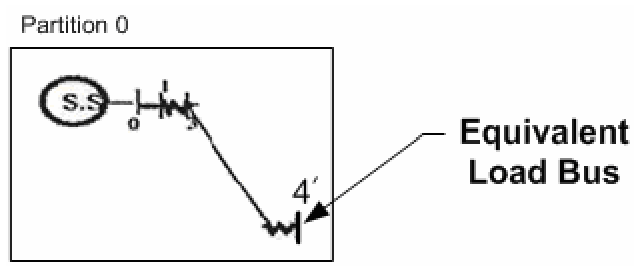

Figure 3.3: Example of an equivalent load bus in the 97 bus test system

3.3.1 EQUIVALENT LOAD MODEL FOR PARTITIONS WITH A GROUNDED EQUIVALENT SOURCE BUS

The following section describes the process for determining an equivalent load to represent a partition if that partition's equivalent source bus is located in a grounded portion of the network. In this case, the equivalent load is passed to the adjacent upstream partition where it is represented as a grounded wye connected load on the adjacent upstream partition's equivalent load bus. This equivalent load bus will also be in a grounded portion of the network because the adjacent equivalent source and equivalent load buses are modeling the same bus of the original system. The equivalent load representing partition *i* will be denoted with the subscript: *eq., i*. The determination of the equivalent load model to be passed is dependent on the load types located throughout the network contained in the partition.

If a partition has all constant current loads, the equivalent current injection of the partition is modeled as a constant current load on the equivalent load bus of the adjacent upstream partition. The equivalent current injection of a partition is determined by the total current output from the equivalent source bus of that partition:

$$\boldsymbol{I}_{eq,\,i}^{(k)}\left(\boldsymbol{V}^{(k)}\right)=\sum_{\forall\; br\in B_i}\boldsymbol{I}_{i,\,br}^{(k)}\left(\boldsymbol{V}^{(k)}\right). \qquad (3.2)$$

where:

$\boldsymbol{I}_{eq,\,i}^{(k)}\left(\boldsymbol{V}^{(k)}\right)$	:	complex (n_{ph} x 1) multi-phase equivalent current injection vector, partition i, iteration k
$\boldsymbol{I}_{i,\,br}^{(k)}$	:	complex (n_{ph} x 1) multi-phase current on branch br, partition i, iteration k
B_i	:	set of all branches connected directly to the equivalent source bus, partition i
$\boldsymbol{V}^{(k)}$	:	complex (n_n x 1) partitioned system voltage vector, iteration k
n_n	:	number of nodes in the distributed system

The single line diagram of a radial distribution system shown in Figure 3.4 is used to illustrate the calculation of the equivalent current injection of a partition. The power flow is computed for the partition and is then post-processed to calculate the partition branch currents.

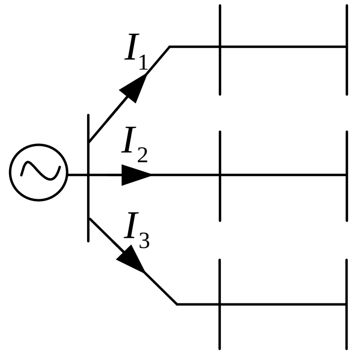

Figure 3.4: Equivalent current injection example

The equivalent current injection of the partition is then calculated by summing the currents on the branches connected directly to the equivalent source bus of the partition:

$$I_{eq} = I_1 + I_2 + I_3 \tag{3.3}$$

If a partition has all constant impedance loads, the equivalent admittance of the partition is modeled as a constant impedance load on the equivalent load bus of the adjacent upstream partition. The equivalent admittance of a partition is determined using its equivalent current injection and the value of its equivalent source bus voltage at the current iteration:

$$\boldsymbol{Y}_{eq,\,i}^{(k)}\left(\boldsymbol{V}^{(k)}\right) = 1./\left(\boldsymbol{V}_{i,0}^{(k)} \boldsymbol{./I}_{eq,\,i}^{(k)}\right). \tag{3.4}$$

where:

$\boldsymbol{Y}_{eq,\,i}^{(k)}\left(\boldsymbol{V}^{(k)}\right)$: complex (n_{ph} x *1*) multi-phase equivalent admittance vector, partition i, iteration k

$\boldsymbol{V}_{i,0}^{(k)}$: complex (n_{ph} x *1*) multi-phase equivalent source bus voltage

vector, partition *i*, iteration *k*

partition i, iteration *k*

./ : element-wise division

If a partition has all constant power loads, the equivalent power injection of the partition is modeled as a constant power load on the equivalent load bus of the adjacent upstream partition. The equivalent power injection of a partition is determined using its equivalent current injection and the value of its equivalent source bus voltage at the current iteration:

$$\boldsymbol{S}_{eq,\,i}^{(k)}\left(\boldsymbol{V}^{(k)}\right) = \boldsymbol{V}_{i,0}^{(k)} .* \left(\boldsymbol{I}_{eq,\,i}^{(k)}\right)^{*}. \tag{3.5}$$

where:

$\boldsymbol{S}_{eq,\,i}^{(k)}\left(\boldsymbol{V}^{(k)}\right)$: complex (n_{ph} x *1*) multi-phase equivalent power injection vector,

partition i, iteration *k*

.* : element-wise multiplication

superscript * : complex conjugate

Lastly, if a partition contains a mixture of constant impedance, constant current, and constant power loads, the partition is modeled as an equivalent current injection as calculated in (3.2).

3.3.2 EQUIVALENT LOAD MODELS FOR PARTITIONS WITH AN UN-GROUNDED EQUIVALENT SOURCE BUS

If a partition's equivalent source bus is located in an un-grounded portion of the network, the equivalent load is passed to the adjacent upstream partition and modeled as an un-grounded delta connected load on that partition's equivalent load bus. Again, for the same reasons as in the grounded case, this equivalent load bus will also be located in an un-grounded portion of the network.

If a partition has all constant current loads, the equivalent current injection of the partition is modeled as a constant current injection into the equivalent load bus of the adjacent upstream partition. The state variables for ungrounded portions of the network are the line-to-line voltages V^{ab} and V^{bc}. The required current injection to represent the equivalent ungrounded load will be two dimensional as well, selecting I^a and I^b with I^c being redundant and equal to $-(I^a + I^b)$. The equivalent current injection of a partition is determined by the total current output from the equivalent source bus of the partition as in (3.2).

If a partition has all constant impedance loads, all constant power loads, or a mixture of load types, the equivalent load will be represented as an equivalent current injection into the equivalent load bus of the adjacent upstream partition. The equivalent current injection of a partition is again determined by the total current output from the equivalent source bus of that partition as in (3.2).

3.3.3 REPRESENTING EQUIVALENT LOADS ON THE EQUIVALENT LOAD BUS

If a partition has multiple adjacent downstream partitions, the injections of the equivalent loads of each adjacent downstream partition must be aggregated. In addition, any injections from distributed generators, capacitors, or existing load present on the corresponding bus of the original system must also be included. The general expression for the total current injection into bus *j* of the distribution system can be expressed as:

$$\boldsymbol{I}_j\left(\boldsymbol{V}^{(k)}\right)=\boldsymbol{I}_j^{DG}\left(\boldsymbol{V}^{(k)}\right)+\boldsymbol{I}_j^{C}\left(\boldsymbol{V}^{(k)}\right)+\boldsymbol{I}_j^{L}\left(\boldsymbol{V}^{(k)}\right) \tag{3.6}$$

where:

$\boldsymbol{I}_j^{DG,C,L}\left(\boldsymbol{V}^{(k)}\right)$: complex (n_{ph} x 1), multi-phase currents injected by distributed generators, capacitors, and loads, respectively, on bus *j*

In the distributed analysis, at each equivalent load bus, the equivalents loads of each adjacent downstream partition will be included in this injection as well. The general expression for the total current injection into the equivalent load bus *j* of partition *i* is then:

$$\boldsymbol{I}_{i,j}\left(\boldsymbol{V}^{(k)}\right)=\boldsymbol{I}_{i,j}^{DG}\left(\boldsymbol{V}^{(k)}\right)+\boldsymbol{I}_{i,j}^{C}\left(\boldsymbol{V}^{(k)}\right)+\boldsymbol{I}_{i,j}^{L}\left(\boldsymbol{V}^{(k)}\right)+\sum_{\forall\, d\in D_i}\boldsymbol{I}_{eq,\,d}^{(k)}\left(\boldsymbol{V}^{(k)}\right) \tag{3.7}$$

where:

$\boldsymbol{I}_{i,j}^{DG,C,L}\left(\boldsymbol{V}^{(k)}\right)$: complex (n_{ph} x 1), multi-phase currents injected by distributed generators, capacitors, and loads, respectively, on bus *j*, partition *i*

$\boldsymbol{I}_{eq,\,i}^{(k)}\left(\boldsymbol{V}^{(k)}\right)$: complex ($n_{ph}$ x 1) multi-phase equivalent current injection, partition *i*

D_i : set of all adjacent partitions downstream from partition *i*

The aggregation of equivalent current injections from adjacent downstream partitions requires that each of these partitions has communicated the required information for the current iteration. Therefore, a processor must wait for all current information to be communicated from downstream before it can perform a power flow on its local partition.

3.4 DISTRIBUTED SYSTEM MODEL

Using the equivalent source and equivalent load component models, a system model for the partitioned network can be established. Formulating the nodal analysis equations for the partitioned network results in a set of equations which can be expressed as:

- a block diagonal admittance matrix,
- a vector of the partitioned system voltages, and
- a vector of current injections, including the equivalent partition injections.

Three example systems will be shown using each of the different equivalent loads to represent downstream partitions. The example distributed system models will first derived using equivalent impedance loads, then equivalent power injection loads, and lastly equivalent current injections loads.

These examples are derived from the single-phase 5 bus system seen in Figure 3.5. First, the nodal analysis equations of the original system are expressed in (3.8).

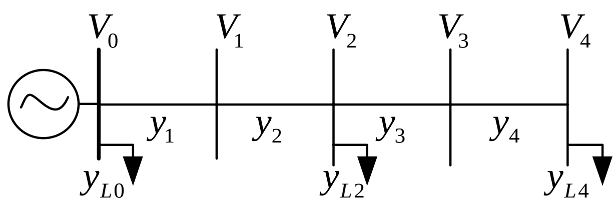

Figure 3.5: Single-phase 5 bus system

$$\begin{bmatrix} y_1+y_{L0} & -y_1 & 0 & 0 & 0 \\ -y_1 & y_1+y_2 & -y_2 & 0 & 0 \\ 0 & -y_2 & y_2+y_3+y_{L2} & -y_3 & 0 \\ 0 & 0 & -y_3 & y_3+y_4 & -y_4 \\ 0 & 0 & 0 & -y_4 & y_4+y_{L4} \end{bmatrix} \begin{bmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} I_0(\boldsymbol{V_o}) \\ I_1(\boldsymbol{V_o}) \\ I_2(\boldsymbol{V_o}) \\ I_3(\boldsymbol{V_o}) \\ I_4(\boldsymbol{V_o}) \end{bmatrix} \tag{3.8}$$

where:

V_j	:	complex voltage at bus *j*
y_j	:	complex admittance of branch *j*
$I_j(\boldsymbol{V})$	:	complex current injection, bus *j*
$\boldsymbol{V_o}$	:	complex (5 x 1) voltage vector of the un-partitioned system

The system is then partitioned into three subsystems according to the previously discussed models as seen in Figure 3.6. For this example, equivalent admittance models are used for the equivalent loads.

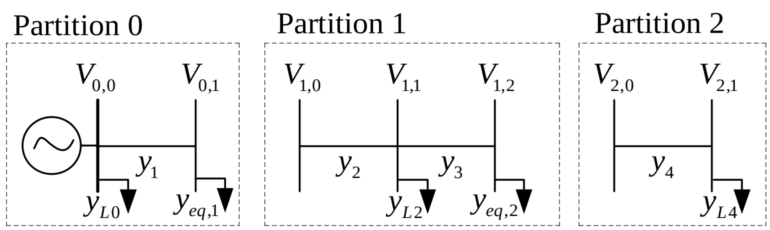

Figure 3.6: Partitioned single-phase 5 bus system – equivalent admittance

The resulting nodal analysis equations, seen in (3.9), will now include the equivalent admittances in the system admittance matrix.

$$\left[\begin{array}{cc|ccc|cc} y_1+y_{L0} & -y_1 & 0 & 0 & 0 & 0 & 0 \\ -y_1 & y_1+y_{eq,1} & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & y_2 & -y_2 & 0 & 0 & 0 \\ 0 & 0 & -y_2 & y_2+y_3+y_{L2} & -y_3 & 0 & 0 \\ 0 & 0 & 0 & -y_3 & y_3+y_{eq,2} & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & y_4 & -y_4 \\ 0 & 0 & 0 & 0 & 0 & -y_4 & y_4+y_{L4} \end{array}\right] \left[\begin{array}{c} V_{0,0} \\ V_{0,1} \\ \hline V_{1,0} \\ V_{1,1} \\ V_{1,2} \\ \hline V_{2,0} \\ V_{2,1} \end{array}\right] = \left[\begin{array}{c} I_{0,0}(\boldsymbol{V}) \\ I_{0,1}(\boldsymbol{V}) \\ \hline I_{1,0}(\boldsymbol{V}) \\ I_{1,1}(\boldsymbol{V}) \\ I_{1,2}(\boldsymbol{V}) \\ \hline I_{2,0}(\boldsymbol{V}) \\ I_{2,1}(\boldsymbol{V}) \end{array}\right] \tag{3.9}$$

Next, the system is partitioned into three subsystems as seen in Figure 3.7 using equivalent power injection models for the equivalent loads. The resulting nodal analysis equations, seen in (3.10), will now include the current injections from the linearized equivalent power injections in the current injection vector.

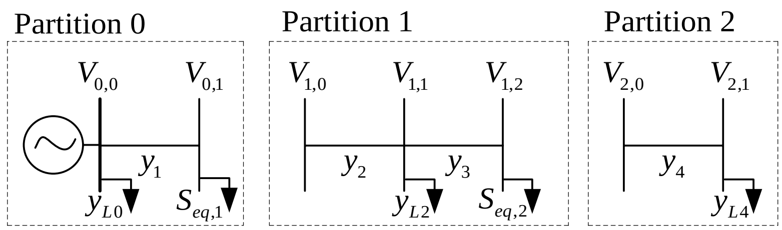

Figure 3.7: Partitioned single-phase 5 bus system – equivalent power injection

$$\left[\begin{array}{cc|ccc|cc} y_1+y_{L0} & -y_1 & 0 & 0 & 0 & 0 & 0 \\ -y_1 & y_1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & y_2 & -y_2 & 0 & 0 & 0 \\ 0 & 0 & -y_2 & y_2+y_3+y_{L2} & -y_3 & 0 & 0 \\ 0 & 0 & 0 & -y_3 & y_3 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & y_4 & -y_4 \\ 0 & 0 & 0 & 0 & 0 & -y_4 & y_4+y_{L4} \end{array}\right]\left[\begin{array}{c} V_{0,0} \\ V_{0,1} \\ \hline V_{1,0} \\ V_{1,1} \\ V_{1,2} \\ \hline V_{2,0} \\ V_{2,1} \end{array}\right] = \left[\begin{array}{c} I_{0,0}(\boldsymbol{V}) \\ (S_{eq,1}./V_{0,1})^* \\ \hline I_{1,0}(\boldsymbol{V}) \\ I_{1,1}(\boldsymbol{V}) \\ (S_{eq,2}./V_{1,2})^* \\ \hline I_{2,0}(\boldsymbol{V}) \\ I_{2,1}(\boldsymbol{V}) \end{array}\right] \quad (3.10)$$

Lastly, Figure 3.8 shows the system partitioned into three subsystems using equivalent current injection models for the equivalent loads to represent downstream partitions. The nodal analysis equations used to describe this partitioned system are seen in (3.11).

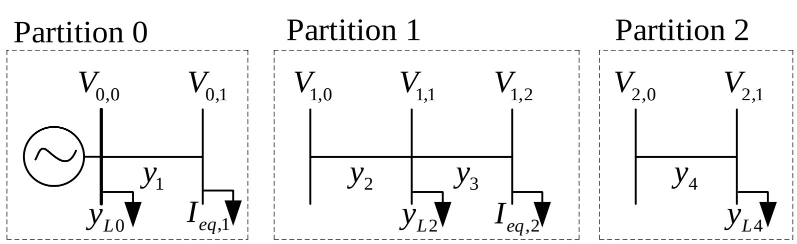

Figure 3.8: Partitioned single-phase 5 bus system – equivalent current injection

$$\left[\begin{array}{cc|ccc|cc} y_1+y_{L0} & -y_1 & 0 & 0 & 0 & 0 & 0 \\ -y_1 & y_1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & y_2 & -y_2 & 0 & 0 & 0 \\ 0 & 0 & -y_2 & y_2+y_3+y_{L2} & -y_3 & 0 & 0 \\ 0 & 0 & 0 & -y_3 & y_3 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & y_4 & -y_4 \\ 0 & 0 & 0 & 0 & 0 & -y_4 & y_4+y_{L4} \end{array}\right]\left[\begin{array}{c} V_{0,0} \\ V_{0,1} \\ \hline V_{1,0} \\ V_{1,1} \\ V_{1,2} \\ \hline V_{2,0} \\ V_{2,1} \end{array}\right]=\left[\begin{array}{c} I_{0,0}(\boldsymbol{V}) \\ I_{0,1}(\boldsymbol{V})+I_{eq,1}(\boldsymbol{V}) \\ \hline I_{1,0}(\boldsymbol{V}) \\ I_{1,1}(\boldsymbol{V}) \\ I_{1,2}(\boldsymbol{V})+I_{eq,2}(\boldsymbol{V}) \\ \hline I_{2,0}(\boldsymbol{V}) \\ I_{2,1}(\boldsymbol{V}) \end{array}\right] \tag{3.11}$$

where:

$V_{i,j}$: complex voltage at bus *j*, partition *i*

$I_{i,j}(\boldsymbol{V})$: complex current injection at bus *j*, partition *i*

$I_{eq,i}(\boldsymbol{V})$: complex equivalent current injection at bus *j*, partition *i*

Equation (3.11) is the system model for the partitioned network. In this example, the block diagonal form of the admittance matrix along with the system voltage and current vectors can be seen. It is also of note that each have increased in dimension by 2. The increase in system dimension is due to the equivalent source and load buses in the partitioned network.

Figure 3.9 shows the block system model for a single phase representation of a general multi-phase radial network of n_b buses analyzed with n_p partitions using equivalent current injections. The corresponding nodal analysis equations can be seen in (3.12).

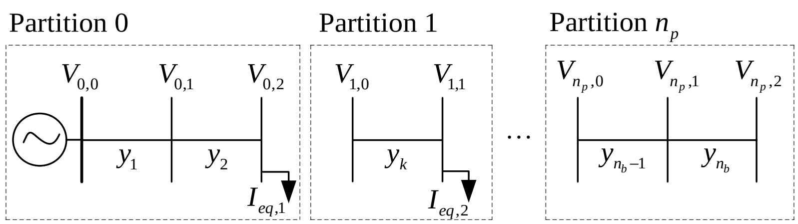

Figure 3.9: General n_b bus, n_n node, n_p partition radial system

$$\begin{bmatrix} \boldsymbol{Y}_{00} & 0 & 0 & 0 \\ 0 & \boldsymbol{Y}_{11} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \boldsymbol{Y}_{n_p n_p} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_0 \\ \boldsymbol{V}_1 \\ \vdots \\ \boldsymbol{V}_{n_p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_0(\boldsymbol{V}) + \boldsymbol{I}_{eq,1}(\boldsymbol{V}) \\ \boldsymbol{I}_1(\boldsymbol{V}) + \boldsymbol{I}_{eq,2}(\boldsymbol{V}) \\ \vdots \\ \boldsymbol{I}_{n_p}(\boldsymbol{V}) \end{bmatrix} \tag{3.12}$$

where:

$\boldsymbol{Y}_{ii}$: complex (n_n^i x n_n^i) admittance matrix, partition i

n_n^i : number of nodes, partition i

$\boldsymbol{V}$: complex (n_n x 1) voltage vector of the partitioned system

3.5 EXTENSIONS FOR MESHED SYSTEMS

The equivalent models derived in this chapter can be applied to meshed system which can be partitioned such that:

- any loops in the system are contained within each of the partitions and
- the system of partitions maintains a radial structure

An example of such a system is shown in Figure 3.10 below.

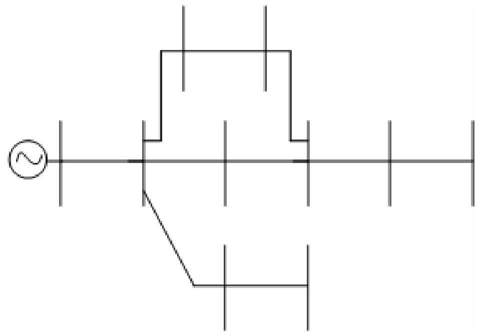

Figure 3.10: Meshed system which can be partitioned in a radial structure

This system may be partition such that the loop is included within one partition and a radial structure of the system of partitions is maintained. An example of the system in Figure 3.10 partitioned in this way can be seen in Figure 3.11 below.

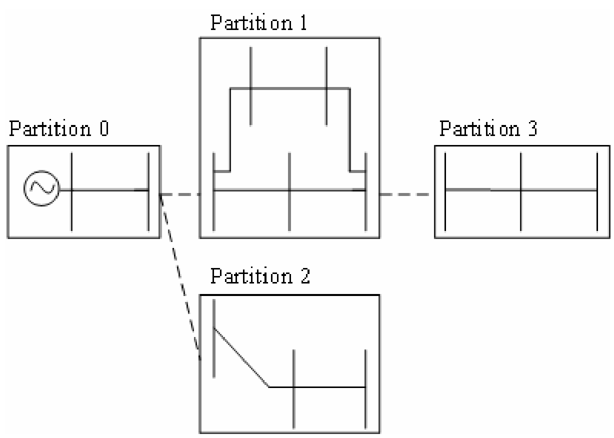

Figure 3.11: Meshed system with radial structure of partitions

The generalizations made for the application of equivalent models derived in this chapter must be re-evaluated when performing a distributed analysis on a meshed system and the resulting system of partitions is meshed in addition to the networks contained in each partition. An example of this type of system is shown in Figure 3.12.

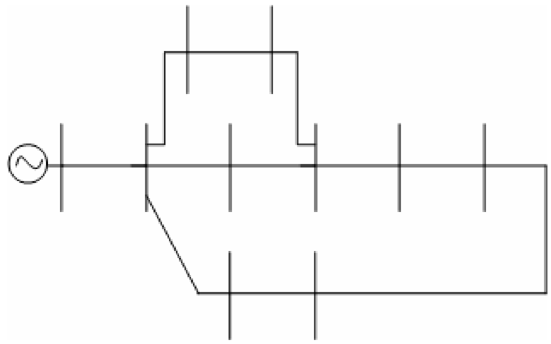

Figure 3.12: Meshed system which cannot be partition into a radial structure

Partitioning this system will result in a meshed structure of partitions. An example of such a partitioning can be seen in Figure 3.13.

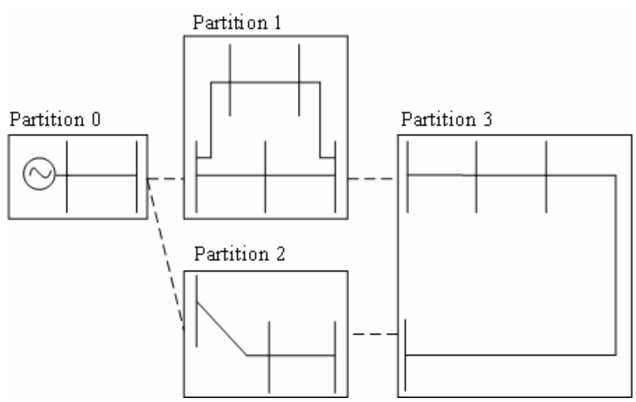

Figure 3.13: Meshed system with meshed structure of partitions

For a meshed system of partitions, such as the example above, the concepts of upstream and downstream are less clearly defined. Therefore generalizations made for radial systems in the selection of the equivalent type of each boundary bus cannot be applied. In a meshed system of partitions, each of boundary buses at which the network is partitioned could serve as either an equivalent source bus or equivalent load bus. One method for determining which type of equivalent to use is by looking at the direction of the estimated net power flow through each boundary bus. In [10], a method is described which allows the path of active power flow through each branch and bus from each generator to be traced. Applying this concept to the distributed analysis of a meshed system will allow the direction of power flow into and out of any partition through the boundary bus to be identified. This information can then be used to dictate whether the boundary bus should be modeled as an equivalent source or equivalent load.

With a method for designating the equivalent type, source or load, of each boundary bus, the application of the equivalent models discussed in this chapter will need to be revisited. In the meshed structure, there may be, for example, for more than one bus to be designated as an equivalent source bus. The single equivalent source bus in previously assumed radial partition structure was used as the reference bus for power flow on a partition, with the voltage magnitude and angle at each phase held constant. To handle the case of multiple source buses, one possibility is to allow one equivalent source bus in each partition to function as the reference bus while allowing the remaining equivalent sources to function as constant power injection and voltage magnitude buses, or P|V| buses, for power flow on the partition.

An example of this type of system is shown below in Figure 3.14.

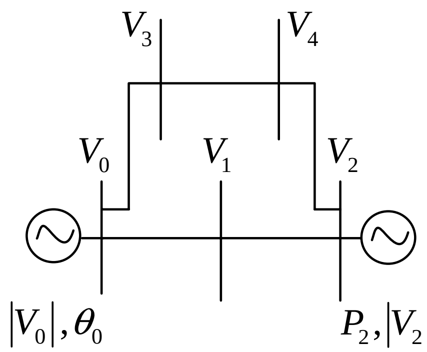

Figure 3.14: Meshed partition with multiple equivalent sources

The equivalent source at bus 0 will receive both its voltage magnitude and voltage angle from an adjacent partition while the equivalent source at bus 2 will function as a P|V| bus with voltage magnitude and power injection set from its adjacent partition. The value of these equivalents can then be iteratively updated during the solution of the distributed power flow. Additional considerations and future work for meshed systems will be discussed in Chapter 7.

4 DISTRIBUTED MULTI-PHASE POWER FLOW ALGORITHIM

With the required models in place, a solution algorithm for calculating the system states is required. The distributed power flow method in this thesis calculates the state of the whole system by iteratively running a traditional power on each partition using distributed processors which communicate results. This chapter will present the solution algorithm for solving the distributed power flow as well as an implementation of the proposed algorithm using commercial software packages.

4.1 SOLUTION ALGORITHM

The following section details an optimal partition calculation ordering scheme, step-by-step algorithm, and convergence criterion for the distributed power flow. To calculate the power flow within a partition, any traditional power flow algorithm may be used. These include the implicit Z-bus Gauss [16, 17], backward forward sweep [7], fast decoupled [18], or Newton Rhapson algorithms [19]. The choice of power flow algorithm at each partition is left to the discretion of the local processor.

4.1.1 OPTIMAL PARTITION CALCULATION ORDER

The power flow calculations at each partition should be performed in a prescribed order because of the radial topology of the partitions. While this ordering is suggested as optimal in terms of reducing the number of iterations, it is not required, and as such equivalent load or source updates could proceed in any arbitrary order.

For the general case of n_p partitions, to update the equivalent load values, the partitions should be processed in reverse breadth first search order [18]. This corresponds to the backward substitution of the equivalent load values, $\boldsymbol{I}_{eq,i}$, in the system model presented in (3.12) and repeated in (4.1) below. This backward substitution is the process of solving the power flow at a partition, calculating the equivalent load, and then using that equivalent load in the power flow of an upstream partition. The suggested partition ordering allows for the most up-to-date equivalent load values to be used for each power flow at a given iteration be starting with the partitions representing the end of lines and laterals with no equivalent load and working back toward the substation.

$$\begin{bmatrix} \boldsymbol{Y}_{00} & 0 & 0 & 0 \\ 0 & \boldsymbol{Y}_{11} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \boldsymbol{Y}_{n_p n_p} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_0 \\ \boldsymbol{V}_1 \\ \vdots \\ \boldsymbol{V}_{n_p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_0(\boldsymbol{V}) + \boldsymbol{I}_{eq,1}(\boldsymbol{V}) \\ \boldsymbol{I}_1(\boldsymbol{V}) + \boldsymbol{I}_{eq,2}(\boldsymbol{V}) \\ \vdots \\ \boldsymbol{I}_{n_p}(\boldsymbol{V}) \end{bmatrix} \tag{4.1}$$

To subsequently update the equivalent source bus voltages, the partitions should be processed in breadth first search order corresponding to forward substitution in (4.1) of the equivalent source bus voltages. This allows for the most up-to-date equivalent source bus voltages to be used for a given iteration by starting with the partition representing the substation with known voltage and working towards the partitions containing the ends of lines and laterals.

To proceed in the suggested order, a partition indexing scheme is proposed which dictates the order in which to process each partition. This in turn sets the order of the blocks of the system model in (4.1). Similar to the way a distribution system may be viewed as a main feeder containing laterals which are located on different levels away from the main feeder [6], the partitioned network may be viewed as a main set of partitions containing lateral partitions located on different levels away from the main set. In this way, each partition can be identified by an ordered triple (i, j, k) representing the level, lateral, and partition indices, respectively, of each partition. The partitions must be processed first in decreasing order of levels, then decreasing order of laterals, then decreasing order of partition indices. Because these calculations are being performed on a distributed set of processors, for a given level, the power flows may be calculated in parallel.

Figure 4.1 represents an example of this ordering applied to the partitioned 97 bus system presented in Figure 3.1. Using the partition index 0 to represent the partition containing the substation, and the index 1 to represent the main set, all partitions may be assigned the corresponding ordered triple. This ordered triple dictates the order, seen as the circled numbers in Figure 4.1, in which the partitions must be processed in descending number on the backward sweep and ascending number on the forward sweep of the algorithm.

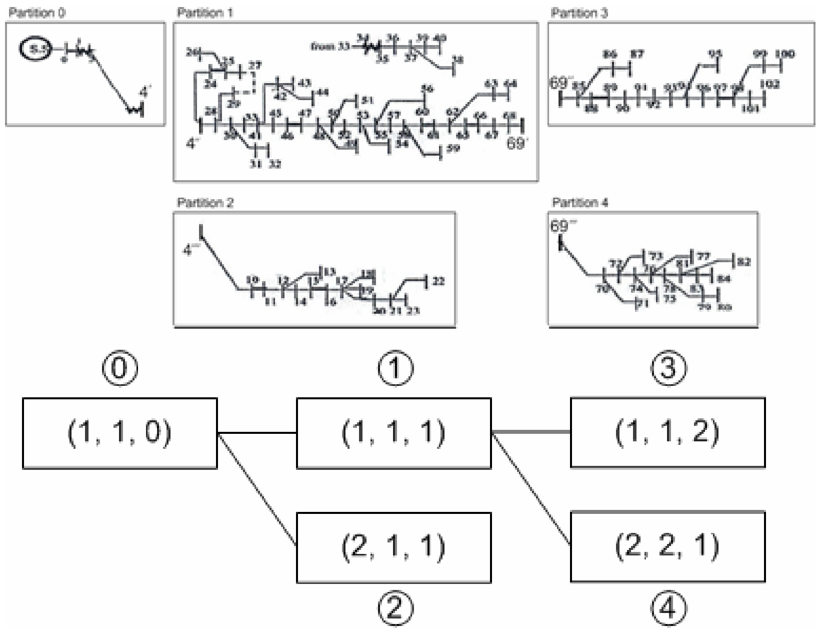

Figure 4.1: Partition indexing scheme applied to the 97 bus test system

4.1.2 SOLUTION ALGORITHM

Because the solution process is an iterative one, the equivalent source bus voltages are initialized for the first iteration. The initial voltage conditions for each equivalent source bus can be specified using a flat, balanced voltage profile or historical power flow data if available. Using the above ordering, the solution algorithm of the distributed power flow can then be expressed in the following 9 steps:

Step 1. Set $k = 0$; initialize all equivalent source bus voltages

Step 2. Perform a power flow for each partition which has no further downstream partitions

Step 3. Post-process the solution of each power flow to calculate an equivalent load

Step 4. Communicate equivalent loads through the communication network to the adjacent upstream partitions/remote processors

Step 5. For each upstream partition, aggregate equivalent loads, if necessary, and perform a power flow

Step 6. If no further upstream partitions exist, let $k = k + 1$, and go to Step 7; else, go to Step 3.

Step 7. Communicate the end bus voltages to adjacent downstream partitions/remote processors

Step 8. If no further downstream partitions exist, go to Step 9; else, perform power flow on these partitions and go to Step 7

Step 9. Check for system convergence. If convergence has occurred, Stop. Else, go to Step 2

Figure 4.2 illustrates a flow chart of the above algorithm using. On the backward sweep, when the power flows are being run on the downstream partitions, the upstream partitions must wait to receive the post-processed data before they can perform their own power flow. Similarly, this lag occurs for the downstream partitions on the forward sweep. The timing considerations of the algorithm will be discussed next, followed by the convergence criterion.

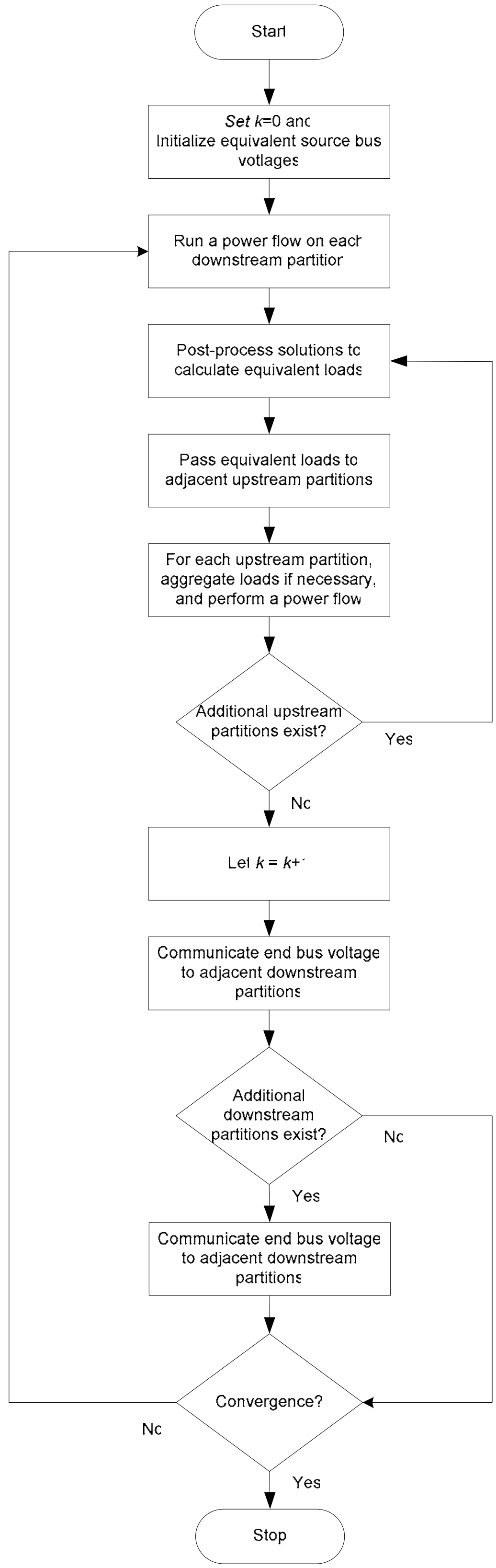

Figure 4.2: Flow chart of distributed power flow algorithm

4.1.3 TIMING CONSIDERATIONS

The distributed power flow algorithm described above calculates the steady state solution to the system of partitions under the assumption that all communication channels are available during the solution process and the communication does not add error to the data transmitted. As such, partition wait times and delays due to the communication network are not detrimental to the stability or accuracy of the algorithm as might be encountered in a distributed dynamic simulation. These lags do however effect the total time required to attain a solution to the system. In this section, a discussion and model of the timing considerations with regard to communication delays and computation times of the distributed power flow is presented. A more detailed treatment of distributed computation timing considerations may be seen in [20].

Network time delays may be may be modeled as an exponential function of the network medium, traffic, and distances as well as other parameters. A stochastic system model for information embedded power systems can be in found in [21] which elaborates on time delays in the communication channels and their impact on power system measurements. Assuming a network and computational time delay model, a model for the total calculation time of the distributed power flow can be formulated.

The total time to run a power flow and perform the required post-processing on partition *i*, will be denoted as τ_i. The time delay in the communication channel between two adjacent partitions *i* and *j*, will be denoted $\tau_{i\text{-}j}$. An illustration of these corresponding times for the example 5 partition system can be seen in Figure 4.3.

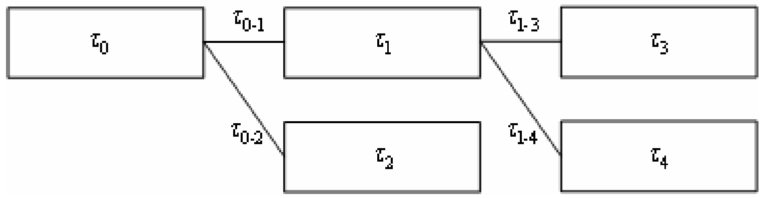

Figure 4.3: Computation and communication times for 5 partition system

The total backwards computation and communication time, t_b, on the backward sweep process of the algorithm is

$$\tau_b = \sum_{i=1}^{n_p} \tau_i + \left(\sum \tau_{i\text{-}j} \ \ \forall\ i,\ j \text{ s.t. } \exists\ i\text{-}j \right). \tag{4.2}$$

On the forward sweep process of the algorithm, the total forward computation and communication time, t_f, will be

$$\tau_f = \sum_{i=0}^{n_p - n_d} \tau_i + \left(\sum \tau_{i\text{-}j} \ \ \forall\ i,\ j \text{ s.t. } \exists\ i\text{-}j \right). \tag{4.3}$$

where:

n_d : number of partitions which have no further adjacent downstream partitions

The time for one iteration of the distributed algorithm, $\tau_{b\text{-}f}$, is then

$$\tau_{b\text{-}f} = \tau_b + \tau_f \ . \tag{4.4}$$

Assuming that the time delay on the channels connecting any two partitions and the computation time at each partition is constant during the solution process, the total time, τ, for the algorithm to determine a solution will be dictated by the total number of iterations, K, required for the algorithm to converge to a solution. The total solution time is then:

$$\tau = K \cdot \tau_{b\text{-}f} \tag{4.5}$$

The five partition system as shown in Figure 4.3 can be used as an example to illustrate these wait times and delays. It can be seen that in order for the processor at partition 1 to calculate an equivalent load, it must wait for the adjacent downstream partitions, 3 and 4, to perform a power and communicate the required data at each iteration. This delay is caused both by the computation time required for downstream power flow and from delays in the communication channels.

4.1.4 CONVERGENCE CRITERIA

Convergence of the algorithm requires that the voltage at each boundary bus is within a specified tolerance for consecutive iterations. This requires that each equivalent source bus and each equivalent load bus in the network satisfies this condition. The equivalent source bus voltages are updated on the forward process of the algorithm and hence at the completion of this process, the system convergence is checked. The convergence criterion in terms of equivalent sources buses is then:

$$\left\| V_{i,0}^{(k+1)} \right| - \left| V_{i,0}^{(k)} \right\| \leq \varepsilon_1 \quad \& \quad \left| \angle V_{i,0}^{(k+1)} - \angle V_{i,0}^{(k)} \right| \leq \varepsilon_2 \quad \forall \; i = 1, \; 2, \; \ldots \; , n_p \, . \tag{4.6}$$

where:

$\left\| V_{i,0}^{(k)} \right\|$	:	equivalent source bus voltage magnitude, partition *i*
$\angle V_{i,0}^{(k)}$	:	equivalent source bus voltage phase angle, partition *i*
$\varepsilon_{1,2}$	:	convergence tolerance for voltage magnitude and phase angle, respectively

The equivalent source bus voltages are updated with the voltages of the equivalent load bus voltages at each iteration. Therefore, convergence is equivalently defined for the equivalent load bus voltages. The convergence criterion in terms of equivalent load buses is then:

$$\left\| V_{i,\,eq.\,load}^{(k+1)} \right| - \left| V_{i,\,eq.\,load}^{(k)} \right\| \leq \varepsilon_1 \quad \& \quad \left| \angle V_{i,\,eq.\,load}^{(k+1)} - \angle V_{i,\,eq.\,load}^{(k)} \right| \leq \varepsilon_2 \quad \forall \; i = 1, \; 2, \; \ldots \; , n_p \, . \tag{4.7}$$

where:

$\left\| V_{i,\,eq.\,load}^{(k)} \right\|$	:	equivalent load bus voltage magnitude, partition *i*
$\angle V_{i,\,eq.\,load}^{(k)}$	:	equivalent load bus voltage phase angle, partition *i*

The equivalence of the two conditions allows for a convergence check using either (4.7) or (4.6) at Step 9 of the algorithm presented above. An analysis of the convergence properties of the algorithm will be presented in the next chapter.

4.2 IMPLEMENTATION

The above method has been implemented using a custom software architecture integrating a Matlab [22] based distribution power flow solver [7] with National Instrument's LabView [23]. This architecture utilizes the computation power and function of Matlab with the communication functions of LabView. This section will review the details of the implementation including the software platforms, internet protocol, system architecture, and user interface.

The selection of these particular programs and the specific architecture was based solely on the need for a platform on which to test and implement the proposed method. When implemented on a set of real world distributed devices, the specific data structures and compiled program languages for each device will then need to be designed and utilized at each location. In addition, data sharing between the devices will need to be done in a universal form such as the Common Information Model which is currently under development for standardization of data sharing between power system applications [24].

4.2.1 SOFTWARE ARCHITECTURE

When performing a distributed analysis, each partition model and power flow solver must be located on a remote processor. In this implementation, remote personal computers (PC) were used to act as the remote processor at each location. Each PC then contains Matlab and LabView software, a case file describing the network partition to be solved, and a multi-phase distribution power flow solver. Communication ports are established between each adjacent partition's processor through which equivalent load and source values are passed. The Transfer Control Protocol (TCP) is used for all communication between each location in a server/client system architecture. Details of TCP can be seen in Appendix A. Figure 4.4 illustrates an example of this software architecture between three partitions.

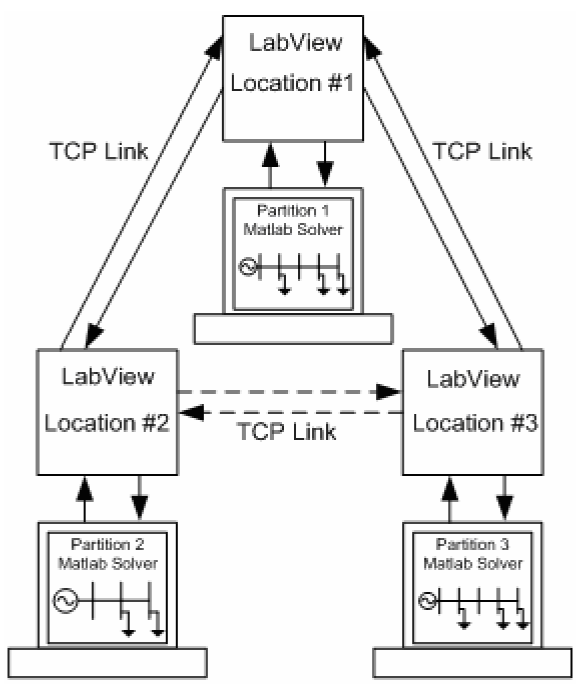

Figure 4.4: Example software architecture between three partitions

Matlab scripts are integrated within the LabView programming environment through a LabView function called *Matlab Script Node* at each remote LabView session. Through this function, LabView calls Matlab via an Active X [25] interface, allowing data to be passed to and from Matlab. The incorporation of Matlab scripts inside the LabView programming environment can be seen in Figure 4.5. Matrices can then be manipulated and passed freely between the LabView and Matlab environment.

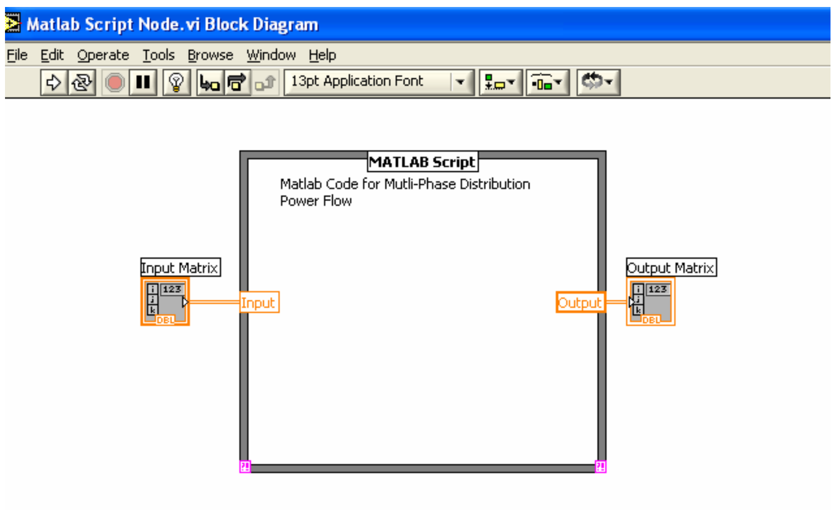

Figure 4.5: Matlab script node implementation

Initial conditions for the equivalent source bus voltages are set via a LabView front panel interface and passed into Matlab at each remote session. Power flow and post-processing operations are performed using the Matlab computation engine on the processor at each respective location. Results are then passed out to LabView as double precision floating decimal values. In LabView, the data is converted to a string, written to the TCP connection, and sent to an adjacent network partition in addition to being displayed on the front panel.

4.2.2 USER INTERFACE

At each location, a local user will set the power flow parameters and observe the results. To facilitate this, a LabView Graphical User Interface (GUI) has been created to allow the local user to set initial conditions, define open communication ports, set convergence tolerances, and view results. Figure 4.6 shows the user interface for Partition 1 of Figure 3.1.

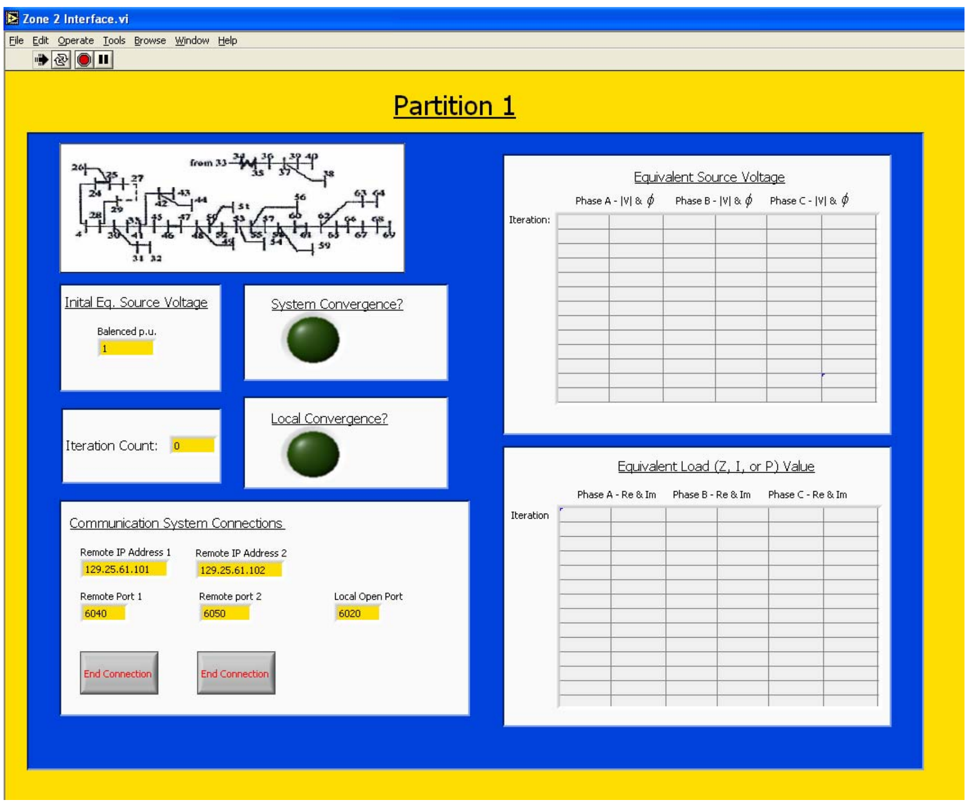

Figure 4.6: Graphical user interface at each location

The communication controls on the user interface tell the local processor which remote internet protocol (IP) addresses and ports on which to send and receive data. They also specify which local ports to open through which to send and receive data.

Coordination must be maintained between communication channels connecting each location to ensure proper data delivery.

5 CONVERGENCE PROPERTIES

When performing power flow studies to determine proper system planning and operation schemes, an accurate and convergent solution algorithm is required. This chapter will present convergence properties of the distributed power flow algorithm presented in this thesis. A review of the implicit Z-bus Gauss power flow will be presented followed by an investigation of the convergence properties of the distributed algorithm. The conditions under which a power flow on each partition will converge are established by showing that the system model is of the form of a Gauss type iterative method. Convergence of the whole distributed system will then be discussed. It will be shown that for radial distribution systems, if the distributed power flow converges, then it converges to the same solution as that of a traditional power flow on the original system.

5.1 IMPLICIT Z-BUS GAUSS POWER FLOW

To demonstrate the convergence of the algorithm within the solution space, the distributed power flow problem is formulated using the implicit Z-bus Gauss power flow to solve the power flow at each partition. This results in the system model as seen in (3.12). The implicit Z-bus Gauss power flow is based on the iterative solution to the system of linear nodal analysis equations given in (5.1).

$$\boldsymbol{YV} = \boldsymbol{I}(\boldsymbol{V}) \tag{5.1}$$

In distribution system analysis, the implicit Z-bus Gauss method is implemented by separating the voltage and current vectors into two parts, the first corresponding to the known voltages, for example the network sources, and the second corresponding to the remaining buses. This is seen in (5.2) below.

$$\left[\begin{array}{c|c} \boldsymbol{Y}_{11} & \boldsymbol{Y}_{12} \\ \hline \boldsymbol{Y}_{21} & \boldsymbol{Y}_{22} \end{array}\right]\left[\begin{array}{c} \boldsymbol{V}_{1} \\ \hline \boldsymbol{V}_{2} \end{array}\right]=\left[\begin{array}{c} \boldsymbol{I}_{1}(\boldsymbol{V}) \\ \hline \boldsymbol{I}_{2}(\boldsymbol{V}) \end{array}\right] \tag{5.2}$$

The power flow solution is then attained by solving for $\boldsymbol{V}_2$, given the known voltages $\boldsymbol{V}_1$. When there are no constant power devices in the network, this solution is directly attained by solving equation (5.3). If constant power loads exist in the network, they are linearized at each iteration based on an estimate of the bus voltages. The current injection vector, $\boldsymbol{I}_2$, then becomes a function of $\boldsymbol{V}_2$ as seen in (5.3).

$$\boldsymbol{V}_2^{(k+1)} = \boldsymbol{Y}_{22}^{-1}(\boldsymbol{I}_2(\boldsymbol{V}_2^{(k)}) - \boldsymbol{Y}_{21}\boldsymbol{V}_1) \tag{5.3}$$

The solution is achieved by iteratively updating $\boldsymbol{I}_2$ at each iteration until $\boldsymbol{V}_2^{(k)}$ convergences to a specified tolerance.

5.2 CONVERGENCE PROPERTIES

To investigate the convergence properties of the distributed power flow algorithm, the system model as presented in (3.12) and shown again in (5.4) below will be examined.

$$\begin{bmatrix} \boldsymbol{Y}_{00} & 0 & 0 & 0 \\ 0 & \boldsymbol{Y}_{11} & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \boldsymbol{Y}_{n_p n_p} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}_0 \\ \boldsymbol{V}_1 \\ \vdots \\ \boldsymbol{V}_{n_p} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I}_0(\boldsymbol{V}) + \boldsymbol{I}_{eq,1}(\boldsymbol{V}) \\ \boldsymbol{I}_1(\boldsymbol{V}) + \boldsymbol{I}_{eq,2}(\boldsymbol{V}) \\ \vdots \\ \boldsymbol{I}_{n_p}(\boldsymbol{V}) \end{bmatrix} \tag{5.4}$$

The system model takes the form of a Gauss type iterative method. Gauss type methods have been shown to converge given the admittance matrix of the system satisfies at least one of the following conditions:

- strict diagonal dominance, or
- diagonal dominance and irreducibility [19, 26], or
- the largest eigenvalue-modulus of the iteration matrix, derived from the admittance matrix, is less than unity [27].

In the below analysis, the condition of diagonal dominance and irreducibility will be focused on to demonstrate the conditions under which a power flow on each partition will converge and to make observations on the convergence of the distributed system as a whole.

Diagonal dominance of the admittance matrix requires:

$$\left|y_{ii}\right| \geq \sum_{\substack{i=1 \\ i \neq j}}^{n_n} \left|y_{ij}\right| \quad \forall \ j = 1, 2, ..., n_n \tag{5.5}$$

For the admittance matrix to be irreducible, there must not exist a permutation matrix, $\boldsymbol{\Pi}$ and integer $z \in \{1, ..., n_n - 1\}$ such that:

$$\boldsymbol{\Pi Y \Pi}^{\mathrm{T}} = \begin{bmatrix} \boldsymbol{Y}_{11} & \boldsymbol{Y}_{12} \\ 0 & \boldsymbol{Y}_{22} \end{bmatrix}, \tag{5.6}$$

where $\boldsymbol{Y}_{11}$ is of dimension z x z, and $\boldsymbol{Y}_{22}$ is of dimension $(n_n - z)$ x $(n_n - z)$ [29].

Most practical distribution systems satisfy the conditions of diagonal dominance and irreducibility because of fact that self-admittances of each bus are usually large compared to the mutual admittances. This is true because the majority of the buses in the system will:

- supply load with impedance typically much less than 1.0 p.u. with respect to the substation ratings, and
- be incident to more than one other bus in the network.

No-load buses can be accounted for using Kron's reduction techniques [30] prior to partitioning.

For the distributed system, in order for the power flow on each partition to converge, each block sub-matrix, $\boldsymbol{Y}_{ii}$, must satisfy the stated conditions. These sub-matrices each represent the admittance matrix of a partition i of the original system.

Therefore, given that the original system satisfies the required conditions, power flow on each partition will converge.

The system model in (5.4) cannot however be used to show absolute conditions for convergence of the distributed power flow. Convergence of the distributed power flow as defined in Chapter 4 requires that each boundary bus voltage is within a specified tolerance for consecutive iterations. This condition, in terms of equivalent source bus voltages is shown again in (5.7).

$$\left\| V_{i,0}^{(k+1)} \right| - \left| V_{i,0}^{(k)} \right\| \le \varepsilon_1 \quad \& \quad \left| \angle V_{i,0}^{(k+1)} - \angle V_{i,0}^{(k)} \right| \le \varepsilon_2 \quad \forall \ i = 1,\ 2,\ \ldots,\ n_p \tag{5.7}$$

Because of the increase in dimension to the original system model, the distributed system model violates the condition of irreducibility. The distributed algorithm though does not attempt to solve the distributed system model in its full form. The algorithm solves the distributed power flow through the sequential solution to each of the sub-systems representing the partitions of the original system. It has been shown that the feasible power flow solution for radial distribution networks always exists and is unique [31]. Therefore, it can be stated that given the power flow on each partition converges and the distributed power flow on the partitioned system converges, the distributed power flow will converge to the same solution as that of a traditional power flow on the original system.

To illustrate the above point, it can be shown that given the boundary buses of the distributed system converge to the same voltage as that of a power flow on the original system, then the system model of original system and the distributed system are equivalent. This concept can be demonstrated by looking at the example single phase system presented in Chapter 3 and shown again below in Figure 5.1. The nodal analysis equations for this system are repeated in (5.8) below.

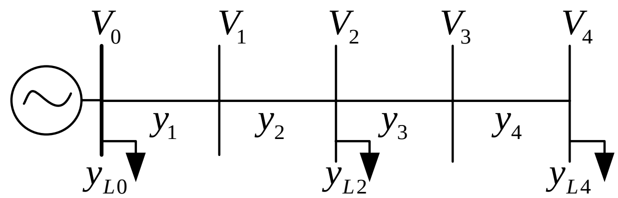

Figure 5.1: Single-phase 5 bus system

$$\begin{bmatrix} y_1+y_{L0} & -y_1 & 0 & 0 & 0 \\ -y_1 & y_1+y_2 & -y_2 & 0 & 0 \\ 0 & -y_2 & y_2+y_3+y_{L2} & -y_3 & 0 \\ 0 & 0 & -y_3 & y_3+y_4 & -y_4 \\ 0 & 0 & 0 & -y_4 & y_4+y_{L4} \end{bmatrix} \begin{bmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} I_0(\boldsymbol{V}) \\ I_1(\boldsymbol{V}) \\ I_2(\boldsymbol{V}) \\ I_3(\boldsymbol{V}) \\ I_4(\boldsymbol{V}) \end{bmatrix} \tag{5.8}$$

The system is partitioned into three partitions, seen again in Figure 5.2. The nodal analysis equations of the distributed system are shown in (3.11) and repeated below in (5.9).

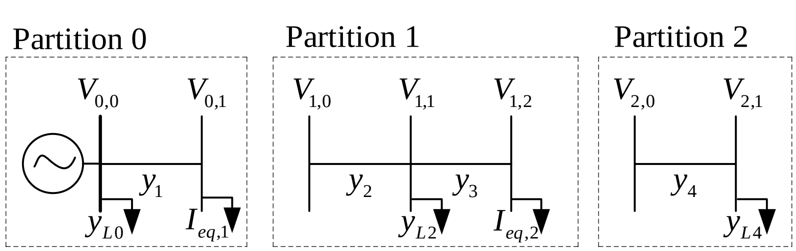

Figure 5.2: Partitioned single-phase 5 bus system

$$\left[\begin{array}{cc|ccc|cc} y_1+y_{L0} & -y_1 & 0 & 0 & 0 & 0 & 0 \\ -y_1 & y_1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & y_2 & -y_2 & 0 & 0 & 0 \\ 0 & 0 & -y_2 & y_2+y_3+y_{L2} & -y_3 & 0 & 0 \\ 0 & 0 & 0 & -y_3 & y_3 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & y_4 & -y_4 \\ 0 & 0 & 0 & 0 & 0 & -y_4 & y_4+y_{L4} \end{array}\right]\left[\begin{array}{c} V_{0,0} \\ V_{0,1} \\ \hline V_{1,0} \\ V_{1,1} \\ V_{1,2} \\ \hline V_{2,0} \\ V_{2,1} \end{array}\right]=\left[\begin{array}{c} I_{0,0}(\boldsymbol{V}) \\ I_{0,1}(\boldsymbol{V})+I_{eq,1}(\boldsymbol{V}) \\ \hline I_{1,0}(\boldsymbol{V}) \\ I_{1,1}(\boldsymbol{V}) \\ I_{1,2}(\boldsymbol{V})+I_{eq,2}(\boldsymbol{V}) \\ \hline I_{2,0}(\boldsymbol{V}) \\ I_{2,1}(\boldsymbol{V}) \end{array}\right] \tag{5.9}$$

The distributed model differs from the model of the original system in that it has a block diagonal admittance matrix and increased dimension because of the presence of the boundary buses. The original network was partitioned at Bus 1 and Bus 3. Therefore, given that the boundary buses converge to the same voltage as a power flow on the original system, at convergence:

$$V_1 = V_{0,1} = V_{1,0} \tag{5.10}$$

$$V_3 = V_{1,2} = V_{2,1} \tag{5.11}$$

Under the assumption that the boundary buses have converged to the same voltage as the corresponding bus of the original system, then because of the radial topology of the system, every other bus in the partition will also match the original system solution. To demonstrate the equivalence of the two systems, the equation corresponding to the second row of the (5.8) and (5.9) are extracted and can seen below in (5.12) and (5.13), respectively.

$$-y_1V_0 + y_1V_1 + y_2V_1 - y_2V_2 = I_1(\boldsymbol{V}) \tag{5.12}$$

$$-y_1V_{0,0} + y_1V_{0,1} = I_{0,1}(\boldsymbol{V}) + I_{eq,1}(\boldsymbol{V}) \tag{5.13}$$

The terms in the distributed system equation differ from that of the original system, however, given the fact that the bus voltages converge to same values as those of the original system and given the definition of the equivalent load value established in Chapter 3, it can be seen:

$$I_{eq,1}(\boldsymbol{V}) = y_2\left(V_{1,0} - V_{1,1}\right) = y_2\left(V_1 - V_2\right) \tag{5.14}$$

Substituting (5.14) into (5.13), the two equations are then seen to be equivalent. A similar relationship can be expressed for the remaining equations; therefore, the two systems are equivalent.

Simulation results will be provided in the next chapter to experimentally validate that the distributed power flow algorithm will converge to the same solution as the original system under varying loading conditions of the original system and varying numbers of partitions.

6 SIMULATION RESULTS

Using the implementation presented in Chapter 4, simulation results obtained from a range of test cases will be used to evaluate the performance of the proposed method. This chapter will present the results and discuss them with respect to the accuracy of the proposed distributed power flow versus the solution of a traditional power flow. The number of iterations and total time required for the distributed power flow to converge will be provided for each case. The specific sets of simulations that are presented were chosen to demonstrate the effect the load models and number of partitions has on the accuracy and calculation time of the distributed analysis.

The details of a multi-phase 97 bus test system will be presented followed by two cases of the system partitioned into 3 and 5 subsystems, respectively. The solution of the resulting partitioned system will be compared to that of the original system and the total time required for the algorithm to converge will be provided. Lastly, observations from the simulation results will be stated. Table 6.1 provides an overview of the 7 simulations cases which were conducted.

Table 6.1: Overview of simulations

Simualtion No.	Original System Load Models	Original System IC (p.u. bal.)	No. of Partitions	Partitioned System Eq. Load Models	Partitioned System IC (p.u. bal.)	Orig. and Part. Convergence Tolerances
1	Constant Z	1.0	3	Eq. Impedance	1.0	10e-10
2	Constant I	1.0	3	Eq. Current	1.0	10e-10
3	Constant P	1.0	3	Eq. Power	1.0	10e-10
4	Constant Z	1.0	5	Eq. Impedance	1.0	10e-10
5	Constant I	1.0	5	Eq. Current	1.0	10e-10
6	Constant P	1.0	5	Eq. Power	1.0	10e-10
7	Constant Z and P	1.0	5	Eq. Current	1.0	10e-10

6.1 SIMULATION SET-UP

The distributed power flow simulations were carried out on a test platform using a set of computers with identical specifications connected together via a local area network (LAN). Each PC operated under Windows XP on a Pentium 4 processor with a rated speed of 2.0 gigahertz and random access memory of 1.0 gigabytes. For each simulation, 10 trials were conducted in order to provide an average total time for convergence of the algorithm on the given system.

The total time required for each simulation to converge was recorded for each simulation. All power flows on the un-partitioned original system use the implicit Z-bus Gauss method with a convergence tolerance of 10^{-10}. In addition, the implicit Z-bus Gauss method is used as the power flow algorithm at each partition, with a convergence tolerance in (4.7) of $\varepsilon_1 = \varepsilon_2 = 10^{-10}$. The total time for the simulation is measured from the start of the first power flow on a partition to the time when all partitions have converged as outlined in Chapter 4.

It is important to note that the presented timing results were obtained on a controlled local area network with no additional network traffic. In a practical implementation, computation and communication delays will vary based on the processing power of each device, distance between devices, network traffic, and communication protocols used to share data.

6.2 UN-PARTITIONED ORIGINAL SYSTEM

The test system used in all the simulations to follow is the 97 bus, 249 node system shown in Figure 3.1 and repeated in Figure 6.1 below.

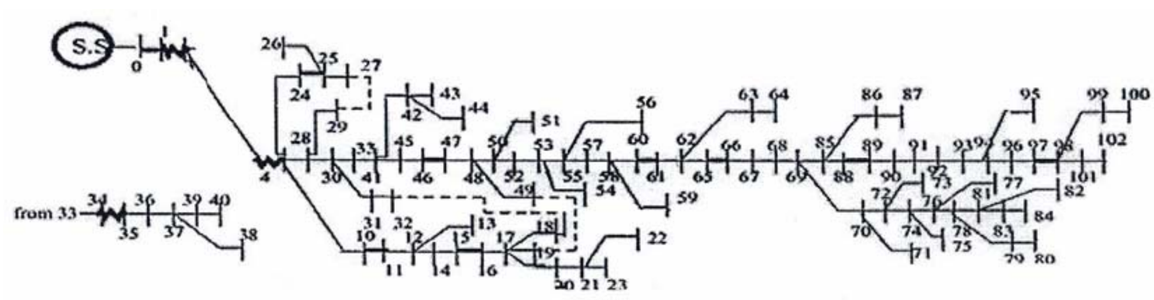

Figure 6.1: 97 Bus, 249 node test distribution system

The total nominal load of the system is 6,314 kW and 3,373 kVar with voltage levels from 115 kV to 4.8 kV. A detailed list of the number of each component can be seen in Table 6.2.

Table 6.2: 97 bus test system – component count

Component	Count
Distribution Lines	84
Switches	10
Transformers	3
Balanced Loads	5
Unbalanced Load	48
Capacitor Banks	5

In the following simulations, this system will be partitioned and then solved according to the algorithm presented in Chapter 4. The solution of the distributed power flow will then be compared with that of a traditional power flow run on the un-partitioned system.

6.3 THREE PARTITION SYSTEM

For the first set of simulations, the network is divided into three partitions seen in Figure 6.1 below. For a given load distribution and the total nominal power value described above, three simulations were conducted in which all the loads of the test system were modeled as either constant impedance, constant current, or constant power, respectively.

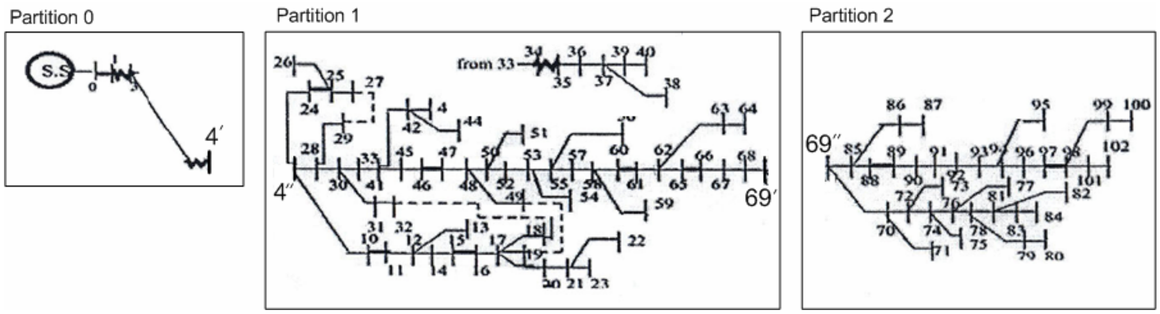

Figure 6.2: Test system partitioned into three partitions

First, the load model used for each load of the original system was selected as constant impedance. Figure 6.3 shows the power flow results for the phase *a* voltage magnitude at each bus of the original system. All of the network loads were modeled as constant impedance so an explicit solution to the power flow was attained using one iteration of the implicit Z-bus Gauss method.

The partitioned system was solved with the distributed power flow algorithm using an implicit Z-Bus Gauss power flow at each processor. The equivalent loads were modeled at each partition as constant impedance. The distributed power flow required 11 iterations to converge to a solution, averaging 46.115 seconds over all trials on the given test platform. Figure 6.3 shows the distributed power flow results for the phase *a* voltage magnitude at each bus. At the converged solution, the three-phase bus voltages converged, within the convergence tolerance, to the same solution as the power flow on the original system. The max absolute error of voltage magnitude between the distributed power flow and a power flow on the original system is displayed on the graph.

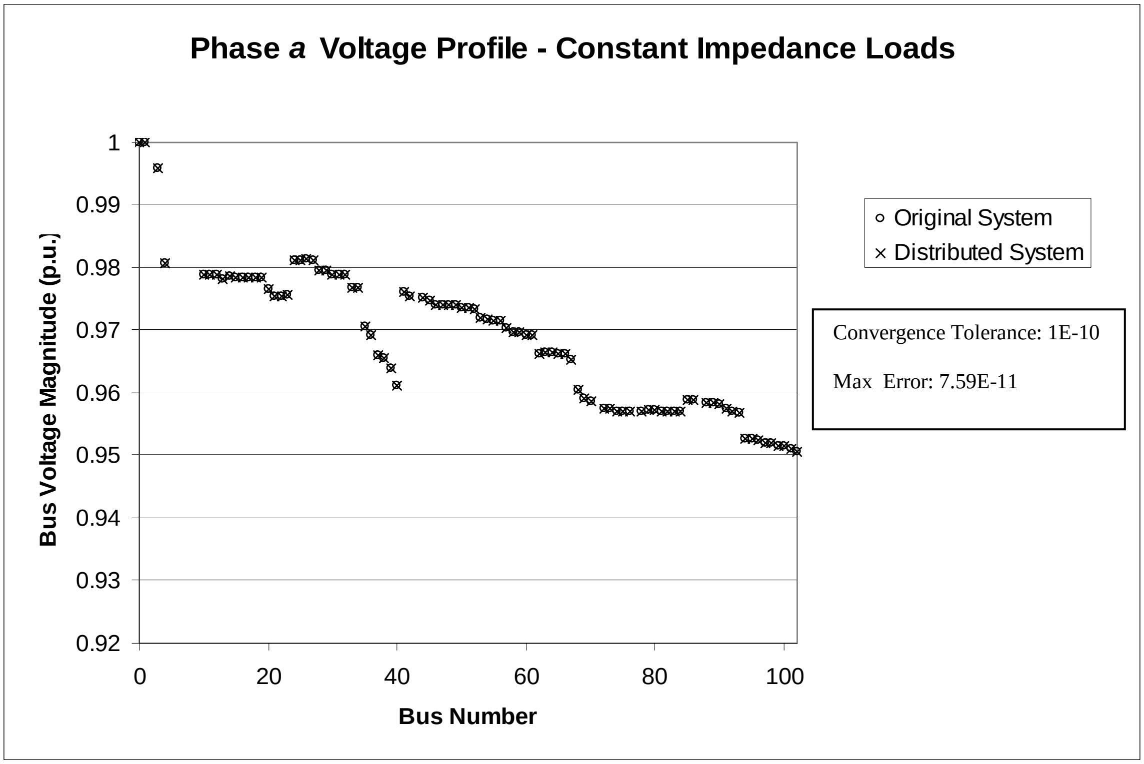

Figure 6.3: Three partition simulation - constant impedance load models

Next, constant current load models are selected for each load of the original system. Constant current loads again allowed for an explicit solution to the power flow to be obtained by one iteration of the implicit Z-bus Gauss power flow method. The equivalent loads in the three partition system were now modeled as constant current. The distributed power flow required 13 iterations to converge to a solution, averaging 56.241 seconds over all trials. Figure 6.4 shows the power flow results for the phase *a* voltage magnitude at each bus of the original network and the partitioned system. The max absolute error of voltage magnitude between the distributed power flow and a power flow on the original system is displayed on the graph.

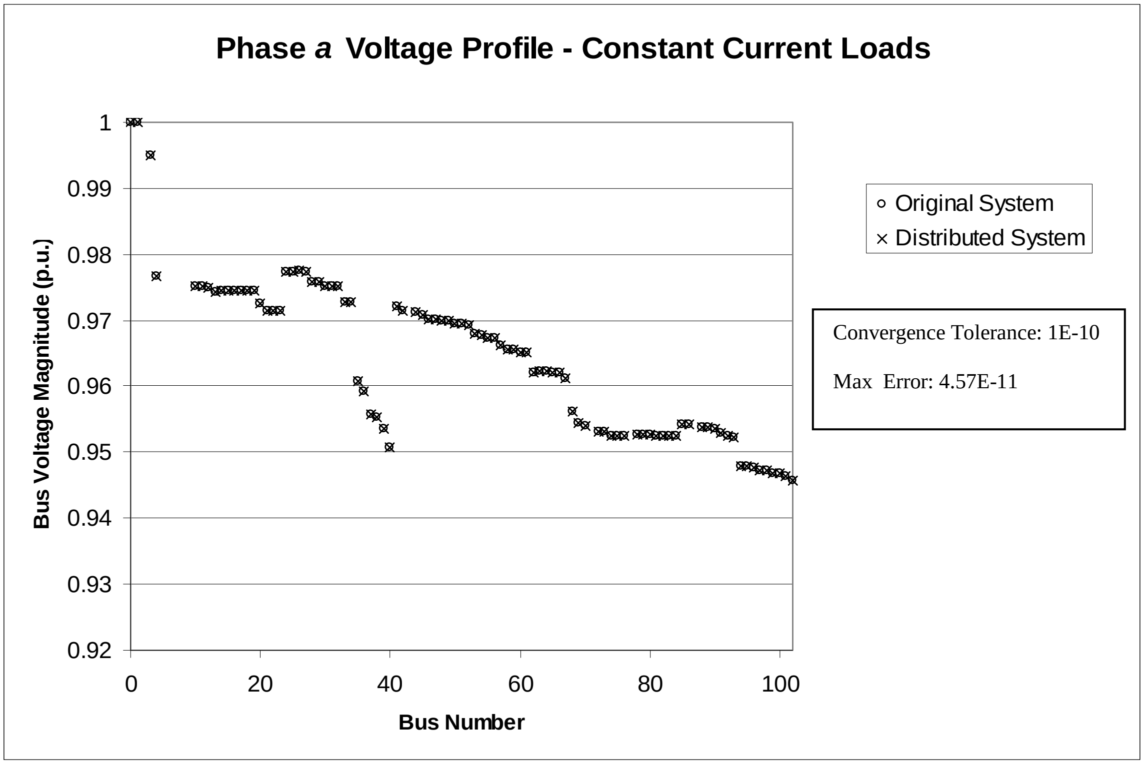

Figure 6.4: Three partition simulation - constant current load models

Then, constant power models were selected for each load of the original system. Solving the original system required 13 iterations to converge within a tolerance of 10^{-10} starting from a flat balanced 1.0 voltage profile. Each of the equivalent loads in the three partition system now used a constant power load model. The distributed power flow required 7 iterations to converge averaging 27.015 seconds over all trials with a maximum of 8 implicit Z-bus Gauss iterations for the power flow at any one processor. Figure 6.5 shows the results for the phase *a* voltage magnitude at each bus of the original network and the partitioned system. Again, the three-phase bus voltages converged to the same solution, within the convergence tolerance, as that of the power flow on the original system.

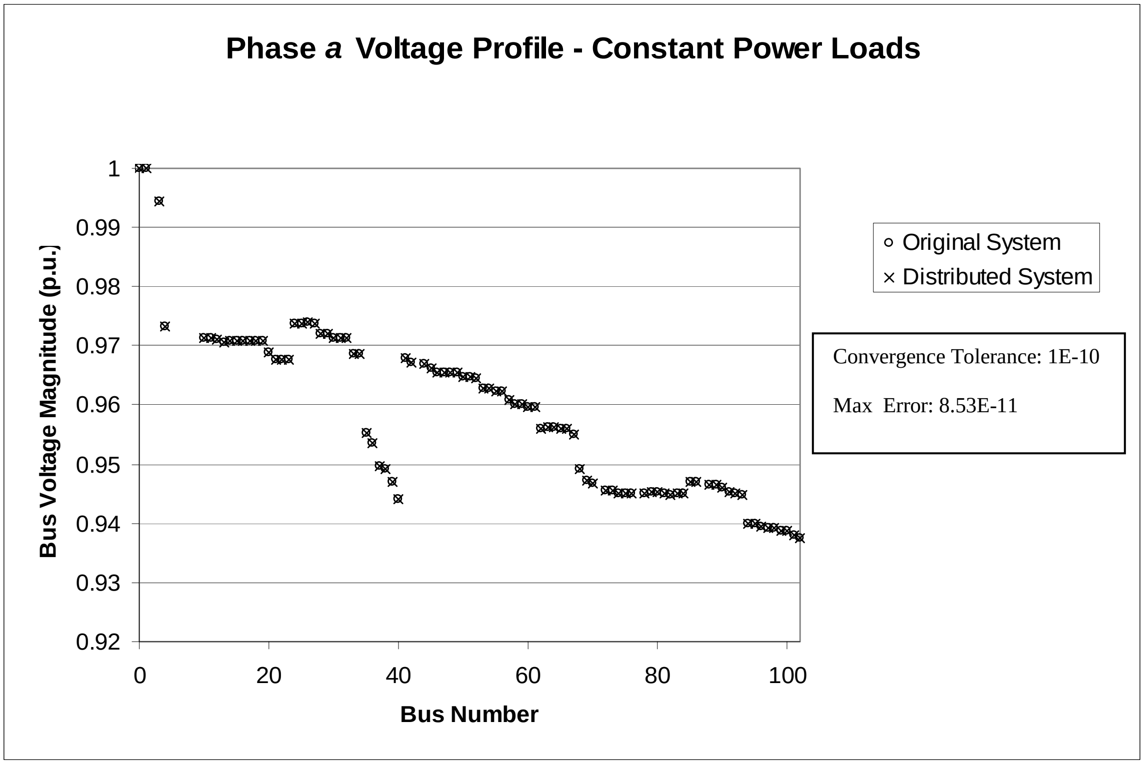

Figure 6.5: Three partition simulation - constant power load models

Table 6.3 summarizes the total number of iterations and required time to converge to a solution for the 3 partition networks on the given test set-up.

Table 6.3: Three partition system results summary

Load Model	Iteration Count	Average Total Time (s)
Constant Impedance	11	46.115
Constant Current	13	56.241
Constant Power	7	27.012

6.4 FIVE PARTITION SYSTEM

For the next set of simulations, the original network was divided into five partitions as seen in Figure 6.6. In this case, multiple adjacent downstream partitions were present from each equivalent load bus, hence the equivalent loads from each would need to be aggregated. Four simulations were run for this partitioning scheme. The first three simulations, modeled all the system loads as either constant impedance, constant current, or constant power, respectively. The last simulation used a mixture of constant impedance and constant current load models in for the original system.

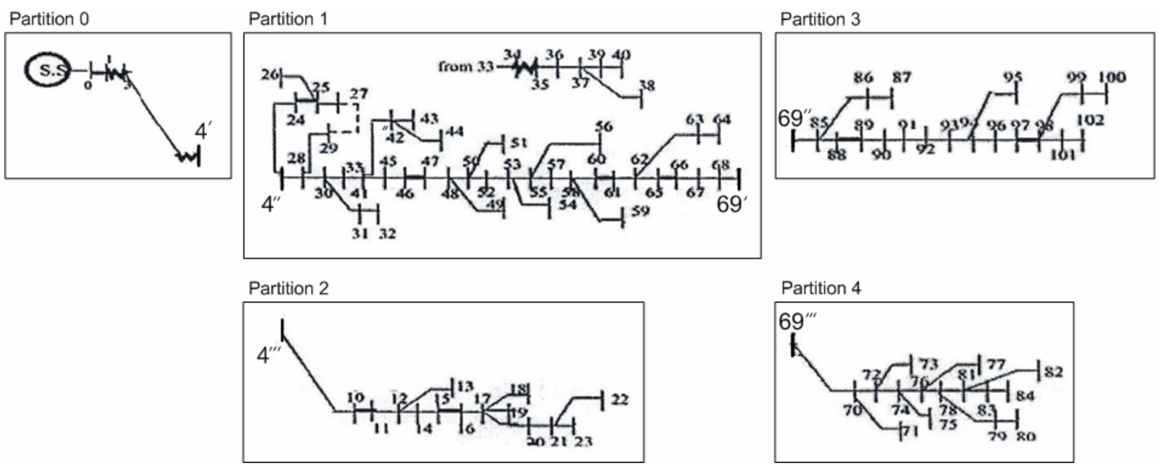

Figure 6.6: Test system partitioned into five partitions

For the first three simulations of this set, in terms of accuracy, the increase to the number of partitions did not affect the solution of the distributed power flow. At convergence, the distributed power flow results matched the results of the three partition system, within the convergence tolerance, for each of the given load models of the original system. Table 6.4 summarizes the iteration count and total time required to converge to a solution for each of these cases.

For the last simulation, all the loads of the original system were modeled as constant impedance except those corresponding to the buses in Partition 4 which were modeled as constant power. In this case, a mixture of load models are present which require the equivalent loads to be represented as equivalent current injections. The distributed power flow required 10 iterations to converge within a tolerance of 10^{-10} starting from a flat balanced 1.0 voltage profile. The power flow converged to the same solution as a power flow on the original system within the convergence tolerance requiring an average of 41.706 seconds.

Table 6.4: Five partition system results summary

Load Model	Iteration Count	Average Total Time (s)
Constant Impedance	11	40.256
Constant Current	10	36.613
Constant Power	8	28.809
Mixed – Z and P	10	41.706

6.5 OBSERVATIONS ON SIMULATION RESULTS

The results show that the distributed power flow will converge to the same solution as a traditional power flow under a range of conditions on the original system including different numbers of partitions and different load models. The results validate the proposed models and show that the algorithm is robust under various conditions. As expected, the results have also shown that different conditions will affect the number of iterations and total time required for the algorithm to converge. A summary of the simulation results may be seen below in Table 6.5.

Table 6.5: Summary of simulation results

No. of Partitions	Load Model	Max Error	Iteration Count	Average Total Time (s)
3	Constant Impedance	7.59E-11	11	46.115
3	Constant Current	4.57E-11	13	56.241
3	Constant Power	8.53E-11	7	27.012
5	Constant Impedance	1.9E-14	11	40.256
5	Constant Current	1.16E-12	10	36.613
5	Constant Power	2.97E-11	8	28.809
5	Mixed – Z & P	2.45E-11	10	41.706

For constant impedance loads, the iteration count remained the same for both the 3 and 5 partition systems, but the total time required to converge to a solution decreased as the number of partitions increased. Under the assumption that time delays in the communication channels are assumed identical for each case, this speed-up would be expected. If network time delays are identical, according to the timing models presented in Chapter 4, the difference in the total time would need to be accounted for in the computation at each processor. For the 3 partition case, the networks contained in two of the three partitions were larger than any of those contained in the 5 partition system. With $O(n^3)$ computational complexity of the implicit Z-bus Gauss power flow, the required computation of the processors involved in calculating power flow at these partitions is significantly increased. The increase in partition number therefore decreased the complexity of the problem at each partition resulting in a reduction of total solution time.

For the case of constant current loads, both the iteration count and total time were seen to decrease as the number of partitions increased. This decrease in total time with iteration count is expected because less iterations will reduce both the communication and computation lags. The speed-up may again be explained due to the reasons stated above.

Lastly, when all constant power loads are present an increase in the number of partitions caused both the iteration count and total time to increase. This result was unexpected because of the counter reasons stated above. The change in total time however varied only slightly compared to the changes observed for the other load types. In addition, the simulations with constant power loads resulted in significantly lower iteration counts and total times than the simulations for the other load models in both the three partition and five partition cases.

7 CONCLUSIONS

7.1 CONCLUSIONS

This thesis has presented component models, a solution algorithm, an investigation of convergence properties, and simulation results for a distributed multi-phase distribution power flow solver. The method is designed to utilize the distributed intelligent devices located throughout the distribution system to calculate distribution power flow. This will support the coordination of local and system wide control decisions. The method partitions the distribution system, modeling the portion of the network in each partition in detail using existing multi-phase distribution system component models, while modeling the effect of the rest of the network on the partition through equivalents. To that end, state of the art distribution system power flow models were reviewed and their network control schemes were discussed. In addition, existing methods for network equivalencing were presented along with a discussion of their application to the proposed method.

For distributed analysis, the models and parameter calculation procedures for equivalent loads and equivalent sources were presented along with a model for the distributed system. To solve the distributed power flow, an iterative algorithm was presented which detailed the solution process along with the convergence criterion. Analysis of the convergence properties of the algorithm shows that if the distributed power flow converges, then it will converge to the same solution as that of a traditional power flow on the original system. An implementation of the method was presented which used commercially available software to execute the solution algorithm.

Lastly, simulation results were presented from the implementation of the method on a 97 bus test distribution system. The simulation results were used to compare the solution of the distributed algorithm versus the solution of a traditional power flow on the given test system. The results showed that the distributed method converged to the same solution as a traditional power flow on the test system under various conditions of the original system such as varying load types and varying numbers of partitions. The load models of the original system and equivalent load models of the distributed analysis did however affect the total number of iterations and time required for the distributed power flow to converge.

7.2 SUMMARY OF RESEARCH CONTRIBUTIONS

The work in this thesis presented a new method for calculating multi-phase distribution power flow using remotely distributed processors and was designed to advance the field of distribution automation and analysis. Specifically, this thesis presented and discussed:

- a network partitioning process which partitions a distribution system based on the location of distributed intelligent devices located throughout the system
- derivations of equivalent source and equivalent load models used to represent the effects of the network not retained in each partition,
- a distributed multi-phase distribution power flow algorithm which uses the proposed models to calculate the operating state of a system,
- an investigation of the convergence properties of the proposed distributed algorithm

- verification of the proposed method on a test distribution system demonstrating its accuracy and timing considerations under various conditions of the original system

7.3 FUTURE WORK

The presented work has demonstrated the effectiveness of the proposed distributed method for calculating multi-phase distribution power flow. Further investigation into the extension of the work for use in meshed system as was first described in Chapter 3 is suggested. After identifying and properly modeling the boundary buses in a meshed system, modified algorithms would need to be designed to incorporate boundary matching at each iteration and to dictate a partition calculation order. With meshed system models and algorithms in place, applications for the distributed power flow concept for interconnected power transmission system can then be investigated.

To address the operation of the distributed power flow under communication system contingencies, the development of redundancy schemes for the communication and computation systems should be investigated. This will allow for the operation of the algorithm despite failures in processing hardware and communication channels. Alternate algorithms should also be developed to handle situations when communication is lost between a subset of partitions allowing the algorithm to proceed using the latest known information.

To bring the distributed power flow further towards real world implementation, the standardization of the shared data required to implement the method, such as facilitated by CIM models, should be investigated. This will allow for devices operating on different software platforms and architectures to work together in the proposed manner. In addition, further work may be done to investigate online control and coordination studies using the distributed power flow for distribution management applications such as network reconfiguration, service restoration, and capacitor control.

LIST OF REFERENCES

[1] D. Bassett, K. Clinard, J. Grainger, S. Purucker, D. Ward, "Tutorial Course: Distribution Automation," *IEEE Tutorial Publication* 88EH0280-8-PWR, 1988

[2] Roytelman, I., Melnik, V., Lee, S.S.H., Lugtu, R.L., "Multi-Objective Feeder Reconfiguration by Distribution Management System," *IEEE Trans. on Power Systems*, Vol.11, Issue 2, May 1996 pp. 661 - 667

[3] Wang, J. C., Chiang, H. D., Miu, K. N., Darling, G. R., "Capacitor Placement and Real Tune Control in Large-Scale Unbalanced Distribution Systems: Loss Reduction Formula, Problem Formulation, Solution Methodology and Mathematical Justification," *IEEE Trans. on Power Delivery*, vol. 12, no. 2, pp. 953–958, 1997.

[4] Fukuyama, Y., Nakanishi, Y., Chiang, H-D., "Fast Distribution Power Flow Using Multiple Processors," *Electrical Power and Energy Systems*, Vol. 18, No. 5, 1996 pp. 331-337

[5] Y. Mao, "Protection System Design for Power Distribution Systems in the Presence of Distributed Generation" *Doctoral Disseration,* Drexel University, July 2005.

[6] D'Antona, G., Monti, A., Ponci, F., Rocca, L., "A Distributed State Estimator for Electric Power Systems in Avionic and Naval Applications" Instrumentation and Measurement Technology Conference Sorrento, Italy 24-27 April 2006

[7] R. D. Zimmerman, "Comprehensive Distribution Power Flow: Modeling, Formulation, Solution Algorithms and Analysis, " *Doctoral Dissertation*, Cornell University, Jan. 1995.

[8] Chen, M., Dillon, W., "Power System Modeling, " *Proceeding of the IEEE*, Vol. 62, No. 7, July 1974

[9] A. R. Bergen, "Power Systems Analysis," Prentice Hall, 1986.

[10] D. Kirschen, R. Allan, G. Strbac, "Contribution of Individual Generators to Loads and Flows," *IEEE Tran. Power Systems*, Vol.12, No.1, February 1997, pp 52-60.

[11] Tong, S., Kleinberg, M., Miu, K. "A Distributed Slack Bus Model and It's Impacts on Distribution System Application Techniques" IEEE International Symposium on Circuits and Systems 2005, 23-26 May 2005, pp. 4743-4746

[12] Ward J. B., “Equivalent Circuits for Power Flow Studies,” AIEE Transactions on Power Apparatus and Systems, Vol. 68, pp. 373-382, 1949.

[13] Deckman, S., Pizzolante, A., Monticelli, A., Stott, B. Alsac. O., “Studies On Power System Load Flow Equivalents,” *IEEE Trans. on Power Apparatus and Systems*, Vol PAS-99, No. 6, Nov/Dec 1980, pp. 2301 - 2309

[14] Deckman, S., Pizzolante, A., Monticelli, A., Stott, B. Alsac. O., “Numerical Testing of Power Flow Load Flow Equivalents,” *IEEE Trans. on Power Apparatus and Systems*, Vol PAS-99, No. 6, Nov/Dec 1980, pp. 2292 - 2300

[15] P. Dimo, “Nodal Analysis of Power Systems,” Abacus Press, England. 1975

[16] Sun, D. I. H., Abe, S., Shoults, R. R., Chen, M. S., Eichenberger, P., Farris, D., “Calculation of Energy Losses in a Distribution System,” *IEEE Trans. on Power Apparatus and Systems*, Vol. PAS-99, No. 4, July/August 1980, pp. 1449-1460

[17] Chen, T. H., Chen, M. S., Hwang, K. J., Kotas, P., Chebli, E., “Distribution System Power Flow Analysis - A Rigid Approach,” *IEEE Trans. on Power Delivery*, Vol. 6, No. 3, July 1991, pp. 1146-1152

[18] Zimmerman, R. D., Hsiao-Dong Chiang, “Fast decoupled power flow for unbalanced radial distribution systems,” *IEEE Trans. on Power Systems*, Vol.10, Issue 4, Nov. 1995 pp. 2045 - 2052

[19] B. Stott, “Review of Load-Flow Calculation Methods”, *Proceedings of the IEEE*, Vol. 62, No. 7, July 1974, pp. 916-929

[20] D. P. Bertsekas, J. N. Tsitsiklis, “Parallel and Distributed Computation: Numerical Methods,” Athena Scientific, 1997.

[21] Carullo, S., Nwankpa, C., “Experimental Validation of a Model for an Information- Embedded Power System” *IEEE Trans. Power Delivery*, Vol. 20, No. 3, July 2005, pp. 1853-1863

[22] Matlab 7.0.1, http://www.mathworks.com/products/matlab/, 2006.

[23] LabView 7.1, http://www.ni.com/labview/, 2005.

[24] Wang, X., Schulz, N. Neumann, S., “CIM Extensions to Electrical Distribution and CIM XML for the IEEE Radial Test Feeders,” *IEEE Trans. On Power Systems*, Vol. 18, No. 3, August 2003

[25] J. Anderson, “ActiveX Programming With Visual C++ 5.0” Que Pub 1997

[26] Bagnara, R. “A Unified Proof for the Convergence of Jacobi and Gauss Siedel Methods,” *SIAM Review*, Vol. 37, No. 1, March 1995, pp. 93-97

[27] W. F. Tinney and W. L. Powell, “Convergence of Matrix Iterations Subject to Diagonal Dominance” *SIAM Journal of Numerical Analysis*, Vol. 10, No. 3., June 1973, pp. 478-484

[28] J. D. Glover, M. S. Sarma, “Power System Analysis and Design” Brooks/Cole 2002

[29] P. Lancaster and M. Tismenetsky, "The Theory of Matrices, Second Edition with Applications" Academic Press. 1985

[30] H. Saadat, “Power Systems Analysis” McGraw-Hill 2002

[31] Miu, K. N., Chiang, H. D., "Existence, Uniqueness, and Monotonic Properties of the Feasible Power Flow Solution for Radial Three-Phase Distribution Networks" *IEEE Trans. on Circuits and Systems - I: Fundamental Theory and Applications*, Vol. 47, No. 10, Oct. 2000

[32] D. Bertsekas, R. Gallager, “Data Network” Prentice Hall 1992

A APPENDIX A: LIST OF SYMBOLS

n_{ph}	:	number of phases of corresponding bus or branch
n_p	:	number of partitions of the distributed system
n_b	:	number of buses of the distributed system
n_n	:	number of nodes of the distributed system
n_n^i	:	number of nodes, partition *i*, of the distributed system
$\boldsymbol{Z}_{ik}$	:	(n_{ph} x n_{ph}) series impedance matrix of a distribution line connecting bus *I* and bus *k*
$\boldsymbol{Y}_{ik}^{sh}$	:	(n_{ph} x n_{ph}) shunt admittance matrix modeling line charging at each end of the line connection bus *i* and bus *k*
$\boldsymbol{Y}_{ik}^{phase}$	:	admittance matrix model for a line connecting bus *i* to bus *k*
$\boldsymbol{Y}_{k}^{xfmr}$	:	admittance matrix model for a transformer connecting bus *i* and bus *k*
superscript *	:	complex conjugate
subscript $_{\text{nom}}$	:	nominal value
./	:	element-wise division
.*	:	element-wise multiplication
$\overline{\bullet}$	:	specified value
$\overline{\boldsymbol{S}}_{Lk}$	:	specified nominal value of load power
$\overline{\boldsymbol{I}}_{Lk}$	:	specified nominal value of load current

$\bar{\boldsymbol{y}}_{Lk}$: specified nominal value of load admittance

y_j : complex admittance of branch *j*, un-partition system

V_j : complex voltage at bus *j*, un-partition system

$I_j(\boldsymbol{V})$: complex current injection, bus *j*, un-partition system

$V_{i,j}$: complex voltage at bus *j*, partition *i*

$I_{i,j}$: complex current injection at bus *j*, partition *i*

$\boldsymbol{V}_{i,0}^{(k)}$: complex (n_{ph} x 1), multi-phase equivalent source bus voltage vector of partition *i*, iteration *k*

$\boldsymbol{V}_{i-1,\,eq.\,load}^{(k)}$: complex (n_{ph} x 1), multi-phase equivalent load bus voltage vector of the corresponding upstream partition *i*-1, iteration *k*

k : iteration number

$\boldsymbol{Y}_{eq,\,i}^{(k)}$: complex (n_{ph} x *1*) multi-phase equivalent admittance vector, partition i, iteration *k*

$\boldsymbol{S}_{eq,\,i}^{(k)}$: complex (n_{ph} x *1*) multi-phase equivalent power injection vector, partition i, iteration *k*

$\boldsymbol{I}_{eq,\,i}^{(k)}$: complex (n_{ph} x 1) multi-phase equivalent current injection vector, partition i, iteration *k*

B_i : Set of all branches connected directly to the equivalent source bus, partition i

$\boldsymbol{V}^{(k)}$: complex (n_n x 1) partitioned system voltage vector, iteration *k*

$\boldsymbol{I}_{j}^{DG,C,L}$: complex (n_{ph} x 1), multi-phase currents injected by distributed generators, capacitors, and loads, respectively, on bus *j*

$\boldsymbol{I}_{i,j}^{DG,C,L}$: complex (n_{ph} x 1), multi-phase currents injected by distributed generators, capacitors, and loads, respectively, on bus *j*, partition *i*

D_i : set of all adjacent partitions downstream from partition *i*

$\boldsymbol{Y}_{ii}$: complex (n_n^i x n_n^i) admittance matrix, partition *i*

τ_i : total time to run a power flow and perform the required post-processing on partition *i*,

$\tau_{i\text{-}j}$: time delay in the communication channel between two adjacent partitions *i* and *j*

t_b : total backwards computation and communication time, on the backward sweep process of the algorithm

t_f : total forward computation and communication time forward sweep process of the algorithm

n_d : number of partitions which have no further adjacent downstream partitions

$\tau_{b\text{-}f}$: time for one iteration of the distributed algorithm

τ : total time for the algorithm to determine a solution

K : total number of iterations required for the algorithm to determine a solution

B APPENDIX B: TRANSFER CONTROL PROTOCOL

TCP is part of the transport layer placed on top of the Internet Protocol. It was first established in 1982 and later standardized by the United States Department of Defense [31]. TCP data is organized as a stream of bytes and was designed to provide reliable stream oriented connections. The format of a segment of the TCP/IP can be seen in Figure B.1. In the implementation described in Chapter 4, the built-in high level LabView TCP functions handle the specifics of the protocol, requiring the user to only specify the destination IP address, destination port, and data to be transferred.

<table>
<tr><td colspan="4">Source IP Address</td></tr>
<tr><td colspan="4">Destination IP Address</td></tr>
<tr><td colspan="3">Zero | Protocol</td><td>TCP Length</td></tr>
<tr><td colspan="3">TCP Source Port</td><td>TCP Destination Port</td></tr>
<tr><td colspan="3">TCP Message Length</td><td>TCP Check Sum</td></tr>
<tr><td colspan="3">Source Port</td><td>Destination Port</td></tr>
<tr><td colspan="4">Sequence Number</td></tr>
<tr><td colspan="4">Request Number</td></tr>
<tr><td>Data Offset</td><td>Reserved</td><td>Control</td><td>Window</td></tr>
<tr><td colspan="3">Check sum</td><td>Urgent Pointer</td></tr>
<tr><td colspan="4">Options (if any)</td></tr>
<tr><td colspan="4">Data</td></tr>
</table>

Figure B.1: TCP/IP segment[32]