Mississippi State University Scholars Junction

Theses and Dissertations

Theses and Dissertations

8-12-2016

# PGNME: A Domain Decomposition Algorithm for Distributed Power System Dynamic Simulation on High Performance Computing Platforms

Brian Shane Sullivan

Follow this and additional works at: https://scholarsjunction.msstate.edu/td

#### **Recommended Citation**

Sullivan, Brian Shane, "PGNME: A Domain Decomposition Algorithm for Distributed Power System Dynamic Simulation on High Performance Computing Platforms" (2016). *Theses and Dissertations*. 3521. https://scholarsjunction.msstate.edu/td/3521

This Graduate Thesis - Open Access is brought to you for free and open access by the Theses and Dissertations at Scholars Junction. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Scholars Junction. For more information, please contact scholcomm@msstate.libanswers.com.

PGNME: A domain decomposition algorithm for distributed power system dynamic simulation on high performance computing platforms

By

Brian Shane Sullivan

# A Thesis

Submitted to the Faculty of Mississippi State University in Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical and Computer Engineering in the Department of Electrical and Computer Engineering

Mississippi State, Mississippi

August 2016

Copyright by

Brian Shane Sullivan

PGNME: A domain decomposition algorithm for distributed power system dynamic

simulation on high performance computing platforms

By

Brian Shane Sullivan

Approved:

Michael Mazzola (Major Professor)

Tomasz A Haupt (Committee Member)

Yong Fu (Committee Member)

James E Fowler (Graduate Coordinator)

Jason M Keith Dean Bagley College of Engineering Name: Brian Shane Sullivan
Date of Degree: August 12, 2016
Institution: Mississippi State University
Major Field: Electrical and Computer Engineering
Major Professor: Michael Mazzola
Title of Study: PGNME: A domain decomposition algorithm for distributed power system dynamic simulation on high performance computing platforms
Pages in Study 83
Candidate for Degree of Master of Science

Dynamic simulation of a large-scale electric power system involves solving a large number of differential algebraic equations (DAEs) every simulation time-step. With the ever-growing size and complexity of power grid, dynamic simulation becomes more and more time-consuming and computationally difficult using conventional sequential simulation techniques. This thesis presents a fully distributed approach intended for implementation on High Performance Computer (HPC) clusters. A novel, relaxationbased domain decomposition algorithm known as Parallel-General-Norton with Multipleport Equivalent (PGNME) is proposed as the core technique of a two-stage decomposition approach to divide the overall dynamic simulation problem into a set of sub problems that can be solved concurrently. While the convergence property has traditionally been a concern for relaxation-based decomposition, an estimation mechanism based on multiple-port network equivalent is adopted as the preconditioner to enhance the convergence of the proposed algorithm. The algorithm is presented in detail and validated both in terms of accuracy and capability.

#### ACKNOWLEDGEMENTS

I would like to thank Dr. Michael Mazzola, my major professor, for his guidance and help throughout my time as a master's student. I thank him for his guidance in all of the technical material documented in this thesis. I would like to thank my wife Lauren for her continuous support throughout the time of the master's program. I would also like to thank Jian Shi, Uttam Adhikari, Babak Saravi, and Tomasz Haupt for their support and the roles they played in developing the knowledge and technical expertise I have obtained during the master's program, and their direct contributions to the material presented in this thesis.

# TABLE OF CONTENTS

ACKNOWLEDGEMENTS	i	ii
LIST OF TABLES		V
LIST OF FIGURES		'i
CHAPTER		
I. INTRODUCTION A	ND BACKGROUND	1
<ul> <li>1.1 Dynamics/Tra</li> <li>1.2 Domain Decord</li> <li>1.3 Problem Form</li> <li>1.3.1 ASM/Partia</li> <li>1.3.2 Network F</li> <li>1.4 Existing Meth</li> <li>1.4.1 Direct/Sup</li> <li>1.4.2 Iterative m</li> <li>1.4.3 Define W</li> <li>1.4.4 Specific M</li> <li>1.4.5 PGN</li> <li>1.4.5.1 Derive</li> <li>1.4.5.2 Discus</li> <li>1.5 Contribution a</li> </ul>	nsient Stability	16902334670345
<ul> <li>II. ALGORITHMS FOR LINEAR NETWORK</li> <li>2.1 Algorithm Out</li> <li>2.2 Derivation of 2.2.1 Two Partitic</li> <li>2.2.2 N Partition</li> <li>2.3 Introduction o</li> <li>2.4 Numerical Ap</li> <li>2.5 Complexity A</li> <li>2.5.1 Factorizati</li> <li>2.5.2 Solving</li> <li>2.6 Discussion</li> </ul>	IMPROVING EXISTING METHODS TO THE         SOLUTION       2         line       2         W       3         ion Example       3         t Example       3         f MultiPort modeling techniques       3         proach to Analyze the W matrix       3         allysis       3         on       3         4       4	77002667880

III.	IMPLEMENTATION IN SOFTWARE	42
	3.1 Program Structure	42
	3.2 Graph generation	43
	3.3 Partitioning	44
	3.4 Simulation	45
IV.	RESULTS	49
	4.1 Large scale results on speed and scale	49
	4.1.1 Original PUR vs PGNME	50
	4.1.2 Increasing Number of Partitions	51
	4.1.3 Repartitioning the same system	54
	4.1.4 Scalability of the simulator	56
	4.2 Dynamic Simulation	58
	4.3 Matlab results on accuracy and different models	60
V.	CONCLUSION AND FUTURE WORKS	62
REFEI	RENCES	64
APPEN	NDIX	
A.	W MATRIX DERIVATIONS	67
	A.1 Two Partition Example	68
	A.2 Three Partition Example	70
	A.3 Explicit W Validation of the Three Partition Example	81

# LIST OF TABLES

4.1	Comparing PUR and PGNME performance	50
A.1	Parameters for W matrix verification	82

# LIST OF FIGURES

1.1	Gauss Seidel and Jacobi iteration	7
1.2	Four bus example decomposed into two subgraphs each terminated by a dummy bus(d1 and )	9
1.3	ASM solution procedure	11
1.4	Iterative method algorithm	15
1.5	PGN original system	20
1.6	PGN updating strategy	21
1.7	PGN Jacobi based updating strategy	22
1.8	PGN Gauss-Seidel based updating strategy	23
1.9	Schematic representation of PGN	24
2.1	ASM approach combined with PGNME	28
2.2	Three Partition System	33
2.3	Decomposed system with added dummy buses	34
2.4	Relaxed equivalent representation of a 3 partition system	35
3.1	Original IEEE 118 graph	43
3.2	Level two, three times scale of original IEEE 118 system	44
3.3	Gephi Partitioned graph visualization	45
3.4	Simulation Flowchart	46
3.5	Intel MPI Trace Analyzer	47
3.6	Zoom in Trace Analyzer Simulation	48
4.1	Speedup of 1062 bus system	52

4.2	Speedup curve of 258,066 bus system	53
4.3	Effect of increasing partitions	54
4.4	Repartitioning the 3186 bus system	55
4.5	Cumulative distribution of iterations from partitioning	56
4.6	Scalability of the simulator	58
4.7	Speedup curve using classical generator models in the Eastern Interconnect scale system	59
4.8	Generator speed response during an event	60
4.9	Bus Voltage response during an event	61
A.1	Two Partition Equivalent Description	68
A.2	Subcircuit 1 and 2	70
A.3	Subcircuit 3 and 4	71
A.4	1-2-1 System Verifying Explicit W	81

## CHAPTER I

## INTRODUCTION AND BACKGROUND

#### 1.1 Dynamics/Transient Stability

Dynamic simulation has been a huge area of research focus in electrical engineering for a very long time, stability was first noted as a problem for the power system in 1920[Kundur]. In the early 1950s initial simulations were performed on computers and since then the effort has been nonstop to improve the performance [Kundur]. There has always been interest in obtaining faster, more accurate simulation results. In recent trends the interest is focused on obtaining very fast results on large scale systems. In the interest of pursuing the ability for wide area monitoring and control, maintaining stability in the system, and better forecasting demands, simulations of extremely large system models such as the Eastern interconnect or the entire interconnected North American power grid need to be performed, in which detailed models could contain hundreds of thousands of buses. The complexity of the system is also rapidly growing with the introduction of renewables and smart grid technologies.

Transient stability simulation is an integral part of power system simulation studies as it obtains the necessary information to capture relevant system operating conditions, determine the dynamic system response subject to disturbances or damages, and identify the corrective actions. It is critical for various types of power system analysis such as Dynamic contingency analysis, look-ahead dynamic simulation, dynamic state estimation, online stability analysis, etc. Solving the interactive electrical and mechanical dynamics, especially for a large-scale power grid, involves setting up and solving thousands or tens of thousands of system equations in the time-domain which is typically time consuming and computationally intensive. Accurate assessment of the system dynamic behaviors of interest, without excessive computational overhead, has become a serious concern and challenge for practical application of electrical power system design, analysis, optimization and control.

Computer simulation of power systems has been long developed and serial techniques have been the focus of the simulations, but as serial processing has reached its limitations, the need for parallel and distributed processing for future development has been noticed, and advances have been made in multi-processor multi-threaded applications and architectures. Realistically, for this ability to simulate such large systems in real time or faster than real time, there has to be a move onto a suitable architecture. In recent attempts by national labs, large shared memory machines have been used to perform dynamic simulation, but they are limited in scalability and are very costly [PNNL]. Distributed memory architectures scale much better and come at a lower cost but have an added cost of inter node communication. Parallel and distributed simulation is a big area of research interest especially in the power system area. Many fields of science and engineering have a need to accomplish parallel or distributed simulation. Most existing simulators cannot perform in real time on large systems. There is a large effort to achieve real time simulation for large scale power systems, and this will only be possible in a parallel/distributed computing environment [DOE].

Most simulation tools that can work in real time are limited to smaller system models or to extremely expensive, customized, shared memory machines. Some examples come from Opal-RT [OpalRT], RTDS [RTDS], and the Pacific Northwest National Laboratory (PNNL) [PNNL]. Opal-RT and RTDS have power system specific programs and create specialized shared memory hardware for customers, but these simulations are limited in terms of how large system models can be. PNNL has done extensive work towards the goal of real time simulation on very large, powerful, shared memory general use machines. These machines are expensive compared to distributed memory machines, and they also do not scale as well. In the future, as the grid continues to grow, the shared memory machine will not be adequate.

There are also other applications of parallel/distributed simulation to the power system (i.e. State Estimation, Optimal Power Flow, Security Constrained Unit Commitment (SCUC), and Security Constrained Economic Dispatch (SCED)). Specifically, this thesis focuses on dynamic simulation, in particular, the solution of the network equations and how this calculation can be separated to obtain faster overall performance. In order to perform dynamic simulation in a distributed fashion, one must partition the system into subsystems, find some way to represent the rest of the system at the cut of the partition, and iteratively solve for the system states at each time step.

Some parallel simulations use data parallel methods to gain speedup. Data parallel solution methods are straight forward and do not need special considerations for coupling. Task oriented parallel simulation requires a coupling method. Where and how a system is decoupled greatly affects the parallel runtime and performance [QHuang]. If a decoupling point is not chosen well for partitioning, the algorithm could take a very long

time to converge or may not converge at all. Speed and accuracy are desired from simulations but usually some middle ground must be found.

Papers exist on ways to couple subsystems for different purposes. Some methods are faster, some more accurate, some are too unstable. In general there are three types of methods, V, I, and I-V coupling. An I-V variant is used in [JWu] for distributed simulation. In [WRen] various algorithms are compared in Power Hardware-in-the-Loop simulation which requires decoupling for simulation.

The existing papers present different methods, in [Russian] there is an attempt to quantitatively analyze the methods presented, but the paper does not provide a well detailed comparison. The results in [Russian] show runtime of simulations using different coupling methods, but only the runtimes and a very brief analysis are presented. In [WRen] the methods are analyzed in greater detail, but the methods are applied to hardware in the loop simulation. As well as not giving detail in the analysis, the typical paper that discusses the coupling mechanism does not focus solely on the coupling mechanism, but instead are focused on the simulation implementation.

Existing commercial software does not aim at the scale that this thesis will demonstrate, the existing tools are not intended for wide area monitoring type studies and can only handle local studies or simplified models of large areas. Many simulators do not attempt to test the limits of their software, and they do not promote scalability. Very few tools give limitations in number of buses or models. The existing tools, where limitations of system sizes are given, are: PSSE, PowerWorld, and PSLF. Some academic creations exhibit large test systems. Among the documented test systems, the largest available are from PNNL [PNNL], OpalRT [VOparlRT], and in [Chinese2].

Commercial software limits are very high. PSLF limitation is 60,000 buses. PSSE is limited to 150,000 buses and was the leader in size until PowerWorld came about. PowerWorld claims a limitation of 250,000 buses, and that is the largest available for commercial software. The existing research oriented simulators do not boast limitations but the test system sizes are reported. At PNNL in [PNNL] a 16,000 bus model is used on a shared memory machine and real time simulation is performed. At OpalRT [VOpalRT] a 9984 bus model is used but cannot be simulated in real time on their hardware, but a 7020 bus system is used and simulated in real time. In [Chinese2] an 18,000 bus system is used. These are all pure power system simulators.

Recently hybrid simulation has become a new trend by simulating a discrete communication system along with the continuous dynamic power system. The communication system often being the SCADA system or other sensor networks. These are typically much larger simulations because of the detailed models. In [PNNL2] a 2,063,494 node model is used, however the number of physical components vs number of communication components are not clear. This model includes many elements such as substation models, battery storage models, wind generation, solar generation, and distributed generation models.

The hybrid simulation tools are often used to study the effects of the interactions between controls, communications, and the continuous electro-mechanical dynamics of the power system. [THYME] is a hybrid simulation toolkit which contains simple machine models with simple controls, and a framework for modeling discrete sensors. It is a fairly well developed library, but it is poorly documented. It has built in line loss, load loss, and generator loss methods.

# **1.2** Domain Decomposition

The electric power industry has traditionally been a computer oriented industry. As a large-scale, geographically distributed system, the design and analysis of power system is computationally-intensive. Especially with the increasing system size and complexity, the ever-growing amount of data, and the greater need for resilience enhancement to handle fast dynamic phenomena that could lead to cascading system failures and blackouts, the transition to a smart future electric power system inevitably involves large scale computation, modeling, and data handling. However, the current power system design, analysis and software tools are developed heavily based on single processor architecture. The power system community is aware of the need to use the latest advances in computation techniques, and to develop visionary approaches to reevaluate the legacy power system analysis methods and shift from the traditional off-line, steady-state based analysis to a faster, on-line, dynamic and robust platform [DOE]. To achieve this objective, a key design element is the concept of domain decomposition (DD).

DD methods were originally proposed as a numerical analysis approach to solve a problem (esp. boundary value problems) defined over a domain by decomposing the original problem into smaller problems on sub-domains and coordinating the solution among subdomains normally iteratively to derive a globally converged result [DKeyes] [VDolean]. The DD method is inherently suitable and adaptable for parallel computing architectures as it can effectively handle the type of problem that does not fit into available memory space, and by splitting the original problem into sub-problems, concurrency and parallelism can be gained. The guiding principle of applying DD on

power system analysis is to develop a DD algorithm specifically tailored in accordance to the unique physical properties of power systems and the state-of-art parallel computing architecture to realize the overall enhanced large-scale parallelism. Although a general DD technique can be applied in various aspects of power system analysis, this thesis will particularly focus on the application of parallel dynamic simulation.

The DD techniques presented in this thesis focus on decomposition of a system with linear behavior. There are two ways to solve the problem using these methods, serial or parallel. The serial version is based on the Gauss-Seidel method, and the parallel version based on the Jacobi method. The two versions are depicted in Figure 1.1.



Figure 1.1 Gauss Seidel and Jacobi iteration

It is always true that the Jacobi method requires more iterations than the Gauss Seidel method. The way an iteration is defined for the methods is fundamentally different. In the Gauss-Seidel method, each subsystem is solved in sequential order, the updated states from each solution are used during the current iteration. One iteration involves solving each system one time, but using the advantage of using the most updated states from the other systems. In the Jacobi method, each subsystem is solved simultaneously in parallel, and then the information is used only on the next iteration. While Jacobi method requires more iterations the fundamental definition of an iteration is the root, and the Jacobi method is usually faster if all of the sub problems can be run concurrently.

With these methods the concept of a boundary bus must be introduced, also known as a dummy bus. With all of these iterative methods, there is a topological description of the system and the method. When the original system is decomposed n subsystems remain. To obtain a self-consistent solution, the rest of the system must influence each sub problem. This can be done with a variety of types of terminations, and they can be mixed or symmetrical as well. These terminations are what are called dummy buses. It can be shown in Figure 1.2, if one four bus system is decomposed into two sub systems, then there must be an added bus to represent the behavior of the rest of the system.





Figure 1.2 Four bus example decomposed into two subgraphs each terminated by a dummy bus(d1 and )

This dummy bus is described schematically different for the different methods and it is updated in a unique way corresponding to the iterative method.

# **1.3 Problem Formulation**

In power system dynamic simulation the problem is formed in different ways depending on the study. In general, the continuous power system dynamics can be mathematically described by a set of first-order Differential Algebraic Equations (DAEs) in the form of:

$$\dot{x} = f(x, y) \tag{1.1}$$

$$0 = g(x, y) \tag{1.2}$$

where x represents a vector of dynamic state variables, such as rotating machine variables, and y represents a vector of the algebraic state variables, such as network variables where no derivatives are present.

Once the DAEs are constructed, two approaches can be applied to solve the DAE: either the Alternating Solution Method (ASM) or the Direct Solution Method (DSM) [CFu]. With ASM, the new set of algebraic equations from discretization of the ODEs in Equation 1.1 are solved separately from the algebraic network equations within the DAE in Equation 1.2. In contrast, with DSM, all of the algebraic equations representing the dynamic components and the network are solved simultaneously [BStott] [HDommel]. It has been pointed out in literature that the ASM approach, combined with the explicit integration method, has various advantages over the DSM approach mainly in that the components dynamics are solved separately from the linear algebraic network equations [CFu]. Thus in the network equations, the dynamic devices such as generators are represented as constant current sources, and their admittances can be combined and considered within the network admittance matrix. By doing so, the system can be solved efficiently with the use of a sparse linear algebraic solver such as SuperLU, UMFpack, or PETsc.

It is known that the network calculations become a limiting factor at large scale [JWu]. For this reason it is best to use ASM and use Domain Decomposition to solve the problem of dynamic simulation at a large scale. This means all machine dynamics will be solved separately from the network equations. The models of the generators and loads can be replaced with more complicated or more simplified models depending on the study. A common approach is to use detailed models near events under study and to use simplified models for equipment far away from the event.

#### **1.3.1** ASM/Partitioned Approach

In phasor based simulation, the power system formulation is that of Equation 1.1 and Equation 1.2, which consist of sets of differential algebraic equations that are coupled to each other through a set of linear algebraic network equations. While there are different ways to solve this set of DAE's, a common approach is to use the partitioned approach, also known as Alternating Solution Method (ASM), that is to iteratively solve the differential equations and solve the linear algebraic equations separately. There are other methods such as the simultaneous solution approach where the differential equations are discretized into a set of algebraic equations and these equations are lumped together with the network equations to be solved simultaneously.



Figure 1.3 ASM solution procedure

As shown in Figure 1.3 the ASM approach applies as a general treatment to DAEs. The stages shown in the algorithm represent stages for multi stage numerical integration techniques such as Runge Kutta methods or other multi stage methods.  $\Delta t$  can

be fixed or variable depending on the numerical integration technique. Variable time step methods are usually preferred as they try to optimize the speed of the simulation with a given error tolerance, so if a large time step can be used with acceptable error the variable step method will select the largest possible time step, but when an event such as a fault occurs and the error would be large with a large time step, the variable rate methods will reset the time step to a short enough value to maintain the error within a bound.

#### **1.3.2** Network Formulation

In the power system network formulation traditional nodal analysis is used. Based on the power system description the nodal analysis can be simplified as the system can be represented as a mesh network of impedances with current injections at each node. Lines are modeled as an impedance between nodes. Loads, generators, and shunt compensations are modeled as a shunt impedance at the node. This model leads to the formulation of the power system admittance matrix. The admittance matrix formulation can be shown as in Equation 1.3.

$$\begin{bmatrix} I_1 \\ \vdots \\ I_n \end{bmatrix} = \begin{bmatrix} Y_{11} & \cdots & Y_{1n} \\ \vdots & \ddots & \vdots \\ Y_{n1} & \cdots & Y_{nn} \end{bmatrix} \begin{bmatrix} V_1 \\ \vdots \\ V_n \end{bmatrix}$$
(1.3)

$$Y_{ii} = \sum_{1}^{j} Y_{ij} \tag{1.4}$$

$$Y_{ij} = -Y_{ij} \tag{1.5}$$

The bus injection current is calculated from the changed dynamic state variables. The generator admittance is hard to determine for large scale systems and is often predicted rather than known [Thyme]. Loads are often modeled as constant impedance and the admittance value is calculated from the initial bus voltage and power from power flow information. If any topological changes happen, the admittance matrix must be modified. The diagonal elements of the admittance matrix are the sum of all admittances connected to that node as in Equation 1.4, and off diagonal elements are the negative of the mutual admittance between the two nodes as in Equation 1.5.

#### 1.4 Existing Method for Linear Network Solution

There are two distinct classes of methods to solve the large set of linear equations that form the network equation, Direct and Indirect methods. Direct methods solve directly either by factorization or direct inversion of the Y matrix. There are parallel and serial direct methods, which all have a long history of development. Indirect methods usually are iterative and do not operate directly on the original matrix. Iterative methods, in general, have a long history of development in the mathematics community. The iterative methods are based on relaxation theory.

## 1.4.1 Direct/SuperLU

Popular direct methods include Gaussian elimination, Cholesky decomposition, and LU factorization. As far as direct methods for large sparse matrices the leader in performance is LU factorization and many commercial tools exist to solve the problem. Most leading simulation software uses LU factorization to solve their linear matrix calculations. SuperLU is a well-developed library created in 1999 for a PhD thesis at University of California Berkeley [SUPERLU] add reference. SuperLU now receives federal support from the DoE, NSF, and DARPA. The software is maintained by its creator at Lawrence Berkley National Lab.

There are several versions of SuperLU that work on different computer architectures. The original version is sequential SuperLU used with sequential computing. SuperLU\_MT is optimized for multithreaded environments for newer architectures with multiple cores. SuperLU\_DIST is optimized to run on distributed memory machines such as general purpose cluster computers.

The main attraction to SuperLU and LU factorization is that non-zero entries can be indexed and efficiently stored. Many other commercial software tools exist, both old and new such as Portable, Extensible Toolkit for Scientific Computation (PETSc) which is somewhat harder to use, but has built in capability for LU factorization, ILU factorization, SOR, Jacobi methods, and other methods. PETSc is a newer tool, older tools exist such as Linear Algebra Package (LAPACK) which is a considered a standard software library and was initially released in 1992.

#### **1.4.2** Iterative methods

Popular iterative methods include Richardson's method, Jacobi method, Gauss-Siedel method, successive over-relaxation (SOR), and symmetric successive overrelaxation (SSOR). Many relaxation based methods have been presented in literature. Some relaxation techniques show huge promise although some exhibit poor performance. Some relaxation techniques have been shown to be unstable and inaccurate [WRen] [Russian]. With many iterative techniques, performance directly depends on the quality of partitioning. Coarseness of the partitioning affects all methods and some methods need special attention to certain parameters. For all methods considered in this thesis, if tightly coupled vertices are not lumped together then the algorithm may converge very slowly, on the contrary if too many vertices are lumped together the benefits of using these

methods is lost[Russian]. From this knowledge, it is easy to see that some systems will not benefit from these methods especially systems which are tightly coupled.

The idea of iterative methods applied to solving circuit problems is generic and can easily be shown algorithmically at a high level. The details will change depending on the method used.



Figure 1.4 Iterative method algorithm

The detailed algorithms for each method vary. But in general the methods are used to solve an equation in the form of

$$Ax = b \tag{1.6}$$

where **A** is an  $n \times n$  nonsingular matrix and **x** and **b** are  $n \times 1$  column vectors.

#### 1.4.3 Define W

In iterative methods general form, all state variables are a function of other state variables at the previous iteration. It is defined that

$$x^{i+1} = Wx^i + C \tag{1.7}$$

where W is known as the iteration matrix and C is a column vector of constants. To guarantee convergence for any iterative method, the spectral radius of W must be less than one [LHageman]. That is

$$\rho(W) = \lim_{k \to \infty} \left| W^k \right|^{\frac{1}{k}} \tag{1.8}$$

(1.8) can also be defined as

$$\rho(W) = \max\{|\lambda_1|, |\lambda_2|, \dots, |\lambda_n|\}$$
(1.9)

The methods considered in this thesis are considered linear stationary iterative methods of first degree. First degree meaning that  $x^{i+1}$  depends only on  $x^i$  and no other previous states. Linear meaning that **W** and **C** are not dependent on **x**, and stationary meaning that **W** and **C** are also not dependent on the iteration count. Many iterative method books refer to these methods as basic iterative methods [LHageman] [KELLEY] [SAAD].

It is interesting to note that the smaller the spectral radius, the smaller the iteration count. While there are ways to predict the performance, there is no way to predict the exact number of iterations.

Formulation exist to predict convergence rate in [Hageman]

$$R_{\infty}(W) = \lim_{n \to \infty} R_n(G) = -\log \rho(W)$$
(1.10)

where  $R_{\infty}(W)$  is the asymptotic rate of convergence, and where  $R_n(W)$  is defined as the average rate of convergence and can be defined as

$$R_n(W) = -n^{-1} \log \|W^n\|_{\beta}$$
(1.11)

Convergence rate can be predicted, but if and only if the basic criteria is satisfied for convergence, that is  $\rho(W) < 1$ . If this condition is satisfied, then an approximation can be found to reduce the norm of the initial error vector by a factor of  $\zeta$ 

$$n \cong -(\log \zeta)/R_{\infty}(W) \tag{1.12}$$

However, using the average rate of convergence in this formula leads to a more accurate estimation, but is often not obtainable [Hageman].

#### 1.4.4 Specific Methods

[Russian] presents three generic iterative linear methods for boundary value reconciliation: V, I, V-I. These techniques were first implemented in circuit simulators in early seventies [Russian]. Relaxation techniques such as these still remain promising areas of research. Zdorov points out that there are some instances where direct methods cannot be used, such as when there is a need for parallel computation and simulation time is critical. He also points out that these methods give flexibility in the model descriptions, and that these methods allow models to scale more efficiently when memory requirements cannot be met with the unpartitioned system.

According to all of these derivations, larger line impedance between subsystems leads to smaller spectral radii. This is very important to acknowledge during partitioning. The impedance information can be used to improve the convergence rate. According to the V derivation smaller  $Z_a$ ,  $Z_b$  lead to a smaller spectral radius. Using other methodspecific information for partitioning can be extremely beneficial to the performance of these methods.

These methods are interesting, but they were not derived with the power system in mind. Out of these methods, the V method is the most stable and is used often in VLSI studies. Zdorov concludes that of these methods I-V is the best for a general approach as it can be modified to behave similar to the V or I method. These approaches look good at first glance, but the need for asymmetrical boundary adds complexity. At a generic scale, the V method seems to be more suitable since it is stable and symmetric. The VI method would require a complex evaluation of the partitioning to determine what the value of  $Z^*$  should be, and the convergence would still rely on the slowest converging component of the partitioned system which could void all benefit of using the VI method. It seems that V is the only scalable solution of these methods. The partitioning of the system directly determines the performance. The downside of all of these methods is that they do not fit into the traditional power system network description such as that of Equation 1.3. If they are to be used in an existing simulator, the solver must now take the more generic form of Equation 1.6.

In [WRen] Ren presents several existing methods. The methods he presents are: Ideal Transformer Model (ITM), Time-variant First-order Approximation (TFA), Transmission Line Model (TLM), Partial Circuit Duplication (PCD), and Damping Impedance Method (DIM). These methods are used in Hardware in the Loop (HIL) simulation, but they can also be used in a pure software implementation. In fact, Ren presents the PCD method which is the V method as presented by Dmitriev-Zdorov in [Russian]. He also presents the DIM method which looks like the I-V method presented by Dmitriev-Zdorov. The ITM method looks similar to the I method presented by Zdorov.

According to Ren, the TFA method is too complex and limited. He says special care must be taken during solving as the matrix can easily become ill conditioned. He also claims it is unstable and inaccurate. Ren also claims that the TLM algorithm is highly stable, along with the PCD method.

Overall, these methods provide multiple ways to solve the network. They still lack the symmetry that is necessary for scaled simulation. TLM and PCD are both symmetric, but the remainder are not. The recommendations provided by Ren are not intended for simulation, but for hardware in the loop simulation. For scalable simulations, it seems that TLM and PCD are both plausible options with PCD being the better choice. PCD is the most attractive method of all of the methods presented by Ren. This is interesting as it is also the most attractive method presented by Zdorov in his comparison. Because his paper is focused on Hardware in the Loop simulation, he does not give good analysis or insight into the scalability of the algorithms for simulation on high-performance computers

All of these methods are based on solid relaxation theory and have solid backgrounds. But they were not intended for power system simulation, where the unique form of the network allows a unique way to mathematically model the network, and this model does not allow any vertex to be represented by anything other than a Norton Equivalent lumped model.

# 1.4.5 PGN

PGN is a method documented by Jian Wu and Noel Schulz and shows promise applied to power system analysis from existing literature [JWu]. It is a specifically useful form for the power system dynamic problem, as it takes advantage of the fact that the power system network formulation consists of Norton Equivalent circuits linked together through transmission lines. If using an existing software which is developed specifically for power system simulation then it is likely difficult to change the network formulation which takes the form of Equation 1.3. This form strictly depends on each vertex being represented as a Norton equivalent. If the form changes the solver will need to be changed accordingly into a more general form.

The general system description of PGN can be depicted as two portions of the transmission system tied together by one linking transmission line, such as that of Figure 1.5. Existing literature only addresses a two partition example and does not show a method to scale to more partitions.



Figure 1.5 PGN original system



Figure 1.6 PGN updating strategy

The partitioned version is shown in the schematic of Figure 1.6. As shown, the dummy circuit represents the other portion of the system, and it is controlled by the other system during the iterative updating strategy. Two strategies can be used, the Gauss-Seidel or Jacobi strategy as presented in Figure 1.1. The details of the procedure are shown in Figure 1.7 and Figure 1.8.



Figure 1.7 PGN Jacobi based updating strategy



Figure 1.8 PGN Gauss-Seidel based updating strategy

# 1.4.5.1 Derive W

The iterative matrix for the PGN method can be obtained for any system but in studies presented by Schulz and Wu [JWu] there are only studies on two partition systems. For the two partition system, it can be shown as an equivalent reduced system such as that of Figure 1.9.



Figure 1.9 Schematic representation of PGN

For this simplified system **W** can easily be obtained by for both the Gauss Seidel version and Jacobi version. The resulting spectral radii are that of Equation 1.13 and Equation 1.14.

$$\rho(W) = \frac{(Z_{d_1} - Z_a)(Z_{d_2} - Z_b)}{(Z_{d_2} + Z_a + Z_{ab})(Z_{d_1} + Z_{ab} + Z_b)}$$
(1.13)

$$\rho(W) = \sqrt{\frac{(Z_{d1} - Z_a)(Z_{d2} - Z_b)}{(Z_{d2} + Z_a + Z_{ab})(Z_{d1} + Z_{ab} + Z_b)}}$$
(1.14)

It is very important to note that when  $\rho(W) \approx 0$ , convergence is very fast making the iteration count less and the gain from parallelization more. In this original work by Wu it is easy to see for the two partition system a dummy bus parameter can be selected to make  $\rho(W) \approx 0$  by setting  $Z_{d1} = Z_a$  and/or setting  $Z_{d2} = Z_b$ . For larger systems with many partitions, the PGN equations do not represent the equivalent at each cut line. Meaning the only indication from these equations that can be used to partition is finding large  $Z_{ab}$  which will not lead to optimal performance with most systems.

# 1.4.5.2 Discussion

This method seems very attractive for use in power system distributed simulation. It uses a Norton Equivalent representation and was created with power system simulation in mind. It seems better for power system analysis use than all of the methods presented by Ren and Zdorov. However, this method has not been used in large studies and has only been shown to be used in two partition examples. The limitations at scale need to be resolved. There are limitations in using this method at scale because of the iterative process. According to Equation 1.9, spectral radius depends on the largest eigenvalue of the iteration matrix. When moving to more partitions, the size of **W** grows and there is no way to set the dummy bus with current literature to improve the convergence rate.

Looking forward to future size power systems, where detailed models are required to accurately predict the behavior of the system, convergence rate must be controlled. Future simulation will need to consider solar models, wind turbines, distributed generation, sensor networks and controllers, etc. With the existing literature, it seems that scalability of large simulations on HPC with PGN is limited. This thesis will provide a novel contribution to enhance the performance of PGN when applied to multiple partition examples.

In other papers produced by Wu and Schulz, they discuss different options of simulation. They also present an algorithm to update the dummy node impedance, but this algorithm is not numerically stable as it uses a difference in state variables in the denominator of the updating equations. This causes instability when the states become close to their final converged value.

### **1.5** Contribution and organization of this thesis

The main contribution presented in this thesis will focus on greatly improving the method presented by J.Wu and N.Schulz to be suitable for large scale simulation by introducing a novel approach to enhance its performance which shall be called Parallel
General Norton with Multiport Equivalent (PGNME). There is also a smaller contribution of creating scalable testing systems.

Chapter 1 has given adequate introductory and background information to follow the remainder of this thesis. Chapter 2 will provide detail of the contributions presented in this work, namely the PGNME method. Chapter 3 will provide specifics of the implementation used to obtain the results in Chapter 4. Chapter 4 will present simulation results using the solver detailed in Chapter 3 and provide a comprehensive evaluation of the PGNME method presented in Chapter 3. Chapter 5 will conclude the thesis.

#### CHAPTER II

# ALGORITHMS FOR IMPROVING EXISTING METHODS TO THE LINEAR NETWORK SOLUTION

#### 2.1 Algorithm Outline

From Chapter 1, it is shown that a common approach to dynamic simulation is the phasor based ASM approach. To reiterate, ASM separates the DAE posed in Equation 1.1 and Equation 1.2 into three distinct subsets for the power system problem. Subset 1 contains the first order differential equations which model the dynamic behavior of generators and loads. Subset 2 contains all algebraic equations which model the behavior of generators and loads. Subset 3 is the linear algebraic network equation. This can be illustrated in Figure 2.1. Subset 1 and 2 are combined but solved completely independent of each other in the general case. Each generator and load is connected through subset 3. With large systems, subset 1 and 2 can easily be distributed to CPU's to gain speedup using the ASM strategy, but as the system grows subset 3 becomes a burden and needs to be decomposed for efficient parallel/distributed implementation, this is done by using PGN and is improved using the multiport equivalent information. The resulting method is PGNME, which is described in this chapter.



Figure 2.1 ASM approach combined with PGNME

Two approaches exist to solve the sparse linear set of equations constituting the network, iterative and direct. From existing literature, it seems common for iterative methods to be explored, and in this thesis, the particular iterative method is Parallel Updated Relaxation (PUR). This allows for a graph based / topology based decomposition which is natural and straightforward. The PUR method has not been explored on large systems or in many partitions. In the implementation results shown in this thesis, it can be seen that this method performs worse when more partitions are created making this method infeasible to scale. However, with appropriate selection of the boundary dummy impedances, the impedance can be set such that iteration is minimized.

To do this, the details of iterative methods and relaxation theory must be examined. The PGNME algorithm uses a relaxation based approach to relax the network solution allowing for parallel solving of subsystems. The relaxation based method holds the form of general linear iterative methods. After examining exactly what is happening, MultiPort modeling techniques can be used, which are well known to model the subsystems and approximately satisfy the conditions necessary for optimal performance of this method, with certain limitations.

While the PUR method is one of the best methods available in applying the ASM relaxation based technique from existing literature, it still has its flaws. With a parallel architecture, iteration adds not only more computation, it also adds inter processor communication. High iteration counts are unacceptable and will significantly reduce the gains obtained from the parallelization. Without a decent initial guess at the boundary state variable and without properly selecting the boundary equivalent impedance, the method may not give much gain in performance, and in fact may make performance worse under certain conditions.

After partitioning the original system many partitions remain which require a boundary equivalent that represents the remaining portion of the system that has been removed. Conceptually, it is logical that if this boundary equivalent accurately represents the missing system then faster or more accurate results can be obtained. Since existing literature does not explain what happens to performance at larger scale, this thesis will attempt to add to the existing literature and give insight on large scale system performance.

Before understanding the method, it is important to understand exactly what is happening when the method is introduced to more than two partitions. It is also important to understand the background of multiport modeling techniques.

29

#### 2.2 Derivation of W

To derive the **W** matrix on a scaled system it is important to first define a generic way to obtain the **W** matrix. This section will derive the **W** matrix in detail of a two partition example and show how the derivation extends to multiple partitions. First, each system must be represented as an equivalent. For the two partition example, this is straightforward, calculate the Norton Equivalent for each system. When more partitions are involved, multiport modeling techniques are used to derive the system equivalents. When this process is applied to the two partition single line cut example, the resulting process is known as PGN.

#### 2.2.1 Two Partition Example

First, it is important to see that the W matrix, as traditionally described, is not a diagonal matrix for the PGN method, so using the traditional derivation leads to complicated calculations of the eigenvalues of a non-diagonal matrix. This generic derivation will instead find a diagonal form for an equivalent of the W matrix. The generic iterative form is shown in Equation 1.7, and it can be shown further into the derivation that a completely diagonal form for the two partition example if  $W^2$  is found. Using Equation 1.7 a simple derivation of  $W^2$  can be found.

$$x^{i+2} = Wx^{i+1} + C (2.1)$$

$$x^{i+2} = W(Wx^{i} + C) + C$$
(2.2)

$$x^{i+2} = W^2 x^i + (W+I)C$$
(2.3)

This representation still holds valid in the iterative form as a future state depends on the update of the current state. The difference is that in the generic derivation,  $W^2$  is in diagonal form and the eigenvalues can be observed more easily. The eigenvalues of a diagonal matrix are the diagonal entries, so to find the original W matrix, each eigenvalue of W is the square root of the eigenvalues of  $W^2$ . What an iteration is can be defined arbitrarily in different ways. Since this method is Jacobi inspired, iterations should be counted in a similar way to the Jacobi method.

For this specific problem, it is necessary to show the reduced schematic after finding the Norton Equivalent of each partition. This can be shown in Figure A.1 in the appendix. The derivation of the explicit **W** matrix is shown in Appendix section A.1.

Now to have an equation in the form of Equation 2.3, the explicit **W** matrix in the appendix is squared, and will take the form of Equation 2.4 with eigenvalues in the form of Equation 2.5. For this simple case, the  $W^2$  matrix is completely diagonal, and the eigenvalues are a repeated real set. The eigenvalues of **W** can then directly be found by taking the square root of the diagonal elements of the  $W^2$  matrix.

$$\begin{bmatrix} I_a(i+2)\\I_b(i+2)\end{bmatrix} = \begin{bmatrix} \lambda_1^2 & 0\\0 & \lambda_2^2 \end{bmatrix} \begin{bmatrix} I_a(i)\\I_b(i)\end{bmatrix} + \begin{bmatrix} C_1\\C_2\end{bmatrix}$$
(2.4)

$$\lambda_1^2 = \lambda_2^2 = \frac{(\frac{1}{Y_1} - \frac{1}{Y_b})(\frac{1}{Y_2} - \frac{1}{Y_a})}{(\frac{1}{Y_{ab}} + \frac{1}{Y_a} + \frac{1}{Y_1})(\frac{1}{Y_{ab}} + \frac{1}{Y_b} + \frac{1}{Y_2})}$$
(2.5)

This explicitly shows that the value of the spectral radius can be controlled directly by setting the dummy bus impedance equal to the neighboring systems Norton equivalent impedance. If set exactly, then the spectral radius is zero. From the literature this means that this will induce the best possible performance that the particular system can have, making the Rate of Convergence ideal.

#### 2.2.2 N Partition Example

This method poses a challenge for more than two partitions. For the simple two partition example, the Norton equivalent is easy and straightforward to find. When moving to multiple partitions, multiport modeling techniques will extend the generic approach to multiple partitions.

In an N partition example with M ports, the W matrix, which has more dependencies and can be extremely hard and time consuming to find, must be found. Therefore, an estimate of the W matrix can be formed called  $W_E$ . This estimation is based on the idea of relaxing the inter port dependencies. If the system remains relatively small, then an explicit form of W could be obtained by solving symbolic linear equations to find the roots of the eigenvalues of the W matrix with respect to the dummy bus impedances. By setting the dummy impedance correctly to make the eigenvalues of W explicitly equal to zero, one can minimize the iteration to theoretically instantaneous.

For large systems with many partitions, this calculation could be extremely cumbersome and computationally expensive. Since the W matrix changes when the topology changes, there needs to be a way to quickly reset the boundary admittance in case of a network change which degrades performance drastically. Using an estimated W by relaxing other port dependencies, an explicit fully diagonal form of W can be obtained.  $W_E$  is generic in terms that the eigenvalues will repeat in a similar form which is equal to the form of the two port example in Equation 2.5. While this method is much faster and more efficient than finding an explicit form of the W matrix, it suffers from some inaccuracies under certain partitioning. However, even in the worst cases, there is still improvement over PUR without multiport equivalent information. Partitioning optimization should be considered to keep ports distanced from one another to prevent this performance flaw.



Figure 2.2 Three Partition System

For the derivation of the **W** matrix, the system in Figure 2.2 is used. This system is decomposed into three subsystems by cutting two transmission lines. The new schematic is represented in Figure 2.3 with the boundary buses added.



Figure 2.3 Decomposed system with added dummy buses

At this point, either the explicit  $\mathbf{W}$  can be calculated, which is a complex calculation, and is shown in Appendix section A.2, or  $W_E$  can be calculated and is shown in Equation 2.6, the coefficients approximate the appendix coefficients to within a residue. In this work,  $W_E$  is considered to be more advantageous because of its ease of calculation and scalability. Ignoring the residual component allows PGNME to scale with limitations. Those limitations can be addressed in a variety of ways. To calculate the  $W_E$  matrix, the port dependencies are ignored. When calculating the dependencies, each connection can be viewed as a two port network as shown in Figure 2.4, where the other ports are held constant and do not influence the updating process. In the  $W_E$  matrix, 2 sets of eigenvalues are obtained, the eigenvalues take the general form of Equation 2.5. In the generic case, this approach creates a purely block diagonal form of the W matrix, this block diagonal form creates blocks which equal the two partition W matrix, the eigenvalues can be found simply by taking the square root of the  $W_E^2$  matrix.



Figure 2.4 Relaxed equivalent representation of a 3 partition system

$$W_{E} = \begin{bmatrix} 0 & \alpha & 0 & 0 \\ \gamma & 0 & 0 & 0 \\ 0 & 0 & 0 & \eta \\ 0 & 0 & \kappa & 0 \end{bmatrix}$$
(2.6)  
$$W_{E}^{2} = \begin{bmatrix} \lambda_{1}^{2} & 0 & 0 & 0 \\ 0 & \lambda_{2}^{2} & 0 & 0 \\ 0 & 0 & \lambda_{3}^{2} & 0 \\ 0 & 0 & 0 & \lambda_{4}^{2} \end{bmatrix}$$
(2.7)

With this derivation, one can see that the entries in the explicit W matrix consist of difference equations and residual equations. The difference equations are set to zero by setting the dummy bus admittance parameters equal to the multiport equivalent. The remaining residuals are directly dependent on the coupling between multiple systems. It so happens that for the case in the appendix, if all dummy bus admittance properties are set correctly, then all eigenvalues are zero no matter the magnitude of the residual. But this special case will not hold with even one higher degree of generality, this it is possible to hypothesize that in more complex cases the residue will play a role in the eigenvalues at some point, especially if a partitioning strategy leaves no single port partitions.

#### 2.3 Introduction of MultiPort modeling techniques

These subsystem parameters must be derived before this method can be beneficial. To do this, the existing multiport modeling techniques are used. It can be shown that each partition can be modeled with a multiport equivalent reducing the overall **Y** matrix of the partition into a smaller equivalent form. This reduction can be shown in Equation 2.8, where *m* is the number of ports.

$$\begin{bmatrix} I_1 \\ \vdots \\ I_m \end{bmatrix} = \begin{bmatrix} Y_{11} & \cdots & Y_{1m} \\ \vdots & \ddots & \vdots \\ Y_{m1} & \cdots & Y_{mm} \end{bmatrix} \begin{bmatrix} V_1 \\ \vdots \\ V_m \end{bmatrix} + \begin{bmatrix} C_1 \\ \vdots \\ C_m \end{bmatrix}$$
(2.8)

Several techniques exist to find these parameters. The most typical way is to short all ports to calculate the constant vector, then find all of the remaining elements by applying a test voltage at each port while the other ports remain shorted then finding the relation between **V** and **I** with a known constant. In application to this method the off diagonal elements are irrelevant and do not need to be calculated.

$$Y_{ii} = (I_i|_{V_i = V test and V j = 0} - C_i) / V_{test}$$

$$(2.9)$$

#### 2.4 Numerical Approach to Analyze the W matrix

Since the entire approach is based around a linear system model with a linear updating strategy, the W matrix as defined in Equation 1.7 can be derived numerically by Multivariate Linear Regression. After partitioning and running one simulation step the values of the line current can be obtained, sorted, and analyzed. For each set of line current multiple linear regression can be performed with respect to all of the previous iteration data for the other line currents. Explicitly this can be shown as

$$I_{linei}(i+1) = \alpha I_{linea}(i) + \beta I_{lineb}(i) \dots \dots + \eta I_{linex}(i) + C$$
(2.10)

for all i. Using multiple linear regression for each of the line currents, the coefficients can be found numerically within some tolerance of accuracy for each row in the W matrix. These rows can be combined into a large matrix which can be analyzed numerically. The spectral radius and rate of convergence can be determined, thus verifying if the multiport equivalent approximation is accurate or if the system should be repartitioned. This could lead to a simple Monte Carlo type partitioning optimization, which seems tractable on small scale systems, but for large systems, or systems which may change topology frequently, a more sophisticated partitioning optimization scheme may be required.

After running the simulator, the validity of the PGNME method to that exact partition can be evaluated through this process. If the coupling is relatively weak between subsystems, then the multiport equivalent should approximately force the W matrix to equal zero. If there is strong coupling, at the worst case, PGNME performance could degrade and no longer be beneficial depending on the system.

#### 2.5 Complexity Analysis

Complexity analysis is performed to analytically show the benefit of this method. During the simulation, there are four distinct portions; initialization, factorization, solving the network equation, and solving DAE's posed by the network components. Initialization is not as important; it is a one time cost and doesn't need to be analyzed in detail as it could be considered a small portion of the simulation.

#### 2.5.1 Factorization

Traditional LU decomposition has the complexity of  $\frac{2n^3}{3}$ , where *n* is the number of buses in the input. The PGNME method effectively breaks the factorization completely among all subsystems. The complexity of the PGNME method is described as:

$$c_F = \frac{2}{3} \left( \frac{n}{p} + \frac{d}{p} \right)^3$$
(2.11)

where p is number of partitions used, and d is the number of dummy buses in all of the subsystems combined. With this new complexity figure of merit, the theoretical speedup of factorization can be derived by the traditional formulation of

$$S = \frac{t_s}{t_p} \tag{2.12}$$

Where  $t_s$  is the serial algorithm run time, and  $t_p$  is the parallel algorithm runtime. Runtime and complexity have a direct relationship; therefore, complexity and time are interchangeable in the speedup formula. The total speedup given by the PGNME method to factorization is

$$S_F = \frac{n^3 p^3}{(n+d)^3}$$
(2.13)

If only very large problems where  $n \gg d$  are considered, then the speedup formula can be further simplified to be  $p^3$ .

#### 2.5.2 Solving

The decomposition strategy leaves two unique portions left to solve, the DAE representing the components and the network iterative method. The DAE solutions are

completely parallelizable. The original complexity can be described by complexity qn, where q is the average complexity of the DAE across all components. The new complexity for the PGNME method can be shown to be

$$c_{s1} = \frac{qn}{p} \tag{2.14}$$

The new speedup formula can then be shown to simply be

$$S_{s1} = p \tag{2.15}$$

For the network solution, LU method solving complexity can be shown to be  $2n^2$ . For PGNME this is the heart of the problem of partitioning and solving. This large network equation puts a constraint on the overall solution. The PGNME method effectively breaks this network into small pieces but also adds an iterative component and required message passing. The complexity of solving the network equation using PGNME can be shown to be

$$c_{s2} = 2k\left(\frac{n}{p} + \frac{d}{p}\right)^2 + 2kt_c(\frac{2d}{p})$$
 (2.16)

where k is the number of iterations required to converge within the tolerance region and  $t_c$  is the complexity of the MPI communication to send the boundary state data. The overall speedup against the original algorithm can be shown to be

$$S_{s2} = \frac{2n^2}{2k(\frac{n}{p})^2 + 2kt_c(\frac{2d}{p})}$$
(2.17)

For very large problems where n >> d the speedup can be shown to be  $\frac{p^2}{k}$ .

### 2.6 Discussion

This method provides a way to enhance the performance of the existing technique, making this method more attractive to scaling and solving larger systems with many partitions. Rather than randomly selecting the boundary equivalents, an insightful way to select appropriate boundary equivalent parameters is presented which will boost performance, making this method extremely viable for large scale power system dynamic simulation. In existing literature this method has only been shown to work with two partitions and it is not clear if there were dynamics implemented or not.

It is also important to note, that the factorization or inversion of the Y matrix is a very complex operation especially for large Y matrices. This method not only allows parallel solving of the network equations, it also permits the inversion to be broken into p smaller inversions. If there is a scenario where the topological structure of the network often changes then this method can give much more gain. Such scenarios include: Load Shedding, cascading failures, preventative and corrective relaying, etc.

This method not only provides significant gains in parallel computing but it also allows for distributed computing of the power system dynamic problem. The results in Chapter IV will show that this method does give significant gains, and in the interest of future wide area monitoring this algorithm could be used for distributed simulation. Different entities which are in charge of controlling the separate portions of a grid could have their own portion of the system model stored and do not need to share those detailed models with the rest of the system using this method. Each independent entity only needs to know the current boundary state. This could lead to imbalance if some entity is extremely small compared to another, but the large system could be split into smaller

40

subsystems on different resources or the small systems could be aggregated. This type of analysis has been attempted in previous work in 2 different geographical locations. The results were not too promising [KGRavikumar]. The delay in communications was too significant as they were using the internet connection, and packets took too long for fast analysis. But as technology advances speeds will get faster and if dedicated communication lines were installed for use only between the power system control centers, then this method could become an extremely beneficial tool for distributed simulation.

#### CHAPTER III

#### IMPLEMENTATION IN SOFTWARE

#### 3.1 **Program Structure**

The simulation software is constructed in C++ with a generic graph simulator as the frame of the constructed simulator. The simulation has a manager that controls convergence and counts which partitions have converged. All written code is optimized to run on Shadow II at the High Performance Computing Collaboratory at Mississippi State University. Shadow II consists of 110 nodes, each node containing 512 GB of RAM and 2 Intel E5-2680 v2 Ivy Bridge processors, which are each 10 core and operate at 2.8 GHz. The communication system is FDR InfiniBand. Systems of scale are created by duplicating base systems and making arbitrary connections between those systems. The topological structure of the system to be simulated is sent to hMETIS<sup>1</sup> [hMETIS], and the results are reformatted to a text format description of the system. Upon execution, the text file is read, and the system is initialized. Additional files tell the software manager which buses to monitor and which state variables to monitor. There are also separate files to define the generator parameters. The entire operation can be separated into three distinct phases: system generation, partitioning, and simulation.

<sup>&</sup>lt;sup>1</sup> hMETIS is a graph partitioning tool created by George Karypis at the University of Minnesota and is one of the most popular graph partitioning tools.

## **3.2 Graph generation**

System models are created from existing base models. Two sets of data are used; one constructed from an IEEE 118 bus model, and one set constructed from a model of the Eastern Interconnect (45,552 bus model). The data sets used in Chapter IV consist of these two models. The 118 systems are scaled three times on every level. The Eastern Interconnect systems are scaled five times on every level. Figure 3.1 shows the base IEEE 118 model in graph form, while Figure 3.2 shows Level 2, a 3× scale of level 1.



Figure 3.1 Original IEEE 118 graph



Figure 3.2 Level two, three times scale of original IEEE 118 system

## 3.3 Partitioning

The raw data describing the system is converted to an input format for hMETIS, hMETIS is then called to partition the data. The partitioning used in this thesis is random, meaning there is no optimization other than what hMETIS uses internally. Geographical and system specific information can be used to better cluster the partitions and create higher quality partitions; this should improve the performance of the simulation. An example output of the 118 system partitioned can be visualized in Figure 3.3 using Gephi<sup>2</sup> [GEHPI].



Figure 3.3 Gephi Partitioned graph visualization

#### 3.4 Simulation

The simulation process can be shown in Figure 3.4. The simulation manager reads all of the data from file and distributes the relevant data to the corresponding process. The manager then signals the beginning of the simulation to all of the processes. All of the processes solve their local problem then begin to iteratively solve the network equations.

<sup>&</sup>lt;sup>2</sup> Gephi is an open source graph based visualization tool

Each process will send its boundary states to the manager, the manager will share these states with the neighboring systems and check if the systems have converged. This process of updating the local states and iteratively



Figure 3.4 Simulation Flowchart

A short sample using the Intel MPI Trace Analyzer tool is shown in Figure 3.5. There are clearly distinct portions of the simulation: initialization, data reading, calculating multiport equivalents, simulation, and cleanup. First, there are background tasks necessary for the environment to be initialized. Second, the manager is continually reading in data while the other processes wait to receive this data. Third, the manager is waiting, and each process is calculating its own multiport equivalents then exchanging those equivalents with the neighboring processes. After the simulation has begun, the manager is mostly communicating between all of the processes, controlling the convergence checking. Finally, each process has some memory cleanup and other background tasks, and some statistics reporting to file and the environment is terminated.



Figure 3.5 Intel MPI Trace Analyzer

In Figure 3.6 a portion of the simulation is shown. There are four time steps shown in this figure. First the DAE describing the nodes are solved. Then the iterative PGNME process begins, each process is sending its local convergence state to the manager, and the manager makes a decision to continue the iterative process or stop it. In this particular portion, it is easy to see that the first two time steps shown have taken three iterations, and the last two time steps take four iterations.



Figure 3.6 Zoom in Trace Analyzer Simulation

#### CHAPTER IV

#### RESULTS

The results presented in this section are created by using the simulator described in Chapter III.

#### 4.1 Large scale results on speed and scale

In this section the PGNME method is subjected to extreme testing and monitored for its worst performance. In these large scale results an arbitrary excitation is injected at every bus in the system to intentionally impose iterations into the PGNME method. This effectively allows the worst case analysis of PGNME to be compared against PUR without multiport equivalent. This arbitrary excitation removes the solving of dynamic equations which would add to the total speedup. It also forces all of the dummy boundaries off at every iteration. The final solution at every time step would be close to the initial solution if real dynamics were injected, but a poor method can still take many iterations to converge, especially if the event is near a boundary port. The arbitrary excitation is done because the DAE portion, which excludes the network equation, is completely parallelizable, adding extremely detailed models would only add to the speedup and this section's intent is to put the network equation iterative portion to the test.

## 4.1.1 Original PUR vs PGNME

In this section the original PUR method is strictly compared against PGNME on the IEEE 118 system and the set created around it using the graph generation detailed in Chapter 3. The systems, from level 1 to level 9, or 118 bus to 774198 bus, were partitioned in a variety of ways depending on the specific system size. The average gain and minimum and maximum gain were calculated among the partitions.

	Max	Min	Avg	Max	Min		
	iterations	iterations	iterations	speedup	speedup	Avg speedup	
Number	saved per	saved per	saved per	over non	over non	over non	
of Buses	step	step	step	multiport	multiport	multiport	Level
118	10.4817	0.1816	4.680575	2.384768	1.169799	1.758652868	1
354	10.1753	2.0865	5.6309	2.45075	1.524763	2.048176138	2
1062	10.5036	2.0261	4.952067	4.162261	1.432271	2.278644841	3
3186	4.8656	0.4218	2.048767	2.579835	1.153845	1.505306275	4
9558	13.1688	2.5233	5.85	3.283138	1.576421	2.126766216	5
28674	15.9006	8.4218	12.06052	5.197137	3.183372	4.131277935	6
86022	16.1188	6.1277	10.27393	5.130228	3.095462	4.156629679	7
258066	17.1107	8.5457	12.38565	5.322544	2.670353	4.107596802	8
774198	16.9997	5.6635	12.0799	5.334542	2.792362	4.286615856	9

Table 4.1Comparing PUR and PGNME performance

As seen in the results, there is always improvement over the original PUR method, while it is sometimes minimal, especially in smaller systems such as the 1.17

times speedup on the 118 system and the 1.15 on the 3,186 bus system. This fluctuation is believed to be caused by the estimation of the **W** matrix. If the **W** matrix approximation can be maintained, the results should be optimal and iterations should be minimized completely. With larger systems, the improvement is fairly significant, with over five times speedup on the systems above 28,000 buses. This testing shows the clear advantage of PGNME and its contribution to the solution of the network equation in the model. Parallel computing may not be necessary for the small systems, but the systems of significant size all benefit greatly. Overall the addition of the multiport equivalent method gives more than 3.1 times speedup to simulations, on average PGNME needs 3.5 iterations to converge versus the 11.8 required for PUR. This method drastically improves the previous implementation and creates a faster and more efficient parallel implementation. If combined with optimal partitioning, the method could be improved even further.

#### 4.1.2 Increasing Number of Partitions

With all parallel algorithms, increasing the number of partitions does not always lead to an increase in performance. Depending on the application, more partitions eventually leads to worse performance. The granularity of the specific problem plays a huge role in this characteristic of the problem. As the problem is broken into smaller pieces, the overhead cost of parallelization eventually overwhelms the benefit. Very small problems will not benefit from utilization of HPC resources.

In Figure 4.1 a 1062 bus system is shown. It is apparent that the problem is so small that parallelizing it using this method gives a mere 3.3 maximum speed up. While in the large problem of the 258k bus system of Figure 4.2, there is a maximum of over

100x speedup. For any given particular problem, this curve can be created, and based on the curve, an appropriate number of partitions can be selected. After this number is selected, optimal partitioning can be done to ensure that the iteration will be controlled when applying the PGNME method.



Figure 4.1 Speedup of 1062 bus system



Figure 4.2 Speedup curve of 258,066 bus system

The curves presented in Figure 4.1 and Figure 4.2 are created using the curve fitting tool in matlab. To create the smoothed curves, the bad performers are also neglected in these curves. Bad performers are defined as partitioning schemes where the PGNME approximation is invalid, or the residuals dominate the approximation due to port coupling, and iteration becomes significantly larger. In Figure 4.3 the real data is graphed for a 28k system. As seen in the figure, iteration is slightly different between different partitioning runs leading to load imbalances. There are also other small factors which affect the imbalance such as: load imbalance if the number of nodes aren't equally distributed, usage of the computer at the run time, OS housekeeping duties and background threads, and other related factors such as these.



Figure 4.3 Effect of increasing partitions

#### 4.1.3 **Repartitioning the same system**

This section shows the effect of repartitioning the same systems and the resulting performance. In this section the level 4 system of the 118 bus class was selected and 30 hMETIS runs were run to partition the system into 40 partitions. From this random run of hMETIS, it is clear that partitioning still plays a role in the performance of the method. This effect can be minimized if an appropriate algorithm is implemented to prevent the ports from being "electrically close" to each other. Other types of optimization can be used as well such as a Monte Carlo type optimization, which is essentially what is shown here. In this very small sample of 30 hMETIS runs, it is shown that the average iteration varies between less than 6 to above 12. If a larger sample size is used, the iteration could possibly be reduced even further.

From the 30 random hMETIS runs, it is clear that some of the partitions created are relatively poor performers compared to the others. Even if one iteration can be saved on every time step, it can lead to huge gains over the course of an entire simulation.

If the partitions are held constant and the DAE describing the components are balanced between different partitions, then iteration has a linear relationship to the time required for simulations. Iteration is seen as the main bottleneck of this algorithm. As shown in Figure 4.4, 12 iterations leads to a 9s simulation while half as many iterations leads to half the required simulation time. This relationship will remain true with simple DAE's describing the components. When the generator and load models to be solved at each vertex become very complex, then the time required for the solution of the network equation will be an overall smaller portion of the simulation.



Figure 4.4 Repartitioning the 3186 bus system



Figure 4.5 Cumulative distribution of iterations from partitioning

Figure 4.5 shows the cumulative probability distribution of iterations. For this particular system the majority of partitioning schemes need an average of 6 iterations to converge. A figure such as this could be created for an arbitrary system to determine if this particular number of partitions is feasible or if partition number should change

#### 4.1.4 Scalability of the simulator

In this section the data is shown from many levels of graphs to show the scalability of the simulator. For every parallel computing problem, a speedup curve can be created. At some point on the curve, due to the nature of the problem, the curve will reach a maximum. After the maximum, the cost associated with the parallelization of the

problem (communication, added computation, etc) becomes a bottleneck. After this point, the problem cannot be made smaller without penalty.

A particular algorithm is scalable if for an increase in input size there is a corresponding increase in speedup. As shown in Figure 4.5, the small 118 bus system does not benefit much from parallelization with a maximum of around 2× speedup obtained. As the input is increased, the benefit of parallelization steadily increases. This shows that the algorithm is scalable, and as the input continues to grow then the gain from parallel computing will also continue to grow, at least until some portion of the algorithm becomes a bottleneck. The simulator is shown to be scalable up to a 774,198 bus system, and there are also data points from larger systems overlaid here. It is important to note that these curves are estimated from data and that all of the partitions which were bad performers(i.e. high iteration) were removed from the data set, in effect simulating the action of a partitioning optimizer which was not available during this work. These graphs assume that partitioning produces a relatively high quality set of partitions which converge on average in very few iterations.



Figure 4.6 Scalability of the simulator

# 4.2 Dynamic Simulation

For this section transient stability simulation is performed using the classical generator model. The model includes Equations 4.1 - 4.4 in addition to the network equation of Equation 1.3.

$$\dot{\omega} = \frac{f\pi}{H} (P_m - P_e - D(\omega - \omega_0)) \tag{4.1}$$

$$\delta = \omega - \omega_0 \tag{4.2}$$

$$Pe = \frac{|V||E_q|}{X_d} \sin(\delta - \theta)$$
(4.3)

$$I = \frac{|E_q|e^{j\delta}}{jX'_d} \tag{4.4}$$

This model is used in [PNNL] for look ahead dynamic simulation. For a large area multi-machine study, this implementation shows great promise. In this section a 1ms time step is used with Forward Euler to solve Equation 4.1 and Equation 4.2. The results from a model of the Eastern Interconnect (45k bus) are shown in Figure 4.6. The best performance comes from 160 partitions and gives a speedup of 60. The 5s simulation takes 145s in the serial version while taking only 2.38s on 160 CPU.

This results in a faster than real time simulation of a system the size of the Eastern Interconnec. Depending on the intent of the simulation, an efficient varying time step implicit method could be implemented to incorporate a much larger time step as the dynamics permit. If this is done the simulation would be even faster. On top of this, different compilers could be tested to find the optimal combination of MPI libraries and compilers for the given hardware.



Figure 4.7 Speedup curve using classical generator models in the Eastern Interconnect scale system

#### 4.3 Matlab results on accuracy and different models

In this section, the PGNME method is compared against an existing trusted software tool known as MatDyn to ensure accuracy. The MatDyn implementation is of a nine-bus system, which consists of two different generator models(classical model, and a fourth order model), a turbine governor model and an excitation system model. The PGNME method is implemented in Matlab by modifying the source code of MatDyn for this specific case. In this specific case, one line was cut in the nine-bus system, forming two subsystems which are connected by The purpose of this test is to reassure that this method will be convergent on a more realistic model. RK45 is used and the reconciliation tolerance is  $10^{-4}$ . It is shown in Figure 4.8 and 4.9 that the decoupled response overalys to within a tolerance of the original, verifying that the algorithm can handle robust dynamic events.



Figure 4.8 Generator speed response during an event



Figure 4.9 Bus Voltage response during an event
#### CHAPTER V

## CONCLUSION AND FUTURE WORKS

This thesis has considered a novel methodology for solving large scale systems of DAE, such as the model of a large power transmission system, on a high performance cluster computer. The treatment was in the form of the ASM approach. It has also presented a novel approach to evolve an existing technique called PGN, resulting in a new technique called PGNME that scales to large system simulation with many partitions as shown by the results from running the solver on a large HPC cluster computer.

The overall result of this method is very impressive compared to other iterative methods and other relaxation methods which do not have the same form. Other methods show very small speedup on hundreds of processors making them very unattractive. This method shows fairly efficient utilization of HPC resources. High fidelity dynamic models will lead to much more efficient implementation as the solution of the differential equations in the ASM process is extremely parallelizable. When higher fidelity models are used the speedup will increase dramatically. Also if simulations were run where the Y matrix changed often this method would excel even more, as the factorization portion is completely parallel and does not have an iterative component.

While this method shows to be very useful in progress towards look ahead and real time simulation, more work can be done to further enhance its performance. These future works include: optimizing the initialization process, continuing to search for ways to make the code more efficient, partitioning optimization, developing more advanced models especially for the other physical components and control systems that need to be modeled, implementing more integration techniques such as RK45 and Trapezoidal rule, and exploring advanced integration techniques such as multigrid and parareal. Immediately, partitioning optimization would appear to be a high priority to extend the potential benefit of PGNME to the limits of the HPC machinese the simulation runs on.

#### REFERENCES

- [BStott] B. Stott, "Power system dynamic response calculations," Proceedings of the IEEE, vol. 67, no. 2, pp. 219–241, 1979.
- [CFu] C. Fu, J. McCalley, and J. Tong, "A numerical solver design for extended-term time-domain simulation," Power Systems, IEEE Trans- actions on, vol. 28, pp. 4926–4935, Nov 2013
- [PNNL] Shuangshuang Jin, Zhenyu Huang, Ruisheng Diao, Di Wu and Yousu Chen, 'Parallel implementation of power system dynamic simulation', 2013 IEEE Power & Energy Society General Meeting, 2013.
- [OpalRT] Opal-rt.com, 'RTS HIL PHIL RCP | OPAL-RT, World simulation technology leader', 2015. [Online]. Available: http://www.opal-rt.com/. [Accessed: 04- Aug-2015].
- [RTDS] RTDS Technologies Inc., 'RTDS Technologies Inc. | Real Time Digital Power System Simulation', 2013. [Online]. Available: http://rtds.com. [Accessed: 04-Aug-2015].
- [QHuang] Qinghua Huang, Jian Wu, Jimena Bastos And Noel N. Schulz, "Distributed Simulation Applied to Shipboard Power Systems," Proceedings Of The IEEE Electric Ship Technologies Sympoisum (ESTS 07), Arlington, Virginia, May 2007.
- [JWu] J. Wu and N. N. Schulz, "Generalized three phase coupling method for distributed simulation," in Proc. 37th Annual North American Power Symp., Ames, IA, Oct. 23–25, 2005, pp. 441–447.
- [WRen] Wei Ren; Steurer, M.; Baldwin, T.L. "Improve the Stability and the Accuracy of Power Hardware-in-the-Loop Simulation by Selecting Appropriate Interface Algorithms", Industry Applications, IEEE Transactions on, On page(s): 1286 -1294 Volume: 44, Issue: 4, July-aug. 2008
- [Russian] V. B. Dmitriev-Zdorov, "Generalized coupling as a way to improve the convergence in relaxation-based solvers," in Proc. EURO-DAC/ EUROVHDL Exhib., Geneva, Switzerland, Sep. 1996.

- [YFu] A. Kargarian, Y. Fu and Z. Li, 'Distributed Security-Constrained Unit Commitment for Large-Scale Power Systems', IEEE Trans. Power Syst., vol. 30, no. 4, pp. 1925-1936, 2015.
- [DOE] "Computational needs for the next generation electric grid," The U.S. Department Of Energy, April 2011.
- [DKeyes] D. E. Keyes, Domain Decomposition Methods for Partial Differential Equations. No. 55, Siam, 1992.
- [VDolean] V. Dolean, P. Jolivet, and F. Nataf, "An introduction to domain decomposition methods: algorithms, theory and parallel implementation," 2015.
- [LHageman] L.A. Hageman, D.M. young, Applied iterative methods, Academy press, NY. 1981
- [HDommel] Dommel and N. Sato, "Fast transient stability soultions," Power Apparatus and Systems, IEEE Transactions on, no. 4, pp. 1643–1650, 1972.
- [Thyme] Web.ornl.gov, "Toolkit for HYbrid Modeling of Electric power systems", 2016. [Online]. Available: http://web.ornl.gov/~1qn/thyme/docs/. [Accessed: 23- Feb-2016].
- [PNNL2] O. Villa, A. Tumeo, S. Ciraci, J. A. Daily and J. C. Fuller, "A High Performance Computing Network and System Simulator for the Power Grid: NGNS<sup>2</sup>," High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion:, Salt Lake City, UT, 2012, pp. 313-322.
- [VOpalRT] V. Jalili-Marandi, E. Robert, V. Lapointe and J. Bélanger, "A real-time transient stability simulation tool for large-scale power systems," Power and Energy Society General Meeting, 2012 IEEE, San Diego, CA, 2012, pp. 1-7.
- [Chinese2] Z. Min, X. Dechao, L. Yalou and A. Ning, "Algorithm research on parallel topology of large-scale power system on-line simulation," Power System Technology (POWERCON), 2014 International Conference on, Chengdu, 2014, pp. 55-60.
- [Kundur] P. Kundur, N. Balu and M. Lauby, Power system stability and control. New York: McGraw-Hill, 1994.
- [SUPERLU] Lawrence Berkeley National Laboratory, "SuperLU Users' Guide", 1999.
- [KELLEY] C. Kelley, Iterative methods for linear and nonlinear equations. Philadelphia: Society for Industrial and Applied Mathematics, 1995.
- [SAAD] Y. Saad, Iterative methods for sparse linear systems. Philadelphia: SIAM, 2003.

- [KGRavikumar] K. G. Ravikumar, N. N. Schulz and A. K. Srivastava, "Distributed simulation of power systems using real-time digital simulator," Power Systems Conference and Exposition, 2009. PSCE '09. IEEE/PES, Seattle, WA, 2009, pp. 1-6.
- [VDolean] V. Dolean, P. Jolivet, and F. Nataf, "An introduction to domain decomposition methods: algorithms, theory and parallel implementation," 2015.
- [hMETIS] "hMETIS Hypergraph & Circuit Partitioning | Karypis Lab", Glaros.dtc.umn.edu, 2016. [Online]. Available: http://glaros.dtc.umn.edu/gkhome/metis/hmetis/overview. [Accessed: 03- Jun-2016].
- [GEPHI] "Gephi The Open Graph Viz Platform", Gephi.org, 2016. [Online]. Available: https://gephi.org/. [Accessed: 03- Jun- 2016].

APPENDIX A

# W MATRIX DERIVATIONS

This appendix is dedicated to explicitly deriving the W matrix for the PGNME iterative method strategy .

# A.1 Two Partition Example



Figure A.1 Two Partition Equivalent Description

For the explicit derivation it is important to first define all physical laws

controlling the system then to define the iterative updating method.

$$I_a(k) = (V_1(k) - V_{d1}(k))Y_{ab}$$
(A.1)

$$I_b(k) = (V_{d2}(k) - V_2(k))Y_{ab}$$
(A.2)

$$V_1(k) = \frac{I_1 - I_a(k)}{Y_a}$$
(A.3)

$$V_{d1}(k) = \frac{S_1(k) + I_a(k)}{Y_1}$$
(A.4)

$$V_2(k) = \frac{I_2 + I_b(k)}{Y_b}$$
(A.5)

$$V_{d2}(k) = \frac{S_2(k) - I_b(k)}{Y_2}$$
(A.6)

$$S_1(k+1) = V_2(k)Y_1 - I_b(k)$$
(A.7)

$$S_2(k+1) = V_1(k)Y_2 + I_a(k)$$
(A.8)

Using all of the previous equations the W matrix can be derived by solving for the updated states.

$$I_a(k+1) = (V_1(k+1) - V_{d1}(k+1))Y_{ab}$$
(A.9)

$$I_a(k+1) = \left(\frac{I_1 - I_a(k+1)}{Y_a} - \frac{S_1(k+1) + I_a(k+1)}{Y_1}\right) Y_{ab}$$
(A.10)

$$I_a(k+1) = \left(\frac{I_1 - I_a(k+1)}{Y_a} - \frac{V_2(k)Y_1 - I_b(k) + I_a(k+1)}{Y_1}\right)Y_{ab}$$
(A.11)

$$I_a(k+1) = \left(\frac{I_1}{Y_a} - \frac{I_a(k+1)}{Y_a} - \frac{I_2 + I_b(k)}{Y_b} + \frac{I_b(k)}{Y_1} - \frac{I_a(k+1)}{Y_1}\right) Y_{ab}$$
(A.12)

$$I_a(k+1) = \left(\frac{I_1}{Y_a} - \frac{I_a(k+1)}{Y_a} - \frac{I_2}{Y_B} - \frac{I_b(k)}{Y_b} + \frac{I_b(k)}{Y_1} - \frac{I_a(k+1)}{Y_1}\right) Y_{ab}$$
(A.13)

$$I_a(k+1) = \frac{I_1 Y_{ab}}{Y_a} - \frac{I_a(k+1)Y_{ab}}{Y_a} - \frac{I_2 Y_{ab}}{Y_B} - \frac{I_b(k)Y_{ab}}{Y_b} + \frac{I_b(k)Y_{ab}}{Y_1} - \frac{I_a(k+1)Y_{ab}}{Y_1}$$
(A.14)

$$I_a(k+1) = -\frac{I_a(k+1)Y_{ab}}{Y_a} - \frac{I_a(k+1)Y_{ab}}{Y_1} - \frac{I_b(k)Y_{ab}}{Y_b} + \frac{I_b(k)Y_{ab}}{Y_1} + \frac{I_1Y_{ab}}{Y_a} - \frac{I_2Y_{ab}}{Y_B}$$
(A.15)

$$I_a(k+1) = I_a(k+1)\left(-\frac{Y_{ab}}{Y_a} - \frac{Y_{ab}}{Y_1}\right) + I_b(k)\left(-\frac{Y_{ab}}{Y_b} + \frac{Y_{ab}}{Y_1}\right) + \frac{I_1Y_{ab}}{Y_a} - \frac{I_2Y_{ab}}{Y_B}$$
(A.16)

$$I_a(k+1)(1+\frac{Y_{ab}}{Y_a}+\frac{Y_{ab}}{Y_1}) = I_b(k)(-\frac{Y_{ab}}{Y_b}+\frac{Y_{ab}}{Y_1}) + \frac{I_1Y_{ab}}{Y_a} - \frac{I_2Y_{ab}}{Y_B}$$
(A.17)

$$I_a(k+1)(1+\frac{Y_{ab}}{Y_a}+\frac{Y_{ab}}{Y_1}) = I_b(k)(-\frac{Y_{ab}}{Y_b}+\frac{Y_{ab}}{Y_1}) + C$$
(A.18)

$$I_{a}(k+1) = I_{b}(k) \frac{\left(-\frac{Y_{ab}}{Y_{b}} + \frac{Y_{ab}}{Y_{1}}\right)}{\left(1 + \frac{Y_{ab}}{Y_{a}} + \frac{Y_{ab}}{Y_{1}}\right)} + C$$
(A.19)

$$I_a(k+1) = I_b(k) \frac{(\frac{1}{Y_1} - \frac{1}{Y_b})}{(\frac{1}{Y_{ab}} + \frac{1}{Y_a} + \frac{1}{Y_1})} + C$$
(A.20)

$$I_b(k+1) = (V_{d2}(k+1) - V_2(k+1))Y_{ab}$$
(A.21)

$$I_b(k+1) = \left(\frac{S_2(k+1) - I_b(k+1)}{Y_2} - \frac{I_2 + I_b(k+1)}{Y_b}\right) Y_{ab}$$
(A.22)

$$I_b(k+1) = \left(\frac{V_1(k)Y_2 + I_a(k) - I_b(k+1)}{Y_2} - \frac{I_2 + I_b(k+1)}{Y_b}\right)Y_{ab}$$
(A.23)

$$I_b(k+1) = \left(\frac{\frac{I_1 - I_a(k)}{Y_a}Y_2 + I_a(k) - I_b(k+1)}{Y_2} - \frac{I_2 + I_b(k+1)}{Y_b}\right)Y_{ab}$$
(A.24)

$$I_b(k+1) = \frac{I_1 Y_{ab}}{Y_a} - \frac{I_a(k) Y_{ab}}{Y_a} + \frac{I_a(k) Y_{ab}}{Y_2} - \frac{I_b(k+1) Y_{ab}}{Y_2} - \frac{I_2 Y_{ab}}{Y_b} - \frac{I_b(k+1) Y_{ab}}{Y_b}$$
(A.25)

$$I_b(k+1) = I_b(k+1) \left( -\frac{Y_{ab}}{Y_b} - \frac{Y_{ab}}{Y_2} \right) + I_a(k) \left( \frac{Y_{ab}}{Y_2} - \frac{Y_{ab}}{Y_a} \right) + \frac{I_1 Y_{ab}}{Y_a} - \frac{I_2 Y_{ab}}{Y_b}$$
(A.26)

$$I_b(k+1)(1+\frac{Y_{ab}}{Y_b}+\frac{Y_{ab}}{Y_2}) = I_a(k)\left(\frac{Y_{ab}}{Y_2}-\frac{Y_{ab}}{Y_a}\right) + \frac{I_1Y_{ab}}{Y_a} - \frac{I_2Y_{ab}}{Y_b}$$
(A.27)

$$I_b(k+1) = I_a(k) \frac{\left(\frac{Y_{ab}}{Y_2} - \frac{Y_{ab}}{Y_a}\right)}{\left(1 + \frac{Y_{ab}}{Y_b} + \frac{Y_{ab}}{Y_2}\right)} + \frac{\frac{I_1Y_{ab}}{Y_a} - \frac{I_2Y_{ab}}{Y_b}}{\left(1 + \frac{Y_{ab}}{Y_b} + \frac{Y_{ab}}{Y_2}\right)}$$
(A.28)

$$I_b(k+1) = I_a(k) \frac{\left(\frac{1}{Y_2} - \frac{1}{Y_a}\right)}{\left(\frac{1}{Y_{ab}} + \frac{1}{Y_b} + \frac{1}{Y_2}\right)} + C$$
(A.29)

This derivation can be written in the form used mostly in literature as in Equation 1.7.

$$\begin{bmatrix} I_a(k+1)\\ I_b(k+1) \end{bmatrix} = \begin{bmatrix} 0 & \frac{(\frac{1}{Y_1} - \frac{1}{Y_b})}{(\frac{1}{Y_2} - \frac{1}{Y_a})} \\ \frac{(\frac{1}{Y_2} - \frac{1}{Y_a})}{(\frac{1}{(\frac{1}{Y_{ab}} + \frac{1}{Y_{b}} + \frac{1}{Y_{2}})} & 0 \end{bmatrix} \begin{bmatrix} I_a(k)\\ I_b(k) \end{bmatrix}$$
(A.30)

# A.2 Three Partition Example



Figure A.2 Subcircuit 1 and 2



Figure A.3 Subcircuit 3 and 4

A similar approach should be taken for the three partition example. All physical laws and updating equations should be derived first.

$$I_{a}(k) = (V_{a}(k) - V_{d1}(k))Y_{ab}$$
(A.31)

$$I_b(k) = (V_{d2}(k) - V_1(k))Y_{ab}$$
(A.32)

$$I_{c}(k) = (V_{2}(k) - V_{d3}(k))Y_{bc}$$
(A.33)

$$I_d(k) = (V_{d4}(k) - V_c(k))Y_{bc}$$
(A.34)

$$V_a(k) = \frac{I_1 - I_a(k)}{Y_a}$$
(A.35)

$$V_{d1}(k) = \frac{S_1(k) + I_a(k)}{Y_1}$$
(A.36)

$$V_{d2}(k) = \frac{S_2(k) - I_b(k)}{Y_2}$$
(A.37)

$$V_1(k) = \frac{h_{21}V_2(k) + I_b(k)}{Y_{11}}$$
(A.38)

$$V_2(k) = \frac{h_{12}V_1(k) - I_c(k)}{Y_{22}}$$
(A.39)

$$V_{d3}(k) = \frac{S_3(k) + I_c(k)}{Y_3}$$
(A.40)

$$V_{d4}(k) = \frac{S_4(k) - I_d(k)}{Y_4}$$
(A.41)

$$V_c(k) = \frac{I_3 + I_d(k)}{Y_c}$$
(A.42)

$$S_1(k+1) = V_1(k)Y_1 - I_b(k)$$
(A.43)

$$S_2(k+1) = V_a(k)Y_2 + I_a(k)$$
 (A.44)

$$S_3(k+1) = V_c(k)Y_3 - I_d(k)$$
(A.45)

$$S_4(k+1) = V_2(k)Y_4 + I_c(k)$$
(A.46)

$$I_a(k+1) = (V_a(k+1) - V_{d1}(k+1))Y_{ab}$$
(A.47)

$$I_a(k+1) = \left(\frac{I_1 - I_a(k+1)}{Y_a} - \frac{S_1(k+1) + I_a(k+1)}{Y_1}\right) Y_{ab}$$
(A.48)

$$I_a(k+1) = \frac{Y_{ab}I_1}{Y_a} - \frac{Y_{ab}}{Y_a}I_a(k+1) - \frac{Y_{ab}}{Y_1}S_1(k+1) - I_a(k+1)\frac{Y_{ab}}{Y_1}$$
(A.49)

$$I_a(k+1)(1+\frac{Y_{ab}}{Y_a}+\frac{Y_{ab}}{Y_1}) = \frac{Y_{ab}I_1}{Y_a} - \frac{Y_{ab}}{Y_1}S_1(k+1)$$
(A.50)

$$I_{a}(k+1)\left(1+\frac{Y_{ab}}{Y_{a}}+\frac{Y_{ab}}{Y_{1}}\right) = \frac{Y_{ab}I_{1}}{Y_{a}} - \frac{Y_{ab}}{Y_{1}}\left(\left(\frac{h_{21}V_{2}(k)+I_{b}(k)}{Y_{11}}\right)Y_{1} - I_{b}(k)\right)$$
(A.51)

$$I_{a}(k+1)\left(1+\frac{Y_{ab}}{Y_{a}}+\frac{Y_{ab}}{Y_{1}}\right) = \frac{Y_{ab}I_{1}}{Y_{a}} - Y_{ab}\left(\frac{h_{21}V_{2}(k)+I_{b}(k)}{Y_{11}}\right) + I_{b}(k)\frac{Y_{ab}}{Y_{1}}$$
(A.52)

$$I_a(k+1)\left(\frac{1}{Y_{ab}} + \frac{1}{Y_a} + \frac{1}{Y_1}\right) = \frac{I_1}{Y_a} - \frac{h_{21}V_2(k)}{Y_{11}} - \frac{I_b(k)}{Y_{11}} + I_b(k)\frac{1}{Y_1}$$
(A.53)

$$I_a(k+1)\left(\frac{1}{Y_{ab}} + \frac{1}{Y_a} + \frac{1}{Y_1}\right) = \frac{I_1}{Y_a} - \frac{h_{21}}{Y_{11}}V_2(k) + I_b(k)\left(\frac{1}{Y_1} - \frac{1}{Y_{11}}\right)$$
(A.54)

$$V_1(k) = \frac{h_{21}V_2(k) + I_b(k)}{Y_{11}}, V_2(k) = \frac{h_{12}V_1(k) - I_c(k)}{Y_{22}}$$
(A.55)

$$V_2(k) = \frac{h_{12}}{Y_{22}} V_1(k) - \frac{I_c(k)}{Y_{22}}$$
(A.56)

$$V_2(k) = \frac{h_{12}}{Y_{22}} \left( \frac{h_{21}}{Y_{11}} V_2(k) + \frac{I_b(k)}{Y_{11}} \right) - \frac{I_c(k)}{Y_{22}}$$
(A.57)

$$V_2(k) = \frac{h_{12}h_{21}}{Y_{11}Y_{22}}V_2(k) + \frac{h_{12}}{Y_{11}Y_{22}}I_b(k) - \frac{I_c(k)}{Y_{22}}$$
(A.58)

$$V_2(k)(1 - \frac{h_{12}h_{21}}{Y_{11}Y_{22}}) = \frac{h_{12}}{Y_{11}Y_{22}}I_b(k) - \frac{I_c(k)}{Y_{22}}$$
(A.59)

$$V_2(k) = \frac{\frac{h_{12}}{Y_{11}Y_{22}}}{1 - \frac{h_{12}h_{21}}{Y_{11}Y_{22}}} I_b(k) - \frac{\frac{I_c(k)}{Y_{22}}}{1 - \frac{h_{12}h_{21}}{Y_{11}Y_{22}}}$$
(A.60)

$$V_2(k) = \frac{h_{12}}{Y_{11}Y_{22} - h_{12}h_{21}} I_b(k) - \frac{Y_{11}}{Y_{11}Y_{22} - h_{12}h_{21}} I_c(k)$$
(A.61)

$$I_a(k+1)\left(\frac{1}{Y_{ab}} + \frac{1}{Y_a} + \frac{1}{Y_1}\right) = \frac{I_1}{Y_a} - \frac{h_{21}}{Y_{11}}V_2(k) + I_b(k)\left(\frac{1}{Y_1} - \frac{1}{Y_{11}}\right)$$
(A.62)

$$I_a(k+1)\left(\frac{1}{Y_{ab}} + \frac{1}{Y_a} + \frac{1}{Y_1}\right) = \frac{I_1}{Y_a} - \frac{h_{21}}{Y_{11}}\left(\frac{h_{12}}{Y_{11}Y_{22} - h_{12}h_{21}}I_b(k) - \frac{Y_{11}}{Y_{11}Y_{22} - h_{12}h_{21}}I_c(k)\right) + \frac{I_1}{Y_{11}Y_{22} - h_{12}h_{21}}I_b(k) - \frac{I_1}{Y_{11}Y_{22} - h_{12}h_{21}}I_c(k) + \frac{I_1}{Y_{11}Y_{22} - h_{12}h_{21}}I_b(k) - \frac{I_1}{Y_{11}Y_{22} - h_{12}h_{21}}I_c(k) + \frac{I_1}{Y_{11}Y_{22} - h_{12}h_{21}}I_c(k) - \frac{I_1}{Y_{11}Y_{22} - h_{12}h_{21}}I_c(k) + \frac{I_1}{Y_{11}Y_{22} - h_{12}h_{21}}I_c(k) - \frac{I_1}{Y_{11}Y_{22} - h_{12}h_{21}}I_c(k) + \frac{I_1}{Y_{11}Y_{22} - h_{12}h_{21}}I_c(k) - \frac{I_1}{Y_{11}Y_{$$

$$I_{b}(k)(\frac{1}{Y_{1}} - \frac{1}{Y_{11}})$$
(A.63)  

$$I_{a}(k+1)(\frac{1}{Y_{ab}} + \frac{1}{Y_{a}} + \frac{1}{Y_{1}}) = \frac{I_{1}}{Y_{a}} - \frac{h_{12}h_{21}}{Y_{11}(Y_{11}Y_{22} - h_{12}h_{21})}I_{b}(k) + \frac{h_{21}}{Y_{11}Y_{22} - h_{12}h_{21}}I_{c}(k) + I_{b}(k)(\frac{1}{Y_{1}} - \frac{1}{Y_{11}})$$
(A.64)  

$$I_{a}(k+1)(\frac{1}{Y_{ab}} + \frac{1}{Y_{a}} + \frac{1}{Y_{1}}) = \frac{I_{1}}{Y_{a}} + \frac{h_{21}}{Y_{11}Y_{22} - h_{12}h_{21}}I_{c}(k) + I_{b}(k)(\frac{1}{Y_{1}} - \frac{1}{Y_{11}} - \frac{h_{12}h_{21}}{Y_{11}(Y_{11}Y_{22} - h_{12}h_{21})})$$
(A.65)

If MPE is set correctly, then the terms become 0 plus residual which is completely dependent on the electrical coupling of the subsystems.

$$I_{a}(k+1) = \frac{\frac{I_{1}}{Y_{a}}}{\left(\frac{1}{Y_{a}} + \frac{1}{Y_{a}} + \frac{1}{Y_{1}}\right)} + \frac{\frac{h_{21}}{Y_{11}Y_{22} - h_{12}h_{21}}}{\left(\frac{1}{Y_{ab}} + \frac{1}{Y_{a}} + \frac{1}{Y_{1}}\right)} I_{c}(k) + I_{b}(k) \frac{\left(\frac{1}{Y_{11}} - \frac{1}{Y_{11}} - \frac{h_{12}h_{21}}{Y_{11}(Y_{11}Y_{22} - h_{12}h_{21})}\right)}{\left(\frac{1}{Y_{ab}} + \frac{1}{Y_{a}} + \frac{1}{Y_{1}}\right)}$$
(A.66)

$$I_b(k+1) = (V_{d2}(k+1) - V_1(k+1))Y_{ab}$$
(A.67)

$$I_b(k+1) = V_{d2}(k+1)Y_{ab} - V_1(k+1)Y_{ab}$$
(A.68)

$$I_b(k+1) = \frac{S_2(k+1)Y_{ab} - I_b(k+1)Y_{ab}}{Y_2} - \frac{h_{21}V_2(k+1)Y_{ab} + I_b(k+1)Y_{ab}}{Y_{11}}$$
(A.69)

$$I_b(k+1) = \frac{S_2(k+1)Y_{ab}}{Y_2} - \frac{I_b(k+1)Y_{ab}}{Y_2} - \frac{h_{21}V_2(k+1)Y_{ab}}{Y_{11}} - \frac{I_b(k+1)Y_{ab}}{Y_{11}}$$
(A.70)

$$I_b(k+1)\left(1+\frac{Y_{ab}}{Y_{11}}+\frac{Y_{ab}}{Y_2}\right) = \frac{S_2(k+1)Y_{ab}}{Y_2} - \frac{h_{21}V_2(k+1)Y_{ab}}{Y_{11}}$$
(A.71)

$$I_b(k+1)\left(1+\frac{Y_{ab}}{Y_{11}}+\frac{Y_{ab}}{Y_2}\right) = \frac{Y_{ab}}{Y_2}S_2(k+1) - \frac{h_{21}Y_{ab}}{Y_{11}}V_2(k+1)$$
(A.72)

$$V_2(k) = \frac{h_{12}}{Y_{11}Y_{22} - h_{12}h_{21}} I_b(k) - \frac{Y_{11}}{Y_{11}Y_{22} - h_{12}h_{21}} I_c(k)$$
(A.73)

$$I_b(k+1)(1+\frac{Y_{ab}}{Y_{11}}+\frac{Y_{ab}}{Y_2}) = \frac{Y_{ab}}{Y_2}(V_a(k)Y_2 + I_a(k)) - \frac{h_{21}Y_{ab}}{Y_{11}}(\frac{h_{12}}{Y_{11}Y_{22} - h_{12}h_{21}})I_b(k+1) - \frac{h_{21}Y_{ab}}{Y_2}(\frac{h_{22}}{Y_{11}})I_b(k+1) - \frac{h_{21}Y_{ab}}{Y_2}(\frac{h_{22}}{Y_{11}})I_b(k+1) - \frac{h_{22}Y_{ab}}{Y_2}(\frac{h_{22}}{Y_{11}})I_b(k+1) - \frac{h_{22}Y_{ab}}{Y_2}(\frac{h_{22}}{Y_{11}})I_b(k+1) - \frac{h_{22}Y_{ab}}{Y_2}(\frac{h_{22}}{Y_{11}})I_b(k+1) - \frac{h_{22}Y_{ab}}{Y_2}(\frac{h_{22}}{Y_{11}})I_b(k+1) - \frac{h_{22}Y_{ab}}{Y_2}(\frac{h_{22}}{Y_{11}})I_b(k+1) - \frac{h_{22}Y_{ab}}{Y_{11}}(\frac{h_{22}}{Y_{11}})I_b(k+1) - \frac{h_{22}Y_{ab}}{Y_2}(\frac{h_{22}}{Y_{11}})I_b(k+1) - \frac{h_{22}Y_{ab}}{Y_{11}}(\frac{h_{22}}{Y_{11}})I_b(k+1) - \frac{h_{22}Y_{ab}}(\frac{h_{22}}{Y_{$$

$$\frac{h_{21}Y_{ab}}{Y_{11}} \frac{Y_{11}}{Y_{11}Y_{22} - h_{12}h_{21}} I_c(k+1)) \tag{A.74}$$

$$I_b(k+1)\left(1+\frac{Y_{ab}}{Y_{11}}+\frac{Y_{ab}}{Y_2}\right) = \frac{Y_{ab}}{Y_2}I_a(k) + Y_{ab}V_a(k) - \frac{h_{21}Y_{ab}}{Y_{11}}\left(\frac{h_{12}}{Y_{11}Y_{22}-h_{12}h_{21}}\right)I_b(k+1) - \frac{h_{21}Y_{ab}}{Y_2}I_a(k) - \frac{h_{21}Y_{ab}}{Y_2}I_a(k)$$

$$\frac{h_{21}Y_{ab}}{Y_{11}} \frac{Y_{11}}{Y_{11}Y_{22} - h_{12}h_{21}} I_c(k+1)) \tag{A.75}$$

$$I_b(k+1)\left(1+\frac{Y_{ab}}{Y_{11}}+\frac{Y_{ab}}{Y_2}\right) = \frac{Y_{ab}}{Y_2}I_a(k) + Y_{ab}\left(\frac{I_1-I_a(k)}{Y_a}\right) - \frac{h_{21}Y_{ab}}{Y_{11}}\left(\frac{h_{12}}{Y_{11}Y_{22}-h_{12}h_{21}}\right)I_b(k+k)$$

$$1) - \frac{h_{21}Y_{ab}}{Y_{11}} \frac{Y_{11}}{Y_{11}Y_{22} - h_{12}h_{21}} I_c(k+1))$$
(A.76)

$$I_b(k+1)\left(1+\frac{Y_{ab}}{Y_{11}}+\frac{Y_{ab}}{Y_2}\right) = \left(\frac{Y_{ab}}{Y_2}-\frac{Y_{ab}}{Y_a}\right)I_a(k) + Y_{ab}\frac{I_1}{Y_a} - \frac{h_{21}Y_{ab}}{Y_{11}}\left(\frac{h_{12}}{Y_{11}Y_{22}-h_{12}h_{21}}\right)I_b(k+1)$$

$$1) - \frac{h_{21}Y_{ab}}{Y_{11}} \frac{Y_{11}}{Y_{11}Y_{22} - h_{12}h_{21}} I_c(k+1))$$
(A.77)

$$I_b(k+1)\left(1+\frac{Y_{ab}}{Y_{11}}+\frac{Y_{ab}}{Y_2}+\frac{h_{21}Y_{ab}}{Y_{11}}\left(\frac{h_{12}}{Y_{11}Y_{22}-h_{12}h_{21}}\right)\right) = \left(\frac{Y_{ab}}{Y_2}-\frac{Y_{ab}}{Y_a}\right)I_a(k) + Y_{ab}\frac{I_1}{Y_a} - \frac{I_1}{Y_a}$$

$$\frac{h_{21}Y_{ab}}{Y_{11}}\frac{Y_{11}}{Y_{11}Y_{22}-h_{12}h_{21}}I_c(k+1)) \tag{A.78}$$

$$I_{b}(k+1)\left(1+\frac{Y_{ab}}{Y_{11}}+\frac{Y_{ab}}{Y_{2}}+\frac{Y_{22}Y_{ab}}{Y_{11}Y_{22}-h_{12}h_{21}}-\frac{Y_{ab}}{Y_{11}}\right) = \left(\frac{Y_{ab}}{Y_{2}}-\frac{Y_{ab}}{Y_{a}}\right)I_{a}(k) + Y_{ab}\frac{I_{1}}{Y_{a}} - \frac{h_{21}Y_{ab}}{Y_{11}}\frac{Y_{11}}{Y_{11}Y_{22}-h_{12}h_{21}}I_{c}(k+1))$$
(A.79)

$$I_{b}(k+1)\left(1+\frac{Y_{ab}}{Y_{2}}+\frac{Y_{22}Y_{ab}}{Y_{11}Y_{22}-h_{12}h_{21}}\right) = \left(\frac{Y_{ab}}{Y_{2}}-\frac{Y_{ab}}{Y_{a}}\right)I_{a}(k) + Y_{ab}\frac{I_{1}}{Y_{a}} -$$

 $\frac{h_{21}Y_{ab}}{Y_{11}} \frac{Y_{11}}{Y_{11}Y_{22} - h_{12}h_{21}} I_c(k+1)) \tag{A.80}$ 

$$I_{b}(k+1)\left(1+\frac{Y_{ab}}{Y_{2}}+\frac{Y_{22}Y_{ab}}{Y_{11}Y_{22}-h_{12}h_{21}}\right) = \left(\frac{Y_{ab}}{Y_{2}}-\frac{Y_{ab}}{Y_{a}}\right)I_{a}(k) + Y_{ab}\frac{I_{1}}{Y_{a}} - \frac{h_{21}Y_{ab}}{Y_{11}}\frac{Y_{11}}{Y_{12}-h_{12}h_{21}}I_{c}(k+1))$$

$$I_{b}(k+1)\left(1+\frac{Y_{ab}}{Y_{2}}+\frac{Y_{22}Y_{ab}}{Y_{11}Y_{22}-h_{12}h_{21}}\right) = \left(\frac{Y_{ab}}{Y_{2}}-\frac{Y_{ab}}{Y_{a}}\right)I_{a}(k) + Y_{ab}\frac{I_{1}}{Y_{a}} - \frac{h_{21}Y_{ab}}{Y_{11}}\frac{Y_{11}}{Y_{11}Y_{22}-h_{12}h_{21}}I_{c}(k+1))$$

$$I_{b}(k+1)\left(1+\frac{Y_{ab}}{Y_{2}}+\frac{Y_{22}Y_{ab}}{Y_{11}Y_{22}-h_{12}h_{21}}\right) = \left(\frac{Y_{ab}}{Y_{2}}-\frac{Y_{ab}}{Y_{a}}\right)I_{a}(k) + Y_{ab}\frac{I_{1}}{Y_{a}} - \frac{h_{21}Y_{ab}}{Y_{11}Y_{22}-h_{12}h_{21}}I_{c}(k+1))$$

$$I_{b}(k+1)\left(1+\frac{Y_{ab}}{Y_{2}}+\frac{Y_{22}Y_{ab}}{Y_{11}Y_{22}-h_{12}h_{21}}\right) = \left(\frac{Y_{ab}}{Y_{2}}-\frac{Y_{ab}}{Y_{a}}\right)I_{a}(k) + Y_{ab}\frac{I_{1}}{Y_{a}} - \frac{h_{21}Y_{ab}}{Y_{11}Y_{22}-h_{12}h_{21}}I_{c}(k+1)$$

$$I_{b}(k+1)\left(\frac{1}{Y_{ab}} + \frac{1}{Y_{2}} + \frac{Y_{22}}{Y_{11}Y_{22} - h_{12}h_{21}}\right) = \left(\frac{1}{Y_{2}} - \frac{1}{Y_{a}}\right)I_{a}(k) + \frac{I_{1}}{Y_{a}} - \frac{h_{21}}{Y_{11}Y_{22} - h_{12}h_{21}}I_{c}(k+1)$$
(A.84)

 $I_c(k+1) = (V_2(k+1) - V_{d3}(k+1))Y_{bc}$ (A.85)

$$I_c(k+1) = Y_{bc}V_2(k+1) - Y_{bc}V_{d3}(k+1)$$
(A.86)

$$I_{c}(k+1) = Y_{bc}\left(\frac{h_{12}}{Y_{11}Y_{22} - h_{12}h_{21}}I_{b}(k+1) - \frac{Y_{11}}{Y_{11}Y_{22} - h_{12}h_{21}}I_{c}(k+1)\right) -$$

$$Y_{bc}\left(\frac{S_{3}(k+1)+I_{c}(k+1)}{Y_{3}}\right)$$

$$I_{c}(k+1) = \frac{Y_{bc}h_{12}}{Y_{11}Y_{22}-h_{12}h_{21}}I_{b}(k+1) - \frac{Y_{bc}Y_{11}}{Y_{11}Y_{22}-h_{12}h_{21}}I_{c}(k+1) - \frac{Y_{bc}}{Y_{3}}S_{3}(k+1) -$$

$$Y_{bc} = 1$$

$$I_{c}(k+1)(1 + \frac{Y_{bc}Y_{11}}{Y_{11}Y_{22} - h_{12}h_{21}} + \frac{Y_{bc}}{Y_{3}}) = \frac{Y_{bc}h_{12}}{Y_{11}Y_{22} - h_{12}h_{21}}I_{b}(k+1) - \frac{Y_{bc}}{Y_{3}}S_{3}(k+1)$$

$$I_{c}(k+1)\left(1 + \frac{Y_{bc}Y_{11}}{Y_{11}Y_{22} - h_{12}h_{21}} + \frac{Y_{bc}}{Y_{3}}\right) = \frac{Y_{bc}h_{12}}{Y_{11}Y_{22} - h_{12}h_{21}}I_{b}(k+1) - \frac{Y_{bc}}{Y_{3}}(V_{c}(k)Y_{3} - I_{d}(k))$$
(A.90)

$$I_{c}(k+1)\left(1+\frac{Y_{bc}Y_{11}}{Y_{11}Y_{22}-h_{12}h_{21}}+\frac{Y_{bc}}{Y_{3}}\right) = \frac{Y_{bc}h_{12}}{Y_{11}Y_{22}-h_{12}h_{21}}I_{b}(k+1) - \frac{Y_{bc}}{Y_{3}}V_{c}(k)Y_{3} + \frac{Y_{bc}}{Y_{3}}I_{d}(k))$$
(A.91)

$$I_{c}(k+1)\left(1 + \frac{Y_{bc}Y_{11}}{Y_{11}Y_{22} - h_{12}h_{21}} + \frac{Y_{bc}}{Y_{3}}\right) = \frac{Y_{bc}h_{12}}{Y_{11}Y_{22} - h_{12}h_{21}}I_{b}(k+1) - Y_{bc}V_{c}(k) + \frac{Y_{bc}}{Y_{3}}I_{d}(k)$$
(A.92)

$$I_{c}(k+1)\left(1+\frac{Y_{bc}Y_{11}}{Y_{11}Y_{22}-h_{12}h_{21}}+\frac{Y_{bc}}{Y_{3}}\right) = \frac{Y_{bc}h_{12}}{Y_{11}Y_{22}-h_{12}h_{21}}I_{b}(k+1) - Y_{bc}\left(\frac{I_{3}+I_{d}(k)}{Y_{c}}\right) +$$

$$\frac{Y_{bc}}{Y_3}I_d(k) \tag{A.93}$$

$$I_{c}(k+1)\left(1+\frac{Y_{bc}Y_{11}}{Y_{11}Y_{22}-h_{12}h_{21}}+\frac{Y_{bc}}{Y_{3}}\right) = \frac{Y_{bc}h_{12}}{Y_{11}Y_{22}-h_{12}h_{21}}I_{b}(k+1) - Y_{bc}\frac{I_{3}}{Y_{c}} - Y_{bc}\frac{I_{d}(k)}{Y_{c}} + \frac{Y_{bc}}{Y_{3}}I_{d}(k)$$
(A.94)

$$I_{c}(k+1)\left(\frac{1}{Y_{bc}} + \frac{Y_{11}}{Y_{11}Y_{22} - h_{12}h_{21}} + \frac{1}{Y_{3}}\right) = \frac{h_{12}}{Y_{11}Y_{22} - h_{12}h_{21}}I_{b}(k+1) - \frac{I_{3}}{Y_{c}} - \frac{I_{d}(k)}{Y_{c}} + \frac{1}{Y_{c}}I_{b}(k+1) - \frac{I_{3}}{Y_{c}} - \frac{I_{d}(k)}{Y_{c}} + \frac{1}{Y_{c}}I_{b}(k+1) - \frac{I_{3}}{Y_{c}}I_{b}(k+1) - \frac{I_{3}}{Y_{c}}I_{b$$

$$\frac{1}{Y_3}I_d(k) \tag{A.95}$$

$$I_{c}(k+1)\left(\frac{1}{Y_{bc}} + \frac{Y_{11}}{Y_{11}Y_{22} - h_{12}h_{21}} + \frac{1}{Y_{3}}\right) = \frac{h_{12}}{Y_{11}Y_{22} - h_{12}h_{21}}I_{b}(k+1) - \frac{1}{Y_{c}} + I_{d}(k)(\frac{1}{Y_{3}} - \frac{1}{Y_{c}})$$

$$(A.96)$$

$$I_d(k+1) = (V_{d4}(k+1) - V_c(k+1))Y_{bc}$$
(A.97)

$$I_d(k+1) = Y_{bc}V_{d4}(k+1) - Y_{bc}V_c(k+1)$$
(A.98)

$$I_d(k+1) = Y_{bc}\left(\frac{S_4(k+1) - I_d(k+1)}{Y_4}\right) - Y_{bc}\left(\frac{I_3 + I_d(k+1)}{Y_c}\right)$$
(A.99)

$$I_d(k+1) = \frac{Y_{bc}S_4(k+1)}{Y_4} - \frac{Y_{bc}I_d(k+1)}{Y_4} - \frac{Y_{bc}I_3}{Y_c} - \frac{I_d(k+1)Y_{bc}}{Y_c}$$
(A.100)

$$I_d(k+1)(1+\frac{Y_{bc}}{Y_c}+\frac{Y_{bc}}{Y_4}) = \frac{Y_{bc}}{Y_4}S_4(k+1) - \frac{Y_{bc}I_3}{Y_c}$$
(A.101)

$$I_d(k+1)(\frac{1}{Y_{bc}} + \frac{1}{Y_c} + \frac{1}{Y_4}) = \frac{1}{Y_4}S_4(k+1) - \frac{I_3}{Y_c}$$
(A.102)

$$I_d(k+1)(\frac{1}{Y_{bc}} + \frac{1}{Y_c} + \frac{1}{Y_4}) = \frac{1}{Y_4}((\frac{h_{12}V_1(k) - I_c(k)}{Y_{22}})Y_4 + I_c(k)) - \frac{I_3}{Y_c}$$
(A.103)

$$I_d(k+1)\left(\frac{1}{Y_{bc}} + \frac{1}{Y_c} + \frac{1}{Y_4}\right) = \frac{h_{12}}{Y_{22}}V_1(k) - \frac{1}{Y_{22}}I_c(k) + \frac{1}{Y_4}I_c(k) - \frac{I_3}{Y_c}$$
(A.104)

$$V_1(k) = \frac{h_{21}V_2(k) + I_b(k)}{Y_{11}}, V_2(k) = \frac{h_{12}V_1(k) - I_c(k)}{Y_{22}}$$
(A.105)

$$V_1(k) = \frac{h_{21}V_2(k) + I_b(k)}{Y_{11}}$$
(A.106)

$$V_1(k) = \frac{h_{21}}{Y_{11}} V_2(k) + \frac{I_b(k)}{Y_{11}}$$
(A.107)

$$V_1(k) = \frac{h_{21}}{Y_{11}} \left( \frac{h_{12}V_1(k) - I_c(k)}{Y_{22}} \right) + \frac{I_b(k)}{Y_{11}}$$
(A.108)

$$V_1(k) = \frac{h_{21}}{Y_{11}} \left( \frac{h_{12}V_1(k)}{Y_{22}} - \frac{I_c(k)}{Y_{22}} \right) + \frac{I_b(k)}{Y_{11}}$$
(A.109)

$$V_1(k) = \frac{h_{12}h_{21}}{Y_{22}Y_{11}}V_1(k) - \frac{h_{21}}{Y_{11}Y_{22}}I_c(k) + \frac{I_b(k)}{Y_{11}}$$
(A.110)

$$V_1(k)(1 - \frac{h_{12}h_{21}}{Y_{22}Y_{11}}) = -\frac{h_{21}}{Y_{11}Y_{22}}I_c(k) + \frac{I_b(k)}{Y_{11}}$$
(A.111)

$$V_1(k) = -\frac{\frac{h_{21}}{Y_{11}Y_{22}}}{(1 - \frac{h_{12}h_{21}}{Y_{22}Y_{11}})} I_c(k) + \frac{\frac{1}{Y_{11}}}{(1 - \frac{h_{12}h_{21}}{Y_{22}Y_{11}})} I_b(k)$$
(A.112)

$$V_1(k) = \frac{Y_{22}}{Y_{11}Y_{22} - h_{12}h_{21}} I_b(k) - \frac{h_{21}}{Y_{11}Y_{22} - h_{12}h_{21}} I_c(k)$$
(A.113)

$$I_d(k+1)\left(\frac{1}{Y_{bc}} + \frac{1}{Y_c} + \frac{1}{Y_4}\right) = \frac{h_{12}}{Y_{22}}V_1(k) - \frac{1}{Y_{22}}I_c(k) + \frac{1}{Y_4}I_c(k) - \frac{I_3}{Y_c}$$
(A.114)

$$I_d(k+1)\left(\frac{1}{Y_{bc}} + \frac{1}{Y_c} + \frac{1}{Y_4}\right) = \frac{h_{12}}{Y_{22}}\left(\frac{Y_{22}}{Y_{11}Y_{22} - h_{12}h_{21}}I_b(k) - \frac{h_{21}}{Y_{11}Y_{22} - h_{12}h_{21}}I_c(k)\right) + \left(\frac{1}{Y_4} - \frac{h_{22}}{Y_4}\right)$$

$$\frac{1}{Y_{22}}I_c(k) - \frac{I_3}{Y_c}$$
(A.115)  
$$I_d(k+1)\left(\frac{1}{Y_c} + \frac{1}{Y} + \frac{1}{Y}\right) = \frac{h_{12}}{Y_c - h_c - h_c}I_b(k) - \frac{h_{12}h_{21}}{Y_c - (Y_c - Y_c - h_c - h_c)}I_c(k) + \left(\frac{1}{Y_c} - \frac{h_{12}h_{21}}{Y_c - h_c - h_c}\right)I_c(k) + \frac{1}{Y_c - h_c}I_c(k) + \frac{1}{Y_c}I_c(k) + \frac{1}$$

$$\frac{1}{Y_{22}}I_c(k) - \frac{I_3}{Y_c}$$
(A.116)

$$I_{d}(k+1)\left(\frac{1}{Y_{bc}} + \frac{1}{Y_{c}} + \frac{1}{Y_{4}}\right) = \frac{h_{12}}{Y_{11}Y_{22} - h_{12}h_{21}}I_{b}(k) + \left(\frac{1}{Y_{4}} - \frac{1}{Y_{22}} - \frac{h_{12}h_{21}}{Y_{22}(Y_{11}Y_{22} - h_{12}h_{21})}\right)I_{c}(k) - \frac{I_{3}}{Y_{c}}$$
(A.117)

If MPE is set correctly, then the terms become 0 plus residual which is completely dependent on the electrical coupling of the subsystems.

$$I_{d}(k+1) = \frac{\frac{h_{12}}{Y_{11}Y_{22} - h_{12}h_{21}}}{\left(\frac{1}{Y_{bc}} + \frac{1}{Y_{c}} + \frac{1}{Y_{d}}\right)} I_{b}(k) + \frac{\left(\frac{1}{Y_{4}} - \frac{1}{Y_{22}} - \frac{h_{12}h_{21}}{Y_{22}(Y_{11}Y_{22} - h_{12}h_{21})}\right)}{\left(\frac{1}{Y_{bc}} + \frac{1}{Y_{c}} + \frac{1}{Y_{d}}\right)} I_{c}(k) - \frac{\frac{I_{3}}{Y_{c}}}{\left(\frac{1}{Y_{bc}} + \frac{1}{Y_{c}} + \frac{1}{Y_{d}}\right)}$$

$$(A.118)$$

$$I_{c}(k+1)\left(\frac{1}{Y_{bc}} + \frac{Y_{11}}{Y_{11}Y_{22} - h_{12}h_{21}} + \frac{1}{Y_{3}}\right) = \frac{h_{12}}{Y_{11}Y_{22} - h_{12}h_{21}} I_{b}(k+1) - \frac{I_{3}}{Y_{c}} + I_{d}(k)(\frac{1}{Y_{3}} - \frac{1}{Y_{c}})$$

$$(A.119)$$

$$I_{b}(k+1)\left(\frac{1}{Y_{bc}} + \frac{1}{Y_{c}} + \frac{Y_{22}}{Y_{c}} + \frac{Y_{22}}{Y_{c}}\right) = (\frac{1}{Y_{c}} - \frac{1}{Y_{c}})I_{a}(k) + \frac{I_{1}}{Y_{c}} - \frac{h_{21}}{Y_{c}} + I_{c}(k+1)$$

$$I_{b}(k+1)\left(\frac{1}{Y_{ab}} + \frac{1}{Y_{2}} + \frac{Y_{22}}{Y_{11}Y_{22} - h_{12}h_{21}}\right) = \left(\frac{1}{Y_{2}} - \frac{1}{Y_{a}}\right)I_{a}(k) + \frac{I_{1}}{Y_{a}} - \frac{h_{21}}{Y_{11}Y_{22} - h_{12}h_{21}}I_{c}(k+1)$$
(A.120)

$$x = \left(\frac{1}{Y_{bc}} + \frac{Y_{11}}{Y_{11}Y_{22} - h_{12}h_{21}} + \frac{1}{Y_3}\right)$$
(A.121)

$$y = \frac{h_{12}}{Y_{11}Y_{22} - h_{12}h_{21}} \tag{A.122}$$

$$z = \left(\frac{1}{Y_3} - \frac{1}{Y_c}\right)$$
(A.123)

$$q = \frac{I_3}{Y_c} \tag{A.124}$$

$$a = \left(\frac{1}{Y_{ab}} + \frac{1}{Y_2} + \frac{Y_{22}}{Y_{11}Y_{22} - h_{12}h_{21}}\right)$$
(A.125)

$$b = \left(\frac{1}{Y_2} - \frac{1}{Y_a}\right)$$
(A.126)

$$d = \frac{I_1}{Y_a} \tag{A.127}$$

$$e = \frac{h_{21}}{Y_{11}Y_{22} - h_{12}h_{21}} \tag{A.128}$$

$$I_c(k+1)x = yI_b(k+1) - q + I_d(k)z$$
 (A.129)

$$I_b(k+1)a = bI_a(k) + d - eI_c(k+1)$$
(A.130)

$$I_b(k+1)a = bI_a(k) + d - eyI_b(k+1) - eq + I_d(k)ze$$
(A.131)

$$I_c(k+1)x = ybI_a(k) + dy - eyI_c(k+1) - q + I_d(k)z$$
(A.132)

$$I_b(k+1)(a+ey) = bI_a(k) + d - eq + I_d(k)ze$$
(A.133)

$$I_c(k+1)(x+ey) = ybI_a(k) + dy - q + I_d(k)z$$
(A.134)

$$I_b(k+1)(a+ey) = bI_a(k) + I_d(k)ze + d - eq$$
(A.135)

$$I_c(k+1)(x+ey) = ybI_a(k) + I_d(k)z + dy - q$$
 (A.136)

$$I_b(k+1)\left(\left(\frac{1}{Y_{ab}} + \frac{1}{Y_2} + \frac{Y_{22}}{Y_{11}Y_{22} - h_{12}h_{21}}\right) + \left(\frac{h_{21}}{Y_{11}Y_{22} - h_{12}h_{21}}\right)\left(\frac{h_{12}}{Y_{11}Y_{22} - h_{12}h_{21}}\right)\right) = \left(\frac{1}{Y_2} - \frac{1}{Y_a}\right)I_a(k) + \frac{h_{12}}{Y_{11}Y_{22} - h_{12}h_{21}} = \frac{h_{12}}{Y_1} + \frac{h_{12}}{Y_1$$

$$I_d(k)(\frac{1}{Y_3} - \frac{1}{Y_c})(\frac{h_{21}}{Y_{11}Y_{22} - h_{12}h_{21}}) + \frac{I_1}{Y_a} - (\frac{h_{21}}{Y_{11}Y_{22} - h_{12}h_{21}})(\frac{I_3}{Y_c})$$
(A.137)

$$I_{c}(k+1)\left(\left(\frac{1}{Y_{bc}}+\frac{Y_{11}}{Y_{11}Y_{22}-h_{12}h_{21}}+\frac{1}{Y_{3}}\right)+\left(\frac{h_{21}}{Y_{11}Y_{22}-h_{12}h_{21}}\right)\left(\frac{h_{12}}{Y_{11}Y_{22}-h_{12}h_{21}}\right)\right)=$$

$$\left(\frac{h_{12}}{Y_{11}Y_{22}-h_{12}h_{21}}\right)\left(\frac{1}{Y_2}-\frac{1}{Y_a}\right)I_a(k)+I_d(k)\left(\frac{1}{Y_3}-\frac{1}{Y_c}\right)+\left(\frac{I_1}{Y_a}\right)\left(\frac{h_{12}}{Y_{11}Y_{22}-h_{12}h_{21}}\right)-\frac{I_3}{Y_c}$$
(A.138)

$$I_{b}(k+1)\left(\left(\frac{1}{Y_{ab}}+\frac{1}{Y_{2}}+\frac{Y_{22}}{Y_{11}Y_{22}-h_{12}h_{21}}\right)+\left(\frac{h_{21}h_{12}}{(Y_{11}Y_{22}-h_{12}h_{21})^{2}}\right)\right) = \left(\frac{1}{Y_{2}}-\frac{1}{Y_{a}}\right)I_{a}(k)+I_{d}(k)\left(\frac{1}{Y_{3}}-\frac{1}{Y_{3}}\right)\left(\frac{h_{21}}{Y_{11}Y_{22}-h_{12}h_{21}}\right)+\frac{I_{1}}{Y_{a}}-\left(\frac{h_{21}}{Y_{11}Y_{22}-h_{12}h_{21}}\right)\left(\frac{I_{3}}{Y_{c}}\right)(A.139)$$

$$I_{c}(k+1)\left(\left(\frac{1}{Y_{bc}}+\frac{Y_{11}}{Y_{11}Y_{22}-h_{12}h_{21}}+\frac{1}{Y_{3}}\right)+\left(\frac{h_{21}h_{12}}{(Y_{11}Y_{22}-h_{12}h_{21})^{2}}\right)\right) = \left(\frac{h_{12}}{Y_{11}Y_{22}-h_{12}h_{21}}\right)\left(\frac{1}{Y_{2}}-\frac{1}{Y_{2}}\right)I_{a}(k)+I_{d}(k)\left(\frac{1}{Y_{3}}-\frac{1}{Y_{c}}\right)+\left(\frac{I_{1}}{Y_{a}}\right)\left(\frac{h_{12}}{Y_{11}Y_{22}-h_{12}h_{21}}\right)-\frac{I_{3}}{Y_{c}}$$
(A.140)

$$I_{b}(k+1) = \frac{(\frac{1}{Y_{2}} - \frac{1}{Y_{a}})}{(\frac{1}{Y_{ab}} + \frac{1}{Y_{2}} + \frac{Y_{22}}{Y_{11}Y_{22} - h_{12}h_{21}} + \frac{h_{21}h_{12}}{(Y_{11}Y_{22} - h_{12}h_{21})^{2}})}I_{a}(k) + I_{d}(k) \frac{(\frac{1}{Y_{3}} - \frac{1}{Y_{c}})(\frac{h_{21}}{Y_{11}Y_{22} - h_{12}h_{21}})}{(\frac{1}{Y_{ab}} + \frac{1}{Y_{2}} + \frac{Y_{22}}{Y_{11}Y_{22} - h_{12}h_{21}} + \frac{h_{21}h_{12}}{(Y_{11}Y_{22} - h_{12}h_{21})^{2}})}) + C$$

$$I_{c}(k+1) = \frac{(\frac{h_{12}}{Y_{11}Y_{22} - h_{12}h_{21}})(\frac{1}{Y_{2}} - \frac{1}{Y_{a}})}{\frac{1}{Y_{bc}} + \frac{Y_{11}}{Y_{11}Y_{22} - h_{12}h_{21}} + \frac{1}{Y_{3}} + \frac{h_{21}h_{12}}{(Y_{11}Y_{22} - h_{12}h_{21})^{2}}}I_{a}(k) + I_{d}(k) \frac{(\frac{1}{Y_{3}} - \frac{1}{Y_{c}})}{\frac{1}{Y_{bc}} + \frac{Y_{11}}{Y_{11}} + \frac{1}{Y_{3}} + \frac{h_{21}h_{12}}{(Y_{11}Y_{22} - h_{12}h_{21})^{2}}} + C$$
(A.142)

The above derivations can be combined to conform to the standard representation in literature of the form Equation 1.7.

$$\begin{bmatrix} I_{a}(k+1)\\ I_{b}(k+1)\\ I_{c}(k+1)\\ I_{d}(k+1) \end{bmatrix} = \begin{bmatrix} 0 & \alpha & \beta & 0\\ \gamma & 0 & 0 & \delta\\ \varepsilon & 0 & 0 & \eta\\ 0 & \theta & \kappa & 0 \end{bmatrix} \begin{bmatrix} I_{a}(k)\\ I_{b}(k)\\ I_{b}(k)\\ I_{b}(k) \end{bmatrix} + C$$
(A.143)

$$\alpha = \frac{\left(\frac{1}{Y_1} - \frac{1}{Y_{11}} - \frac{h_{12}h_{21}}{Y_{11}(Y_{11}Y_{22} - h_{12}h_{21})}\right)}{\left(\frac{1}{Y_{ab}} + \frac{1}{Y_a} + \frac{1}{Y_1}\right)}$$
(A.144)

$$\beta = \frac{\frac{h_{21}}{Y_{11}Y_{22} - h_{12}h_{21}}}{\left(\frac{1}{Y_{ab}} + \frac{1}{Y_{a}} + \frac{1}{Y_{1}}\right)}$$
(A.145)

$$\gamma = \frac{(\frac{1}{Y_2} - \frac{1}{Y_a})}{(\frac{1}{Y_{ab}} + \frac{1}{Y_2} + \frac{Y_{22}}{Y_{11}Y_{22} - h_{12}h_{21}} + \frac{h_{21}h_{12}}{(Y_{11}Y_{22} - h_{12}h_{21})^2})}$$
(A.146)

$$\delta = \frac{(\frac{1}{Y_3} - \frac{1}{Y_c})(\frac{h_{21}}{Y_{11}Y_{22} - h_{12}h_{21}})}{(\frac{1}{Y_{ab}} + \frac{1}{Y_2} + \frac{Y_{22}}{Y_{11}Y_{22} - h_{12}h_{21}} + \frac{h_{21}h_{12}}{(Y_{11}Y_{22} - h_{12}h_{21})^2})}$$
(A.147)

$$\varepsilon = \frac{\left(\frac{h_{12}}{Y_{11}Y_{22} - h_{12}h_{21}}\right)\left(\frac{1}{Y_2} - \frac{1}{Y_a}\right)}{\frac{1}{Y_{bc}} + \frac{Y_{11}}{Y_{11}Y_{22} - h_{12}h_{21}} + \frac{1}{Y_3} + \frac{h_{21}h_{12}}{(Y_{11}Y_{22} - h_{12}h_{21})^2}}$$
(A.148)

$$\eta = \frac{(\frac{1}{Y_3} - \frac{1}{Y_c})}{\frac{1}{Y_{bc}} + \frac{Y_{11}}{Y_{11}Y_{22} - h_{12}h_{21}} + \frac{1}{Y_3} + \frac{h_{21}h_{12}}{(Y_{11}Y_{22} - h_{12}h_{21})^2}}$$
(A.149)

$$\theta = \frac{\frac{h_{12}}{Y_{11}Y_{22} - h_{12}h_{21}}}{\left(\frac{1}{Y_{bc}} + \frac{1}{Y_{c}} + \frac{1}{Y_{4}}\right)}$$
(A.150)

$$\kappa = \frac{\left(\frac{1}{Y_4} - \frac{1}{Y_{22}} - \frac{h_{12}h_{21}}{Y_{22}(Y_{11}Y_{22} - h_{12}h_{21})}\right)}{\left(\frac{1}{Y_{hc}} + \frac{1}{Y_c} + \frac{1}{Y_4}\right)}$$
(A.151)

## A.3 Explicit W Validation of the Three Partition Example

The explicit **W** matrix derived in A.2 can be validated for the 1-2-1 case. The system in Fig A.4 is used, to avoid instant convergence the dummy admittance parameters are set off ideal to force a non-zero spectral radius. For simplicity non-complex values are used but the theory expands to complex values as well.



Figure A.4 1-2-1 System Verifying Explicit W

The system in Fig A.4 can be represented in the equivalent form of the appendix

figure A.2 and A.3 and the parameters can be derived as:

Parameter	Value
Y <sub>11</sub>	6.0143
Y <sub>22</sub>	6.0143
Y <sub>1</sub>	6.3150
Y <sub>2</sub>	2.6250
Y <sub>3</sub>	3
Y4	6.3150
h <sub>12</sub>	5.0969
h <sub>21</sub>	5.0969
Ya	2.5
Y <sub>ab</sub>	10
Y <sub>bc</sub>	10
Y <sub>c</sub>	

Table A.1Parameters for W matrix verification

Through linear regression the W matrix is found within a 95% confidence bound.

The regression confidence bound explains the non-zero entries in the regression.

	-0.0061	0.7562	-0.6488	-0.0044
(A.152)	-0.0080	0.0005	-0.0009	-0.0217
	-0.0177	-0.0005	0.0013	-0.0086
	0.0125	-0.7058	0.8145	0.0091

From the explicit derivation the equivalent W is:

	0.0000	0.7595	-0.6557	0.0000
(A.153)	-0.0063	0.0000	0.0000	-0.0144
	-0.0131	0.0000	0.0000	-0.0075
	0.0000	-0.7096	0.8219	0.0000

This calculation verifies the explicit form of **W** derived for this case. It is assumed that, as the system complexity and number of partitions grow, there will be a similar form. The trend is expected to continue of many zero entries with some non zero residual entries in the W matrix. These residual entries are a direct reflection of the electrical coupling between multiple port dependencies. If there is a case where the electrical coupling is strong between the ports of a two or more port system, then the performance is expected to degrade, even with the additional help of the Multiport Equivalent information to set the boundary admittance.