Parallel 2-D Finite Element Modeling for Transmission Line and Power Transformer for Electromagnetic Transient Study

by

Qingjie Xu

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science
in
Energy Systems

Department of Electrical and Computer Engineering
University of Alberta

Abstract

The energy exchange between the electric field and the magnetic field is the foundation for electromagnetic equipment in power systems such as transformers and transmission lines. However, not all exchange is desirable. Some sudden exchanges with a large amount of energy involved such as overvoltages and overcurrents may have an adverse impact on the equipment's operation and reliability. The electromagnetic transient (EMT) simulation was traditionally utilized to study the effects of these troublesome energy exchanges. Lumped models such as admittance matrix-based models have been widely used in EMT simulation, however, conventional lumped models are incapable of capturing transient behaviors accurately especially for devices with nonlinear materials and frequency-dependent effects such as eddy currents and hysteresis. As the most prevalent numerical method to solve field-oriented Maxwell's equations to describe the electromagnetic problem accurately, the finite element method (FEM) has attracted increasing attention in EMT simulation. FEM is a powerful tool to provide the capability of modeling irregular geometries, the ability of handling complex material properties, and superior accuracy.

Regardless of the outstanding accuracy and detailed insight provided, the finite element method leads to a significant increase in computational burden for a simulation program. The essential point of the FEM is to discretize a problem domain into many interconnected nodes and elements wherein the value of one field quantity is associated with nodes. Then elemental equations within each element and a global matrix equation built by assembling all elemental equations are generated. With the solution of this global matrix equation, the field quantity distribution within the problem domain is also obtained. On the other hand, the repetitive computation of composing and factorizing a large matrix system is involved to solve nonlinear problems because of the traditional Newton-Raphson method. The fast advancement of parallel computing with high-performance computing hardware provides a path forward. The parallel computing hardware including graphics processing units (GPUs) and field-programmable gate arrays (FPGAs) provides the inherent parallel architecture to execute a program in parallel and thus has the potential to shorten execution time. To accelerate FEM computation with parallel computing hardware, an algorithm design to modify the traditional FEM process to fit in parallelism is required. In this thesis, parallel algorithms are designed to solve EMT in both ionized field and power transformer efficiently on the appropriate parallel computing platform.

First, a massively parallel algorithm to solve a hybrid ionized field around AC/DC transmission lines is proposed. A fine-grained nodal domain decomposition scheme, which enables each sub-domain with only one unknown to be solved independently in a massively parallel fashion, was employed to solve Poisson's equation. Meanwhile, an upwind nodal charge conservation method is applied to solve the current continuity equation without numerical oscillation at each finite element nodal level. The computation of nodal domain decomposition and upwind nodal charge conservation can both be vectorized and mapped to massive computational cores and utilize the computing power of GPUs. The interaction between HVAC and HVDC was solved without Deutsch's assumption to guarantee the accuracy, and the wind influence can be considered. With the node-level massively parallelism of the proposed algorithm, both Poisson's equation and the current continuity equation were solved at each time-step on GPUs to obtain the transient details of the hybrid ionized field. The performance of the proposed method is tested and compared with commercial software showing a significant speedup with guaranteed accuracy.

Second, the transmission-line modeling (TLM) technique is integrated into a parallel and deeply pipelined algorithm to decouple the nonlinear finite elements caused by nonlinear material in a power transformer. The transmission line is utilized to separate the nonlinear finite elements from the linear network and then these decoupled elements can be solved individually in a parallel manner. Without losing the merit of traditional transmission-line modeling, an adaptive transmission-line modeling method is employed to reduce TLM iteration number. The other component of this algorithm is the preconditioned conjugate gradient method, which can be deployed on FPGA to achieve high parallelism. The accuracy of the transformer solver under both current-excited and voltage-excited conditions was validated against the commercial FE simulation tool. In addition, a field-circuit coupling approach to interface the FE model for the transformer and external circuits is also discussed and implemented on FPGAs. The accuracy for the combination between the field-circuit coupling and the above finite element model is also verified.

Preface

The contents of this thesis are based on original intellectual work innovated by Qingjie Xu. Both the algorithm development and hardware implementation present in this thesis were conducted in the Real-Time Experimental Laboratory (RTX-Lab) at the University of Alberta. Qingjie Xu completed primary work in the areas of algorithm design, hardware implementation, case study, and manuscript composition under the supervision of Dr. Venkata Dinavahi. Peng Liu participated in discussions about the concept visualization and algorithm formation while providing constructive suggestions. As described below, some parts of the thesis have been published or submitted as journal articles.

Chapter 3 includes contents published in the following paper:

- Q. Xu, P. Liu and V. Dinavahi, ”Parallel finite element computation of time-varying ionized field around hybrid AC/DC Lines via fine-grained domain decomposition”, *IEEE Access*, vol. 8, pp. 91248-91256, 2020.

Chapter 4 contains contents from the following paper that is currently under review:

- Q. Xu, P. Liu and V. Dinavahi, ”Time-stepped finite-element modeling of three-phase transformer for electromagnetic transient emulation on FPGA”, *under review, IEEE Open Access Journal of Power and Energy*, pp. 1-10, March 2021.

To my parents,
for their love and endless support.

Acknowledgements

I would like to express my sincere gratitude to my supervisor *Prof. Venkata Dinavahi* for his encouragement, full support, and academic guidance through my study at the University of Alberta. His passion and enthusiasm towards research have become a great motivation for my life.

I also would like to deliver special thanks to my colleagues and friends in the RTX-lab including Peng Liu, Jiacong Li, Tong Duan, Tian Liang, Tianshi Cheng. They are all knowledgeable and willing to provide valuable suggestions.

I want to express my deepest thanks to my parents for their understanding, encouragement, and unconditional support all the time.

Table of Contents

List of Tables

List of Figures

List of Acronyms

AC	Alternating Current
API	Application Programming Interface
ASIC	Application-Specific Integrated Circuit
CLB	Configurable Logic Block
CG	Conjugate Gradient
CUDA	Compute Unified Device Architecture
DC	Direct Current
DFF	D-type Flip Flop
DoF	Degree of Freedom
DSP	Digital Signal Processing block
EMF	Electromotive Force
EMT	Electromagnetic Transient
FE	Finite Element
FEM	Finite Element Method
FIFO	First-In-First-Out
FPGA	Field-Programmable Gate Array
GPU	Graphics Processing Unit
HDL	Hardware Description Language
HLS	High-Level Synthesis
HVAC	High-Voltage Alternating Current
HVDC	High-Voltage Direct Current
IOB	Input/Output Bank
LU	Lower-Upper
LUT	Look-Up Table
NCC	Nodal Charge Conservation
NDD	Nodal Domain Decomposition
PCG	Preconditioned Conjugate Gradient
RAM	Random-Access Memory
RHS	Right-Hand Side
ROM	Read-Only Memory
SIMD	Single Instruction Multiple Data
SM	Streaming Multiprocessor
TLM	Transmission-Line Modeling

1
Introduction

Electromagnetic transient (EMT) study has become an essential step for the design of efficient power systems, planning of power grids, performance evaluation of power devices, and many other situations. As the name states, EMT primarily relates to the energy exchange between the electric fields of capacitances and the magnetic fields of inductances in a system while electromechanical transients, as another type of transient in power systems, is caused by the interaction between the electrical energy in the power grid and the mechanical energy in a rotating machine [1]. However, when a huge amount of energy transfer between two fields happens in a quite short time, an overvoltage or overcurrent may be produced by EMTs. This large amplitude on voltage or current generally lasts for a very short time but the impact on the power system and the power devices may be catastrophic. The typical phenomena in EMTs contain lightning strokes on transmission lines, energization of transformer-terminated lines, load rejection, and input surge current in transformers and the result of EMTs might be destroyed component insulation, activated control systems, or interrupted system [2]. EMTs can be caused by various types of disturbances, including the switching actions between devices, the lightning strike, and the short circuit in the system meanwhile the power systems are always subject to these disturbances. Thus, it is important to analyze and investigate the detailed process of the EMTs to diminish its impact on the power system and equipment. EMTs can happen at any chunk of the power system and the EMTs for transmission lines and power transformers have been investigated as two portions in this work.

This chapter will introduce the background first and then the motivation carrying out the research topic of efficient parallel computing algorithm design for finite element models. For the ionized field problem, the importance of analyzing ionized fields, the built-up process of ionized field, and the discussion about existed methods and finite element method (FEM) are presented. The importance of the transformer and the summary of transformer models are also explained. A brief introduction of the general process of FEM is discussed later. At last, the motivation, objectives of this thesis, and thesis outline are summarized.

1.1 FEM for Hybrid Ionized Field Computation

High-voltage direct current (HVDC) transmission provides advantages over high-voltage alternating current (HVAC) transmission because of not only higher power transmission capacity but also less line loss [3]. These benefits and also the growing demand for electric power motivate the development of HVDC-related projects. Moreover, hybrid HVAC/HVDC configuration is selected as a favored choice to apply into projects due to the limited access to new right-of-way and fewer costs at network infrastructure [4]. However, series of associated problems come along as well. An experiment showed that the concentration of ions can still be detected eight hundred meters away from an HVDC transmission line and thus the environmental effect of the ions' generation and migration due to the high voltages stressed on transmission lines is important to investigate [5]. In addition, the exposure of new ions disturbing the natural balance between existed ions in the air, together with audio noise and radio noise around the transmission line may lead to biological effects [6]. To assess all these impacts and design the configuration for HVDC, a detailed analysis of the ionized field around hybrid HVAC/HVDC transmission lines becomes a critical part.

The process of ionized field builds up around transmission lines is briefly introduced here. The ionized field is closely tied with the corona phenomenon. The HVDC transmission lines are generally operated at the voltage of 500kV or above, leading to an intensive electric field near the conductor which exceeds the corona onset field (a critical value of the electric field to start a corona). With the strong electric field, the electrons surrounding the conductor can obtain so high kinetic energy that the air molecules can be ionized by collision. These collisions create an abundance of positive and negative ions and then the corona discharge begins to happen. As a result, an ionization layer around the conductor as a source of ions and an interelectrode space full of ions are formed around HVDC transmission lines. The ions in the space are forced to migrate by the electric field and the moving ions change the electric field in turn. After a while, a continuous steady state of the electric field in this space will eventually arrive either the transmission line is configured as a unipolar HVDC (only positive or negative voltage applied to transmission line) or a bipolar HVDC (one positive and one negative voltage applied to two conductors). And, this electric field affected by the corona is the ionized field. However, when the HVAC transmission lines share the right-of-way with HVDC transmission lines, the ionized field in the space will not reach a static state and become much more complex because of the proximity of the time-varying HVAC. It is important to conduct a transient study for this ionized field around hybrid transmission lines in advance of the project implementation. Moreover, the weather conditions, such as wind, rain, fog, and snow, should also be considered since the ionized field is influenced by them.

Studies about the ionized field have been conducted over the past century. Some classical line models such as the serial π models and traveling wave model hardly simulate the coronas around transmission lines. At the beginning of the last century, researchers intuitively started to find out the analytical solutions for ionized field [7]. However, analytical analysis is only valid for the voltage-current characteristic of the ionized field due to the inherent complexity of the ionized field problem. With the development of the computer, numerical analysis commences being a feasible approach. But the famous Deutsch's assumption was still a critical step to simplify the ionized field before applying the numerical method before 1970 [8, 9]. At the end of the 1970s, an iterative algorithm to solve Poisson's equation and the current continuity equations together with the FEM was proposed to solve ionized field by Janischewskyj [10]. Following this iterative scheme, different techniques were used to solve the abovementioned governing equations leading to various approaches to studying the ionized field.

Investigations on the ionized field around hybrid transmission lines commenced at the beginning of the 1980s [11]. Different from the HVDC-only configurations, the interaction between AC and DC conductors plays an important role on the resultant ionized field which is not time-invariant but time-varying. To study this ionized field, a FEM-based iterative method [12] and an AC-cycle-decomposition-based method [13] were utilized but they were either too computationally costly or losing the dynamics of this time-varying ionized field.

FEM has been employed to solve ionized fields since the numerical methods appeared. It has proved to be a detailed and accurate numerical method to calculate the electric potential distribution and the space charge distribution of the ionized field. With the granular mesh, the high accuracy for the distribution solutions can be provided to analyze the ionized field. However, a small granularity brings about not only a much more detailed solution but also a fairly heavier computation burden. Though the demand for detailed solutions for ionized field problems with a large domain, obtaining a detailed solution efficiently has been studied in few reports. An efficient algorithm using FEM and an upwind technique is proposed in this work to accelerate the computation speed for the ionized field around hybrid transmission lines. Besides, the speedup compared to commercial software is provided.

1.2 FEM for Power Transformer EMT

In a power system, power transmission is no less important than electricity generation. In terms of power distribution, the transformer plays an important role in stepping up and stepping down the voltages and thus it is critical equipment to deliver power successfully and efficiently. Transformers can transfer energy between circuits and mitigate grid congestion caused by more renewable installations with a proper control system. However, transformers may be under too much stress to be damaged when they are exposed to EMTs. Moreover, the costly bill of replacement and suspectable power system reliability will follow behind. Both the potential risk of severe damage and the expensive bill make it more important to implement a detailed transient study on the transformer before manufacture.

Transformer representation is not easy because of the complicated behavior of the transformer caused by the nonlinearity and frequency-dependent characteristics of materials. Several different transformer models have been proposed and applied with the development of technology. There are mainly five groups of transformer models in the literature [14]:

1. **Matrix representation** is based on steady-state equations for one multiphase multi-winding transformer. To analyze the transient-state model, the discrete steady-state equations will be developed. Either branch impedance matrix or admittance matrix can be used to form the equations. But branch impedance matrix is prone to glitches when injecting small exciting currents. Both phase-to-phase coupling and models terminal characteristics are considered in this model while different core or winding topologies cannot be distinguished [15]. Though this model is linear, connecting an external circuit is used to include nonlinearity such as saturation and hysteresis effect. This model needs less computation power because of the fixed admittance matrix for linear components. However, the accuracy is limited in this model.

2. **Saturable transformer component** is developed according to the star-circuit representation. This transformer model consists of resistors, inductors, and two-winding transformer, and an external nonlinear inductor that models the saturation effect. This model has three main restrictions: 1) this model is incapable of transformers with more than three windings due to the invalidity of the star circuit when the number of winding is larger than three; 2) it is not always topologically correct that the nonlinear inductor is connected to the star point [16]; 3)numerical instability for the three-winding case has been unveiled and some modifications are required on the traditional model to solve this issue [17, 18].

3. **Topology-based models** contain two subgroups: geometric models and duality-based models. Geometric models are constructed with previous mathematical description while duality-based models are not. The coupling between magnetic equations and electrical equations is utilized to derive the formulation with consideration of core topology. Though this model is not as detailed as finite element (FE) models, it is more precise than the simpler transformer models. The duality-based models are developed from an equivalent circuit model, which results from a magnetic circuit

model by the principle of duality [19]. The equivalent magnetic circuit is composed of magnetomotive-force sources standing for windings, linear reluctances standing for leakage paths, and saturable reluctances standing for magnetic cores. This model includes the saturation effects in the individual leg of the core, the leakage effects, and interphase magnetic coupling meanwhile it can decouple the core equivalent functions and winding configuration. Both models can guarantee considering the topology and include the winding resistance, core losses, and the capacitive coupling effects by attaching extra circuits. The topology-based models may lead to a larger computational burden with their detailed nature.

4. **Hybrid transformer model** combines matrix representation and topology-based models to achieve improvement of correctness. This model can be partitioned into four parts by function: leakage representation, topologically correct core representation, winding resistances, and capacitive effects [20]. This model has both the strength of matrix representation and also duality methods. Together with more details, the computational burden is a lot for this model.

5. **Finite element model** is derived based on the field-oriented Maxwell's equations which can describe the physical parameters and material properties in detail and thus provide the most accurate results [21]. By meshing the geometries of the transformer, no matter how irregular the geometries are, the solution of the transformer is described by values at hundreds to thousands of nodes. With so many nodes to solve for each transformer, the computation efficiency becomes a concern though more accuracy can be obtained. With the development of parallel computing, computation efficiency can be improved.

1.3 General Procedure of FEM

Compared to simplified lumped models, the FEM is capable of obtaining more accurate transient behaviors for electromagnetic equipment such as transformers. In addition, FEM is well-known for its superb accuracy and ability to analyze irregular geometries and complex material properties for both 2-D and 3-D problems. Solving Maxwell's equations directly by FEM, the electromagnetic field distribution inside and around the equipment is acquired. Then, the relation between voltage and current on devices can be investigated. Although FEM is capable of solving different problems, the general process for FEM is universal. FEM generally follows the five steps listed here [22]:

- Meshing: The problem domain is discretized by interconnected nodes and elements with selected shapes, triangular or quadrangular shapes are popular for 2-D and tetrahedral is popular for 3-D. The continuous electromagnetic field distribution is expressed by values at vertices.

- Deriving elemental equations: Corresponding elemental equations can be generated for each element based on the element shape selection. There are different approaches to obtaining the elemental equations, such as variational principle, Galerkin method, Ritz method.

- Assembling the global matrix system: A global matrix system can be obtained by assembling all the elemental equations within the problem domain with the mesh information. A large sparse matrix is involved for the global system and the size of the global matrix depends on the number of the nodes within the problem domain. For a large problem domain, the size of the matrix may be hundreds to thousands.

- Solving global matrix system: Impose the boundary conditions and solve this global matrix system by sparse matrix solver. Newton-Raphson iteration may be required for nonlinear problems.

- Post-processing: Obtain the results and restore the electromagnetic field distribution, plot, and display.

Amongst the five steps introduced above, this thesis will focus on the algorithm design to improve the efficiency of finite element computing (step two to step four). The meshing and the post-processing will not be discussed in this thesis. Any information relevant to the mesh will be assumed given and Matlab™ will be utilized to complete post-processing. Problems in this thesis will be solved by proposed algorithms with pre-known mesh information. A commonplace concern of solving the ionized field problem is the large problem domain and irregular geometry but FEM can handle the irregularity of geometry by discretizing the whole domain into small elements. With parallel computing for all the elements within the domain, a large domain can be solved very fast. In terms of transformers, the nonlinear material properties increase the difficulty of modeling transformers precisely. However, the FE model for a transformer can discretize the transformer into very small elements and solve the domain inside and outside of the transformer. Besides, the development of high-performance computing hardware reveals opportunities for researchers to improve the computation efficiency of FEM, and this will be discussed in chapter 2.

1.4 Motivation and Objectives of This Thesis

EMT study is extremely important when planning a power system or designing a power equipment because not only phenomena in EMTs are commonplace in power systems, but also these phenomena might lead to disastrous effects such as damaged components and economic loss. Many approaches for different power equipment have been developed over many decades. FEM, as a precise numerical method, together with Maxwell's equations has been investigated to model and analyze electromagnetic problems relevant to transmission lines, transformers, etc. Along with the high accuracy in FEM, the increased computational burden is an expensive cost for detailed analysis, resulting in a much longer design cycle for both power system planning and apparatus prototyping. The demand for detailed but also fast algorithm has greatly increased.

Here are the main objectives of this thesis:

- The retrofitting of existing multi-circuit AC transmission towers with DC lines to fulfill an HVDC project is an economic and efficient approach. However, this hybrid ionized field around the transmission lines becomes more complex because of the proximity of AC circuits and DC circuits. Due to the large problem domain and the time-marching nature of the hybrid ionized field causing a much longer computation time, an efficient numerical computation is desired for prototyping. Few materials have investigated the computational efficiency of the hybrid ionized field. Making progress on improving computation efficiency becomes feasible nowadays, with the development of high-performance computing hardware. Proposing an efficient algorithm to analyze the hybrid ionized field is one of the objectives of this thesis.

- Detailed analysis of the transformer's behavior under EMTs is always an essential step for transformer design. With the help of FEM along with Maxwell's equations, very accurate analysis for a transformer is achievable, but at a cost of the heavy computational burden. In order to deliver detailed results while maintaining precise real-time simulation, an appropriate platform is the important first step. FPGA, known for its inherent parallel architecture and special pipelining technique, has been chosen as the development platform. A suitable real-time algorithm on FPGA for transformer modeling is expected.

1.5 Thesis Outline

This thesis consists of five chapters and is organized as follows:

- **Chapter 1: Introduction** - This chapter introduces the EMTs in the power system, the background for ionized field analysis, and the background for transformer modeling. Besides the motivation of the thesis and the outline are summarized.

- **Chapter 2: FEM and Parallel Computing** - Three critical components to accelerate FEM computation are listed in this chapter: parallel computing hardware, parallel algorithms for FEM computation, and development tools for parallel computing. Both the architecture of GPU and the architecture of FPGA are described here. Prevalent parallel algorithms such as preconditioned conjugate gradient method which can be employed in massively parallel and nodal domain decomposition scheme which follows the divide and conquer algorithm are explained as well. At last, advanced development tools, CUDA APIs for GPU programming and HLS for FPGA design tools, are briefly introduced.

- **Chapter 3: Parallel Finite Element Computation of Time-Varying Ionized Field Around Hybrid AC/DC Lines** - This chapter proposes an algorithm with nodal domain decomposition approach and nodal charge conservation method to solve the hybrid ionized field caused by the proximity of HVDC and HVAC transmission lines. The ionized field is first modeled by finite elements and then the field quantities such as electric potential and charge densities are solved by FEM. The proposed algorithm is utilized to accelerate FEM computation in a massively parallel manner on GPU.

- **Chapter 4: Time-Stepped Finite-Element Modeling of Three-Phase Transformer for Electromagnetic Transient Emulation on FPGA** - In this chapter, a real-time adaptive TLM with PCG solver is explained to solve electromagnetic transients on transformers. This solver is implemented on FPGAs and the comparison between the solver results and the commercial software results demonstrates high accuracy. Besides, an indirect field-circuit coupling technique is discussed to interface the FE model and drive circuits.

- **Chapter 5: Conclusions and Future Works** - The research conclusions are provided and suggestions for future research are discussed in this chapter.

2 FEM and Parallel Computing

The computer hardware manufacturing industry faced a great challenge in the past decade. The golden way to increase the computing power started to show deficiencies – an increment of single-core clock frequency started to demonstrate its limitations in improving computing performance. However, this is not the end of high-performance computing hardware. A new trend, the integration of many computing cores to collaborate, started to make progress in this field. Parallel computing is the most prevalent collaboration strategy. As the name indicated, parallel computing enables the cores to work simultaneously for the same program, which means the computation power is expected to enhance significantly with more cores involved (see Fig. 2.1). For instance, the most powerful supercomputer Fugaku, sponsored by the Japanese government, has over eight million processing cores providing a throughput of over 480P floating-point operations per second [23]. On the other hand, NVIDIA® company supplied high-quality GPUs with over five thousand CUDA cores to step up high-performance computing stations in the laboratory [24], as shown in Fig. 2.2.

Besides, field-programmable gate arrays (FPGAs) are integrating millions of gates and high bandwidth memory to meet the increasing demand for high-performance scientific computation due to Moore's law which states the size of the transistor will shrink exponentially. Take Xilinx® Virtex® UltraScale+ HBM VCU128-ES1 board with the XCVU37P FPGA as an example, it has 9024 DSP48E slices, 2607360 flip-flops, and 1303680 look-up tables to deploy a design in a massively parallel processing manner. FPGA can also deal with data in a pipelining manner to accelerate computing speed.

The parallel computing resource only is not enough to improve the finite element computation efficiency. Based on Amdahl's law, the speedup of a program is limited by the fraction of the program that can be parallelized, meaning that a highly parallelized al-

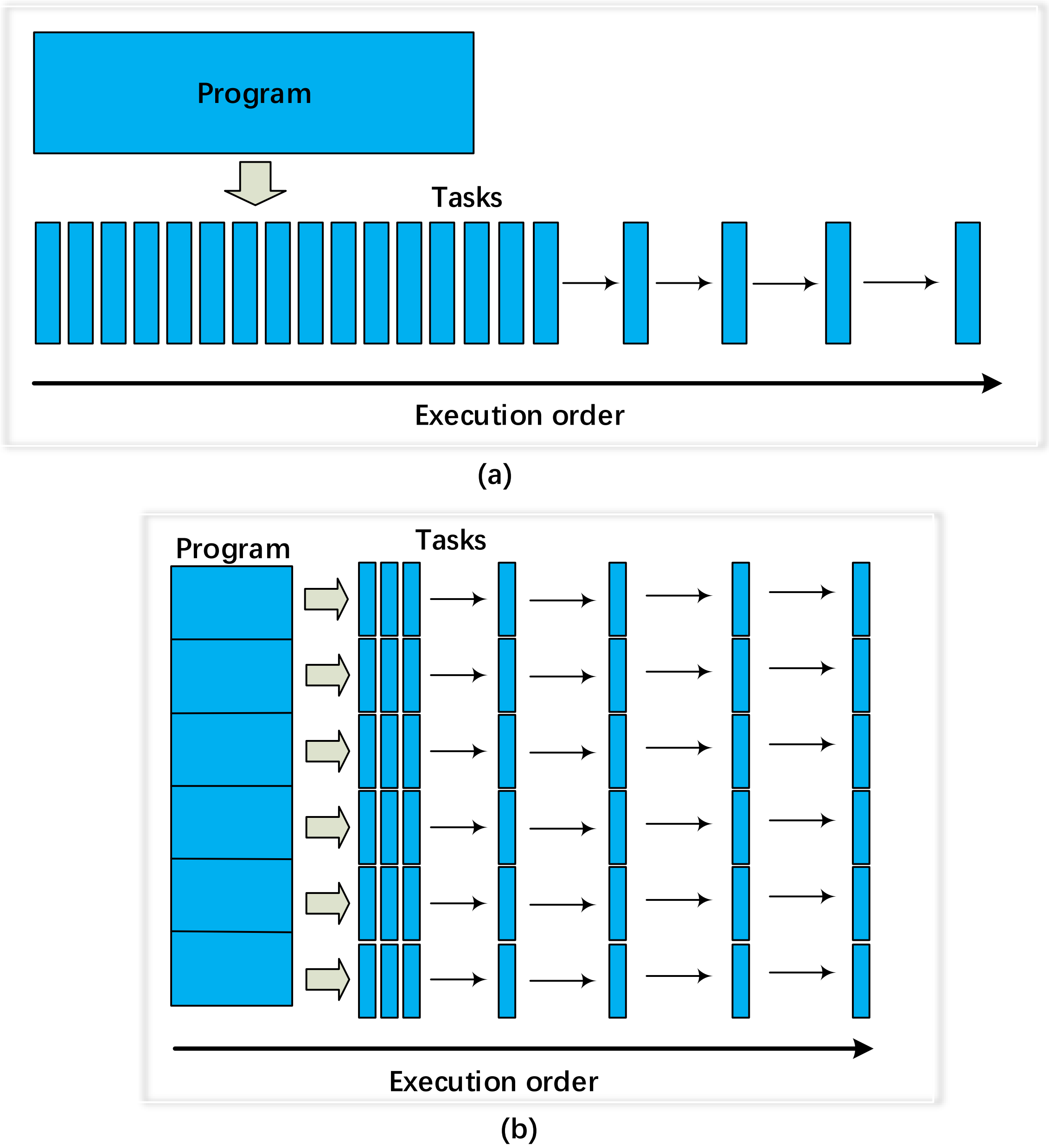
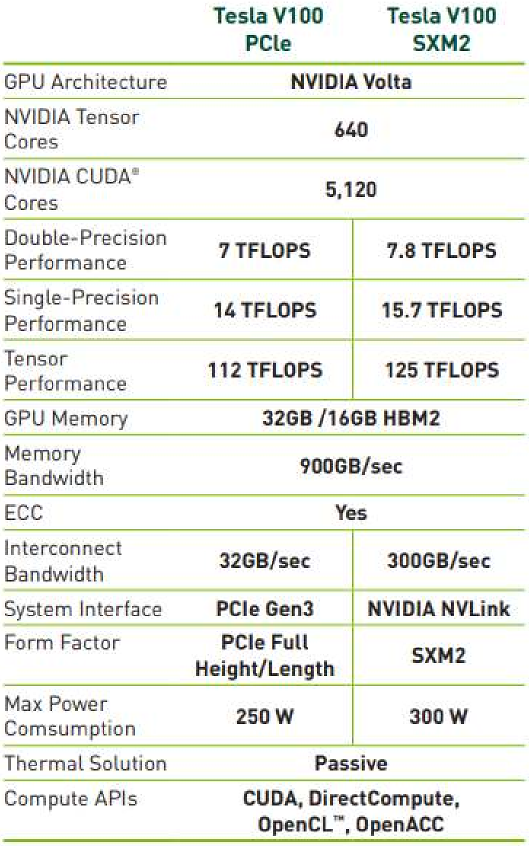


Figure 2.1: The same program is executed sequentially and in parallel. (a) The program is divided into sequential tasks. (b) The program is broken into discrete tasks that can be executed concurrently.

gorithm for finite element computation is required to accelerate the computing speed. Together with the parallel computing resources, a suitable parallel algorithm to use resources wisely will increase computation efficiency.

To design a new parallel algorithm in the parallel computing environment, developer-friendly development tools will be necessary. Different from sequential programs, a parallel algorithm needs to provide commands for many computing cores to make them work



	Tesla V100 PCle	Tesla V100 SXM2
GPU Architecture	NVIDIA Volta	
NVIDIA Tensor Cores	640	
NVIDIA CUDA® Cores	5,120	
Double-Precision Performance	7 TFLOPS	7.8 TFLOPS
Single-Precision Performance	14 TFLOPS	15.7 TFLOPS
Tensor Performance	112 TFLOPS	125 TFLOPS
GPU Memory	32GB /16GB HBM2	
Memory Bandwidth	900GB/sec	
ECC	Yes	
Interconnect Bandwidth	32GB/sec	300GB/sec
System Interface	PCIe Gen3	NVIDIA NVLink
Form Factor	PCIe Full Height/Length	SXM2
Max Power Comsumption	250 W	300 W
Thermal Solution	Passive	
Compute APIs	CUDA, DirectCompute, OpenCL™, OpenACC	

Figure 2.2: Specifications of NVIDIA® Tesla V100 series GPU.

simultaneously. Also, making configuration to store shared data in shared memory is required sometimes. Besides, the communication between the host and the peripherals which perform the parallel computing is also essential. To achieve these goals, dedicated APIs will be required. For FPGAs, a tool is also required to mitigate the common concern about hardware development difficulties which might imply a longer development cycle.

2.1 High-Performance Computing Hardware

2.1.1 Advanced GPU Architecture Used in This Thesis

The parallel computing hardware is capable of supporting concurrent execution of multiple processors or cores to achieve computation speedup. To utilize the parallel computing hardware for computation acceleration, it is important to understand the hardware architecture for the program deployment.

In this thesis, NVIDIA® Tesla® V100 will be used along with the compatible programming API to execute the proposed algorithm in order to accelerate the finite element computation of power system models. The Tesla V100 GPU incorporating Volta™ GV100 GPU architecture, is a cutting-edge parallel computing processor. The Tesla V100 is composed

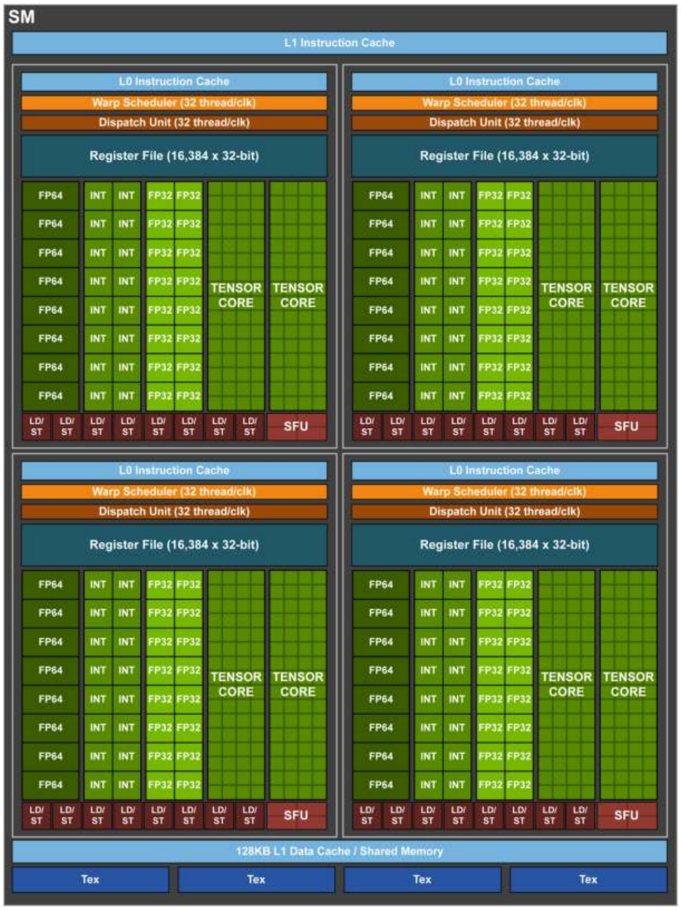

SM
L1 Instruction Cache
L0 Instruction Cache
Warp Scheduler (32 thread/clk)
Dispatch Unit (32 thread/clk)
Register File (16,384 x 32-bit)
FP64
INT
FP32
TENSOR CORE
LD/ST
SFU
128KB L1 Data Cache / Shared Memory
Tex

Figure 2.3: Hardware resources within one streaming multiprocessor in NVIDIA® Tesla V100 GPU.

of 80 streaming multiprocessors (SMs) and each SM has 64 FP32 cores, 64 INT32 cores, 32 FP64 cores, 8 Tenson cores, and 4 texture units, resulting in the capability of running 5120 single-precision operations, 5120 integer operations, and 2560 double-precision operations concurrently. An inner view of one streaming multiprocessor is shown in Fig. 2.3. Besides, a shared memory whose capacity can be configured up to 96KB per SM enables instant communication between SMs [25]. Furthermore, the NVIDIA NVLink™ high-speed interconnect and 16GB HBM2 high-bandwidth memory are included in Tesla V100 to obtain more efficient inter-SM and inter-GPU communications. The flexibility to switch between maximum performance mode and maximum efficiency mode is also provided by Tesla V100.

Since there is no instruction dispatcher for each arithmetic logic unit (CUDA core) in Fig. 2.3, the GPU is designed with single instruction multiple data (SIMD) paradigm. SIMD is one type of parallel architecture and all cores in the computer execute the same instructions at any moment but calculating on different data streams. However, for the execution of each core, the data stream can be operated in sequential manner. If we consider the elements at least one of whose edges is at the boundary as boundary elements. Generally, all the elements except boundary elements in one FEM model can be calculated in the same procedure. Due to the consistency of operations within any non-boundary element for one FEM model, the FEM computation can be paralleled to fit this SIMD architecture to benefit from the power of the advanced GPUs.

2.1.2 FPGA Primer

The greatest innovation on FPGA is that its functionality and wiring are controlled by configurable logic blocks (CLBs), programmable interconnect pattern via switch matrix, and the user-controlled input/output bank (IOBs). Besides, these components are all reprogrammable, which results in instant availability and improved visibility of a failure [26]. A CLB generally includes multiple logic cells, each of which typically contains one four-input look-up table (LUT), one D-type flip flop (DFF), and a multiplexer. The number of logic cells varies with utilized fabrication process technology. The CLB as repeating logic resources can provide operational elements that can implement complex logic functions and execute memory functions for building desired logic. The LUT, which is synthesis friendly despite the size of problem design after the dedicated LUT mapper adopted the simplicity of mapping any functions into LUTs, can implement any of the combinations of its four inputs [27]. The DFF works as a fundamental memory component to register "data". The multiplexer, also known as the data selector, often shares one output port between several input signals by choosing from these inputs and passing the chosen input to a single output line. The IOB can be designed to be input, output, or bidirectional ports in order to define the interface between the peripherals and the internal logic. The interconnect established the routing path between CLBs and IOBs.

The most appealing attribute of FPGA is its intrinsic parallel hardware architecture, which enables FPGA to be configured as multiple parallel execution units that compute data simultaneously. On contrary, the sequential processing applied to CPUs executes instructions sequentially, leading to low computation efficiency. The multi-thread technique has been explored to offer parallelism on CPUs. However, interference may occur between multiple threads when cache, buffers, and other hardware resources are shared, resulting in impaired execution time. Nowadays, multi-core CPUs are exploited in parallel computing to provide parallelism and therefore improve computation performance at the expense of sophisticated strategy and higher power consumption. The corporation between several cores including but not limited to data sharing, tasks scheduling, and synchronization are generally costly. Plenty of memory blocks are available on FPGAs and they can operate as single-port random-access memory (RAM), dual-port RAM, read-only memory (ROM), or First-In-First-Out (FIFO) depending on the configuration. Besides, lots of independent memory blocks on FPGA can be accessed in a parallel manner.

The pipelining is a valuable design mechanism to boost FPGA's performance. In terms

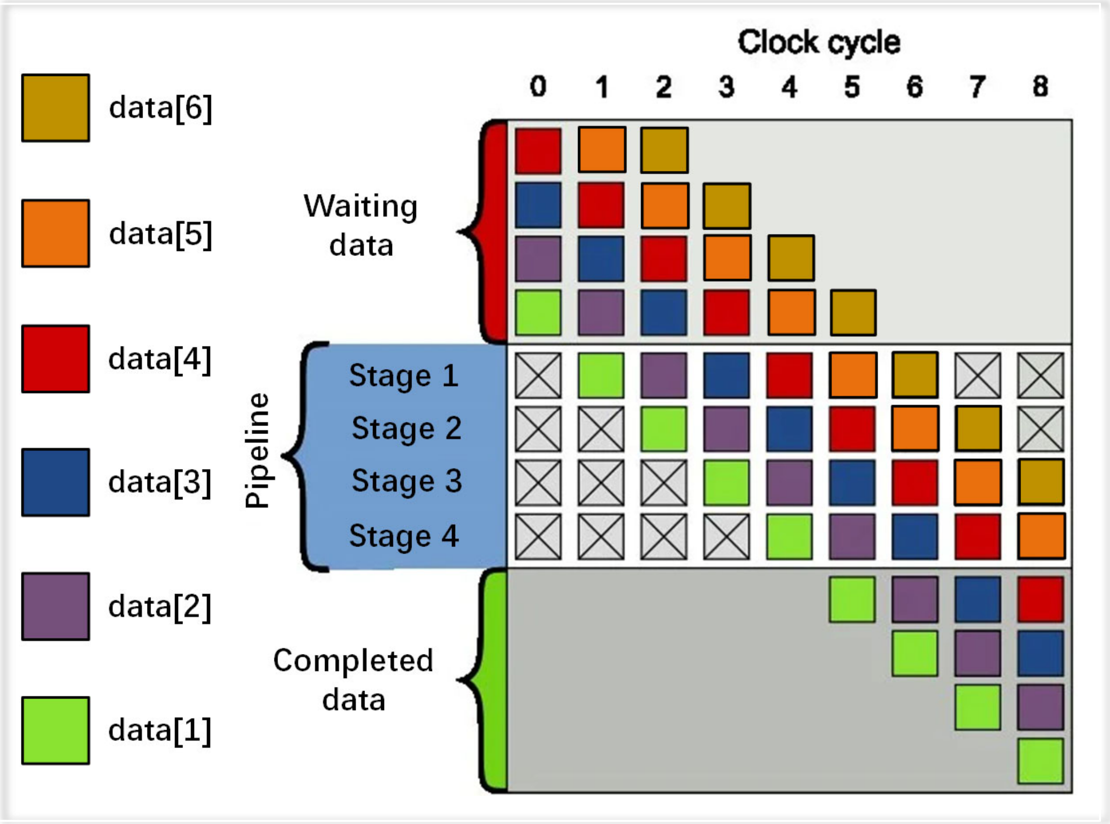


Figure 2.4: An example to illustrate the pipelining on FPGA [28].

of hardware configuration, two main steps are implemented to utilize the pipelining technique. The first step is dividing the process of achieving a function into several stages, such as N stages while the second step is that several registers are inserted between every two stages. As data march through the process, the immediate output of a previous stage will be stored in the register and will be read by the next stage as an input. After these two steps, one function is pipelined and at most N groups of inputs are being processed concurrently at the same cost of hardware utilization as running only one process of the function. Besides, although the latency of this pipeline may be high, the throughput of this pipeline can reach one result per clock cycle except for the first $(N-1)$ clocks. The strong point of a pipeline is the promising throughput. Assume we have an array of six data and each data needs to go through four stages to get the result $(N = 4)$. In Fig. 2.4, the pipelining is illustrated.

With the more-than-30-year development driven by process technology and also application demand, the number of transistors or gates is not the message FPGA vendors deliver to imply its powerful performance anymore. After entering the new millennium, it was not hard to design an FPGA capable of solving a problem with typical size due to the inescapable pace of Moore's law. It was the right time to think about customer experience and the ease-of-use of FPGA. Predefined targeted logic blocks such as high-speed transceivers, communication, and networking blocks were integrated into FPGA, libraries of soft logic functions were constructed and provided, and FPGA toolsets for example High-Level Synthesis (HLS) compiler simplifying the process of program optimization from high-level programming languages were rendered [29]. As a result, FPGA has become more user-friendly than it was.

2.2 Parallel Algorithms for FEM Computation

Parallel computing can be considered as using multiple computing resources simultaneously in order to achieve concurrent calculations. A large problem is broken up into small pieces and each piece is calculated simultaneously with different computing resources. With the state-of-art parallel computing hardware, a remaining challenge is how to design an algorithm to utilize these resources efficiently to accelerate FEM computation.

2.2.1 Preconditioned Conjugate Gradient Method

As described in Section 1.3, a large matrix equation to model an FE problem in the whole analysis domain can be obtained after the assembly step and this matrix equation can be notated as $KX = b$. K is the stiffness matrix, whose size is determined by the number of finite element nodes and whose entries are determined by the material properties and also the mesh. K is always a sparse matrix, in which a multitude of entries is zero since one finite element node is directly connected to its neighboring nodes only (as shown in Fig. 2.5). One FE problem is solved when the large matrix equation is resolved and thus the efficiency of the finite element computation relies heavily on the applied sparse matrix solver.

There are two groups of approaches to solving a sparse matrix system: direct methods which obtain solutions with a single application of one numerical technique, and iterative methods which use the numerical technique repetitively until convergence or user-defined error threshold [30, 31]. The popular direct methods, such as Gaussian elimination, lower-upper (LU) decomposition, and Cholesky decomposition solve a matrix equation in a sequential manner which is not suitable to the parallel computing architecture very well. However, some iterative methods, Gauss-Seidel method, Jacobi method, and Conjugate Gradient (CG) were explored to implement on parallel platforms recently [32–35]. Due to a large amount of matrix-vector multiplication in CG algorithm and the trend that a high volume of dedicated blocks for matrix-vector multiplication operations are included in FPGAs, CG algorithm can be used to solve FE problems efficiently. To move further, a preconditioned conjugate gradient (PCG) algorithm can be applied to converge faster with a featured matrix.

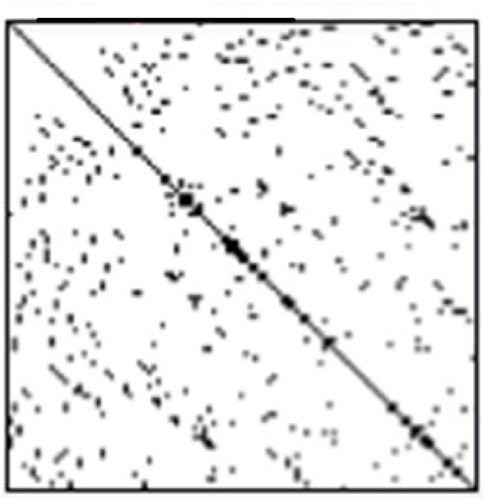


Figure 2.5: Distribution of nonzero entries in a sparse matrix.

2.2.2 Nodal Domain Decomposition Scheme

Solving the large matrix equation after assembly, which follows the traditional steps to solve FEM problems (in Section 1.3), demonstrates the centralized thinking in the traditional solving process, resulting in the difficulties of solving a FEM problem in a massively parallel manner. Following the decentralized divide-and-conquer thinking pattern, one type of strategies to fractionate a finite element problem is the domain decomposition approach. A large problem domain is broken down into multiple small and manageable subdomains, a large matrix is divided into multiple smaller matrices, and then each subdomain or smaller matrix can be conquered individually [36, 37]. According to the abovementioned Amdahl's law, a method to fractionate the problem domain into subdomains as small as possible will maximize the speedup for FEM problem computation. In the nodal domain decomposition method, there is only one unknown in each subdomain, which is sufficiently small to provide a satisfying speedup.

Take the linear problem domain in Fig. 2.6 (a) as an example to introduce the nodal domain decomposition method. For this 2-D finite element problem, the mesh is generated and consists of multiple small triangular elements. Each element is composed of three nodes, each of which is associated with a value for the field variable to be resolved. A subdomain in the mesh can be formed by all the triangles around one node and one example subdomain system is shown in Fig. 2.6 (b). The subdomains in the mesh are overlapped. According to the subdomain in Fig. 2.6 (b), one linear equation with only one unknown (the value at the blue node) and the values at neighboring nodes (the values at the red nodes) can be obtained. The values at neighboring nodes can be considered as known boundary conditions. Iterations are required for subdomains to exchange information in

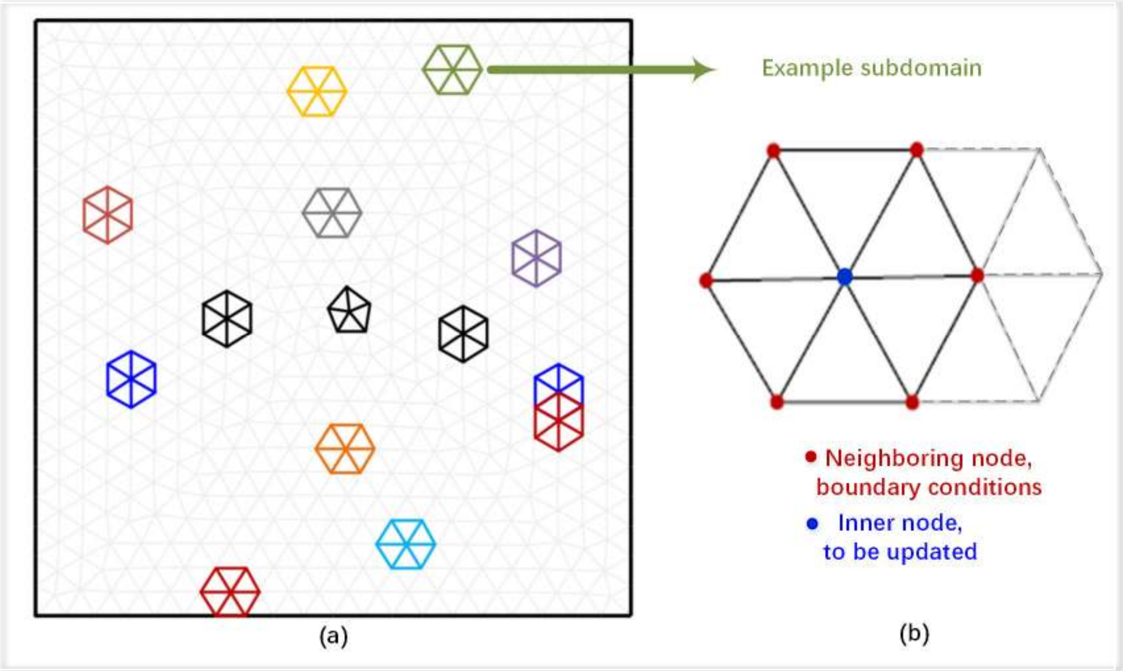

Example subdomain
● Neighboring node, boundary conditions
● Inner node, to be updated
(a)
(b)

Figure 2.6: An example for NDD scheme illustration.

order to reach a convergent solution at all nodes. Initial values are assigned to all the nodes at the beginning of each iteration. Then, the subdomains in the entire problem domain are calculated simultaneously in a massively parallel manner since they are independent. The same pattern of the linear equations for all the subdomains (in Fig. 2.6 (b)) implying the consistency of operations for all the subdomains, makes the nodal domain decomposition method perfectly suitable for SIMD GPU architectures.

2.3 Development Tools Used in This Thesis

2.3.1 CUDA

Both parallel computing hardware and parallel algorithm have been introduced so far, however, in the programmer's view, one remaining question is how to map the concurrent calculations onto parallel computing devices. The compute unified device architecture (CUDA) is an application programming interface (API) model and parallel computing platform developed by NVIDIA® to enable engineers and developers to solve complex computational problems using GPUs. The CUDA platform provides an access to GPU's virtual instruction set and parallel computing cores via a software layer. To access to CUDA platform, developers can use CUDA-accelerated libraries, compiler directives, and extensions to industry-standard programming languages including C, C++, and Python. After writing a CUDA kernel using industry-standard programming languages extended with keywords such as CUDA C and then invoking the CUDA kernel with kernel's execution configuration inside triple-angle-brackets, the same function defined by the kernel is executed on multiple different parallel computing cores to apply the same operations to different data streams. Since the instructions in one kernel are applied to multiple datasets by making one CUDA kernel call, the SIMD program can be developed on GPUs with the help of CUDA. To achieve high performance, the CUDA also provides a fast-shared memory region to share data between threads. CUDA provides support for researchers to develop SIMD programs on GPUs to take full advantage of the GPU architecture.

2.3.2 HLS

With the growing recognition of hardware acceleration, the FPGA prototyping board includes more resources to support the demand for high-performance computing. Although the capacity expansion on FPGA and its increasing complexity and capability results in an increment in design efforts, FPGA vendors have taken efforts to ease this stress from the designers. They developed multiple modern design tools not only for automatic placement, routing, synthesis but also to help improve productivity for hardware designers without the loss of high performance on hardware design [38]. The HLS has proved to be such a tool to save development time while providing a similar performance on latency and resource utilization to hand-written hardware description language (HDL) codes [39]. HLS enables developers to design at a high level of abstraction since it can transform the sequential C and C++ codes into formats compatible with FPGA. Developers can use optimization directives to alter the default behavior of the internal logic and thus optimize the hardware functions, structures for performance and also to modify data access patterns. The HLS tool yields reports with plenty of metrics to analyze the performance such as resource utilization, latency, initiation intervals, and IP cores in HDL format which can synthesis into FPGA. HLS is a very useful tool to provide a large-scale complex hardware design with a short development cycle.

3

Parallel Finite Element Computation of Time-Varying Ionized Field Around Hybrid AC/DC Lines

3.1 Introduction

This chapter proposes a massively parallel algorithm to solve time-varying ionized fields around hybrid AC/DC lines. The performance of this algorithm has been investigated in terms of computation speed and accuracy.

The advantages of high-voltage direct current (HVDC) over high-voltage alternating current (HVAC) are presented in many perspectives: higher power transmission capacity, lower net cost for long-distance transmission, no skin effect, lower line loss, and easier system design [3]. Because of these benefits, numerous HVDC-related projects have arisen in recent years [40]. Among these projects, some are retrofitting of existing multi-circuit AC transmission towers with DC lines to increase power transfer because of the high cost of building new HVDC transmission towers, restrictive access to new right-of-way, and long duration transmission planning.

The proximity of AC circuits and DC circuits on the same transmission towers has an appreciable influence on the ionized field around them due to corona interactions between two circuits [41]. An efficient numerical computation for the ionized field is therefore desired by engineers for designing the conductor configuration and tower geometries, and assessing environmental impact. The computational efficiency is important for a hybrid ionized field because this is a time-marching problem with a large problem domain that requires repeated FEM computation to handle multiple degrees of freedom (DoF).

Research in the ionized field has been carried on for more than a century. However, computation efficiency wasn't a major focus. In 1914, the first analytical solution for coaxial cylindrical geometry was developed in [7], but it had very limited applications. In 1933, Deutsch laid the cornerstone for numerical analysis by proposing an assumption that the space charges affect the magnitude but not the direction of the corresponding charge-free field. Although Deutsch's assumption simplified the calculation, the validity of this assumption was questioned [42]. Researchers began to explore methods independent of Deutsch's assumption. The first FEM-based iterative method was proposed in 1979 in which the space charge was tuned until convergence using Poisson's equation and current continuity equation [10]. Different methods for solving Poisson's equation and current continuity equation and updating space charge were developed later [43–45], but the basic iterative algorithm remained, and computation efficiency has not been improved. Moreover, these methods were only applicable to HVDC scenarios because the steady-state result did not fit the hybrid situation.

The ionized field around hybrid transmission lines has been researched for almost four decades. In 1981, Chartier et al. observed that the voltage gradients on the surfaces of AC and DC conductors were time-varying for hybrid transmission lines, which was in opposition to HVDC where the voltage gradient was time-invariant [11]. Then, the significant impact of the AC-DC circuit interaction on the electric field was discovered [41]. In 1992, the FEM was used to iteratively calculate the space charge distribution [12] but the computation was still costly. Deutsch's assumption was used to reduce the cost, however, it also reduced the accuracy of the result. A calculation method that decomposed one AC cycle into several DC cases and solved these DC cases using the numerical method for the steady-state ionized field was developed in [13], but the dynamics of the hybrid ionized field were neglected.

However, few materials have been reported to handle the computational efficiency problem encountered in the ionized field computation, especially for the time-varying case. The goal of improving computation efficiency has become feasible with the development of high-performance computing hardware, and a critical step is finding an algorithm that can fully utilize this hardware. In this chapter, an algorithm based on FEM for solving the hybrid ionized field is proposed. To obtain the solution for Poisson's equation, a fine-grained nodal domain decomposition (NDD) methodology is implemented on graphic processing units (GPUs) to obtain the massive parallelism and hence to improve the computation efficiency. NDD methodology is chosen to provide sufficiently simple sub-domains with only one unknown in each sub-domain so that each sub-domain can be solved independently. The upwind nodal charge conservation (NCC) method is also a part of the algorithm and it is applied to solve current continuity equations. Each finite element node can be projected to a compute unified device architecture (CUDA) core since both NDD and upwind NCC are methodologies based on nodes.

This chapter is organized as follows. The assumptions to simplify the problem are given at the beginning of Section 3.2. In the same section, the governing equations and boundary conditions are introduced and explained. Then, in Section 3.3 the NDD methodology, upwind NCC method, and application of boundary conditions are described. Next, the hybrid line configuration, the result comparison between the proposed algorithm and COMSOL Multiphysics$^{\text{TM}}$, and the computation speed-up for a hybrid ionized field case are provided in Section 3.4. At last, a summary is given in Section 3.5.

3.2 Problem Description

3.2.1 Assumptions for simplification

To reduce the complexity of corona phenomena around the transmission lines, some reasonable assumptions are employed to build a solvable model [41, 46]:

1. The thickness of the ionization layer around the conductor is small enough to be neglected.
2. Diffusion of positive and negative charge is neglected since it has a very slight effect on the ionized field comparing to the convection.
3. The mobilities of positive ion and negative ion are assumed to be constant. The coefficient of recombination is assumed to be constant.
4. Ions generated by AC conductors due to corona are all restricted in a thin layer around the AC conductor, which is not a part of the analysis domain.
5. The Kaptzov's condition will be guaranteed, which states that the gradient of electric potential on the conductor will not exceed the onset initial gradient when corona happens.

3.2.2 Governing Equations

Poisson's equation and current continuity equations are governing equations for bipolar ionized field [46].

$$\nabla^2\varphi(t) = -\frac{\rho^+(t) - \rho^-(t)}{\epsilon_0}, \tag{3.1}$$

$$\frac{\partial\rho^+(t)}{\partial t} + \nabla \cdot \boldsymbol{J}^+(t) = -\frac{R\rho^+(t)\rho^-(t)}{e}, \tag{3.2}$$

$$\frac{\partial\rho^-(t)}{\partial t} - \nabla \cdot \boldsymbol{J}^-(t) = -\frac{R\rho^+(t)\rho^-(t)}{e}, \tag{3.3}$$

where $\varphi(t)$ is the total electric potential distribution (V). $\rho^+(t)$ and $\rho^-(t)$ represent the absolute value of positive and negative space charge density distribution respectively

(C/m^3). ϵ_0 is the vacuum permittivity, whose value is $8.854 \times 10^{-12} F/m$. $\boldsymbol{J}^+(t)$ and $\boldsymbol{J}^-(t)$ are current density vectors for positive ions and negative ions respectively (A/m^2). R represents the coefficient of recombination and it is approximated to $2.2 \times 10^{-12} m^3/s$ [46]. e is the charge of one electron and its value is $1.602 \times 10^{-19} C$.

The current density vectors used above are defined by (3.4) and (3.5).

$$\begin{aligned} \boldsymbol{J}^+(t) &= \rho^+(t)\boldsymbol{v}^+(t) \\ &= \rho^+(t)\big(k^+\boldsymbol{E}(t) + \boldsymbol{W}(t)\big) \\ &= \rho^+(t)\big(-k^+\nabla\varphi(t) + \boldsymbol{W}(t)\big), \end{aligned} \tag{3.4}$$

$$\begin{aligned} \boldsymbol{J}^-(t) &= \rho^-(t)\big(-\boldsymbol{v}^-(t)\big) \\ &= \rho^-(t)\Big(-\Big(k^-\big(-\boldsymbol{E}(t)\big) + \boldsymbol{W}(t)\Big)\Big) \\ &= \rho^-(t)\big(-k^-\nabla\varphi(t) - \boldsymbol{W}(t)\big), \end{aligned} \tag{3.5}$$

where $\boldsymbol{v}^+(t)$ and $\boldsymbol{v}^-(t)$ represent the velocity of the positive ions and velocity of the negative ions, respectively (m/s). $\boldsymbol{E}(t)$ represents the electric field distribution (V/m). Note that without other external forces or effects, the negative charges move in the opposite direction of the electric field. $\boldsymbol{W}(t)$ represents the wind velocity distribution of the discussed domain (m/s) and it is a vector. k^+ and k^- are the positive ion mobility $1.4 \times 10^{-4} m^2/(V \cdot s)$ and negative ion mobility $1.8 \times 10{-}4 m^2/(V \cdot s)$ [46].

The origin and the derivation of the two current continuity equations (3.2) and (3.3) are explained in [9]. One helpful note is that the left-hand side of the current continuity equation for negative charges describes the change rate of negative charges while the divergence of $\boldsymbol{J}^-(t)$ represents the change of positive charge.

At last, the total current density vector $\boldsymbol{J}(t)$ will be the sum of two current density vectors as shown below.

$$\boldsymbol{J}(t) = \boldsymbol{J}^+(t) + \boldsymbol{J}^-(t). \tag{3.6}$$

3.2.3 Boundary Conditions

Boundary conditions are necessary prerequisites to solve the three coupled governing equations with parameters $\varphi(t)$, $\rho^+(t)$ and $\rho^-(t)$. We will discuss the definitions of two commonly-used boundary conditions in this section while how to apply them into the proposed algorithm will be introduced in Section 3.3.3.

1. Dirichlet boundary condition is the most explicit boundary condition, which provides the values at the boundary. In this chapter, the Dirichlet boundary condition may occur on the ground boundary, conductor surfaces of AC and DC lines, and also other domain boundaries. Assume that the electrical potential at these boundaries is

V_0 and subscript Dirichlet boundary condition with D, the Dirichlet boundary condition can be expressed using the following equation:

$$\varphi(t)|_D = V_0. \tag{3.7}$$

2. Neumann boundary condition describes the normal gradient of the unknown function at the boundary. For this ionized field problem, the opposite of the gradient of $\varphi(t)$ is $\boldsymbol{E}(t)$ and so the Neumann boundary condition will provide the value of the normal component of $\boldsymbol{E}(t)$. Assume the normal component of $\boldsymbol{E}(t)$ is E_0 (E_0 is a known constant) and subscript Neumann boundary conditions with Γ, we will have the equation below.

$$\left.\frac{\partial\varphi(t)}{\partial n}\right|_\Gamma = E_0. \tag{3.8}$$

3.3 Massively Parallel Simulation via Nodal Domain Decomposition

The NDD and upwind NCC methodologies make it possible to fully utilize the computation capability of GPUs. In this section, the details of deploying NDD and applying upwind NCC on GPUs are introduced, followed by the explanation of applying the boundary conditions for solving the Poisson's and current continuity equations.

The main idea of this algorithm is to decompose the problem into as tiny sub-domains as possible, so as to maximize the computing efficiency using GPUs' parallel computation ability. To begin, the mesh for the problem is generated by COMSOL Multiphysics$^{\text{TM}}$. Then, the charge density distribution in the domain, including the charge density on the conductor surfaces and the charge density in the air, is initialized. Because of the coupling of governing equations, (3.1), (3.2), and (3.3) are solved alternatingly and iteratively. For each time step, the boundary conditions are applied before using NDD to solve (3.1) and therefore obtain $\varphi(t)$. When solving Poisson's equation, a massive amount of cores in GPUs are executing simultaneously to solve independent sub-domains. By using GPUs, the computation time for an 8184-node FEM problem can be decreased to 30 ms. The upwind difference method is applied to calculate the gradient of positive charge density and gradient of negative charge density. With the known $\varphi(t)$ and gradients of positive and negative charge density, (3.2) and (3.3) can be solved simultaneously and the positive and negative charge density distribution for this time step can be calculated. To satisfy Kaptzov's condition, the charge density on conductor surfaces is updated according to E_{onset} and the maximum magnitude of $\boldsymbol{E}$ in the domain. The flow chart of the proposed algorithm is shown in Fig. 3.1.

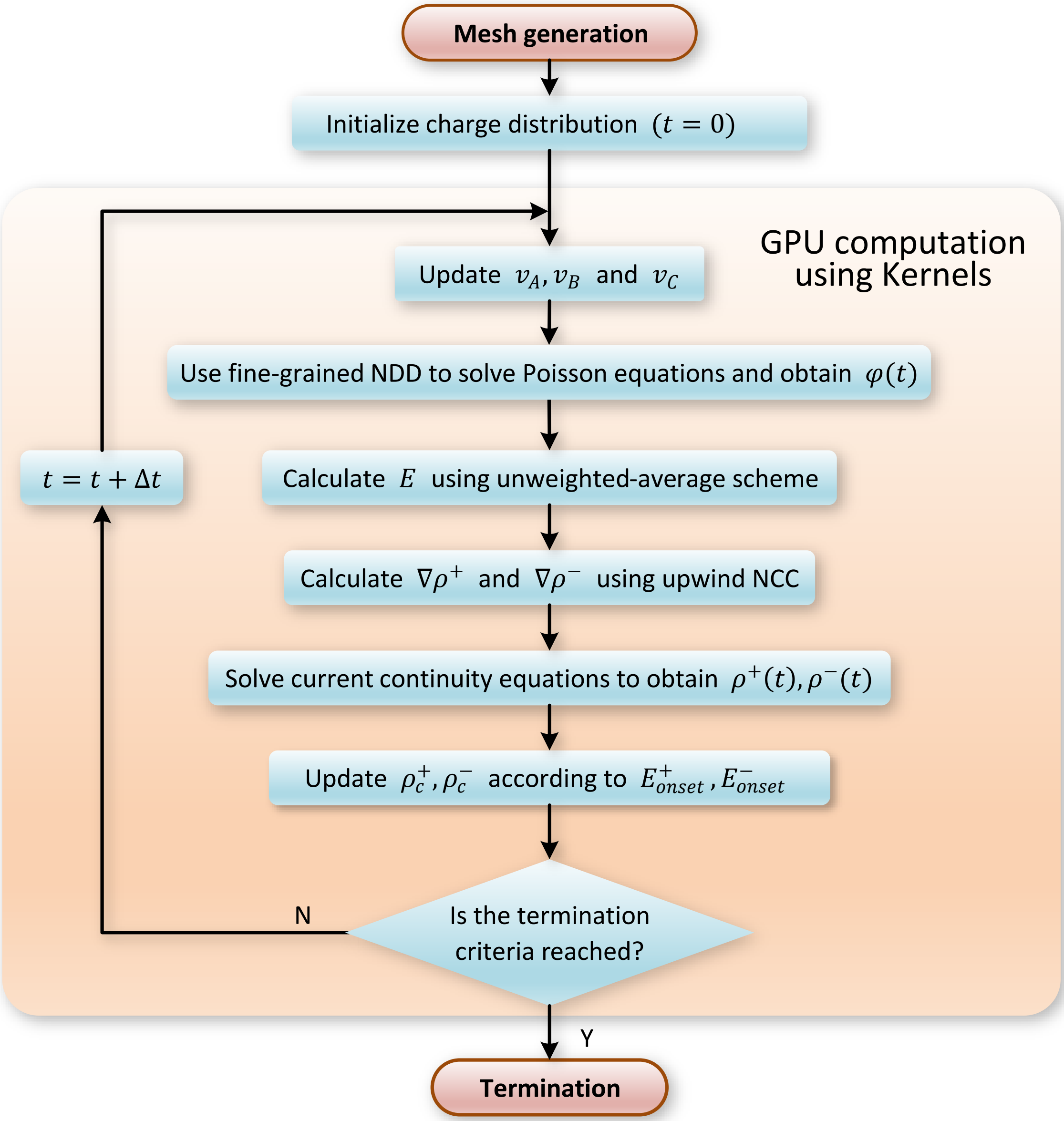

Mesh generation
Initialize charge distribution $(t = 0)$
GPU computation using Kernels
Update v_A, v_B and v_C
Use fine-grained NDD to solve Poisson equations and obtain $\varphi(t)$
$t = t + \Delta t$
Calculate E using unweighted-average scheme
Calculate $\nabla\rho^+$ and $\nabla\rho^-$ using upwind NCC
Solve current continuity equations to obtain $\rho^+(t), \rho^-(t)$
Update ρ_c^+, ρ_c^- according to E_{onset}^+, E_{onset}^-
N
Is the termination criteria reached?
Y
Termination

Figure 3.1: Flow chart of NDD and NCC schemes for parallel computation of ionized field.

3.3.1 Nodal Domain Decomposition

Domain decomposition is an efficient way to solve large-scale problems compared to solving these problems using multiple computational cores in a global system [47, 48]. Following this idea, the nodal domain decomposition (NDD) was proposed in [49]. In NDD, there is only one unknown in each sub-domain and each sub-domain can be solved independently and therefore NDD can be easily deployed in GPUs.

The Poisson's equation (3.1) in Section 3.2.2 can be solved by Galerkin FEM, which follows the steps: domain decomposition using FE, elemental formulation by appropriate

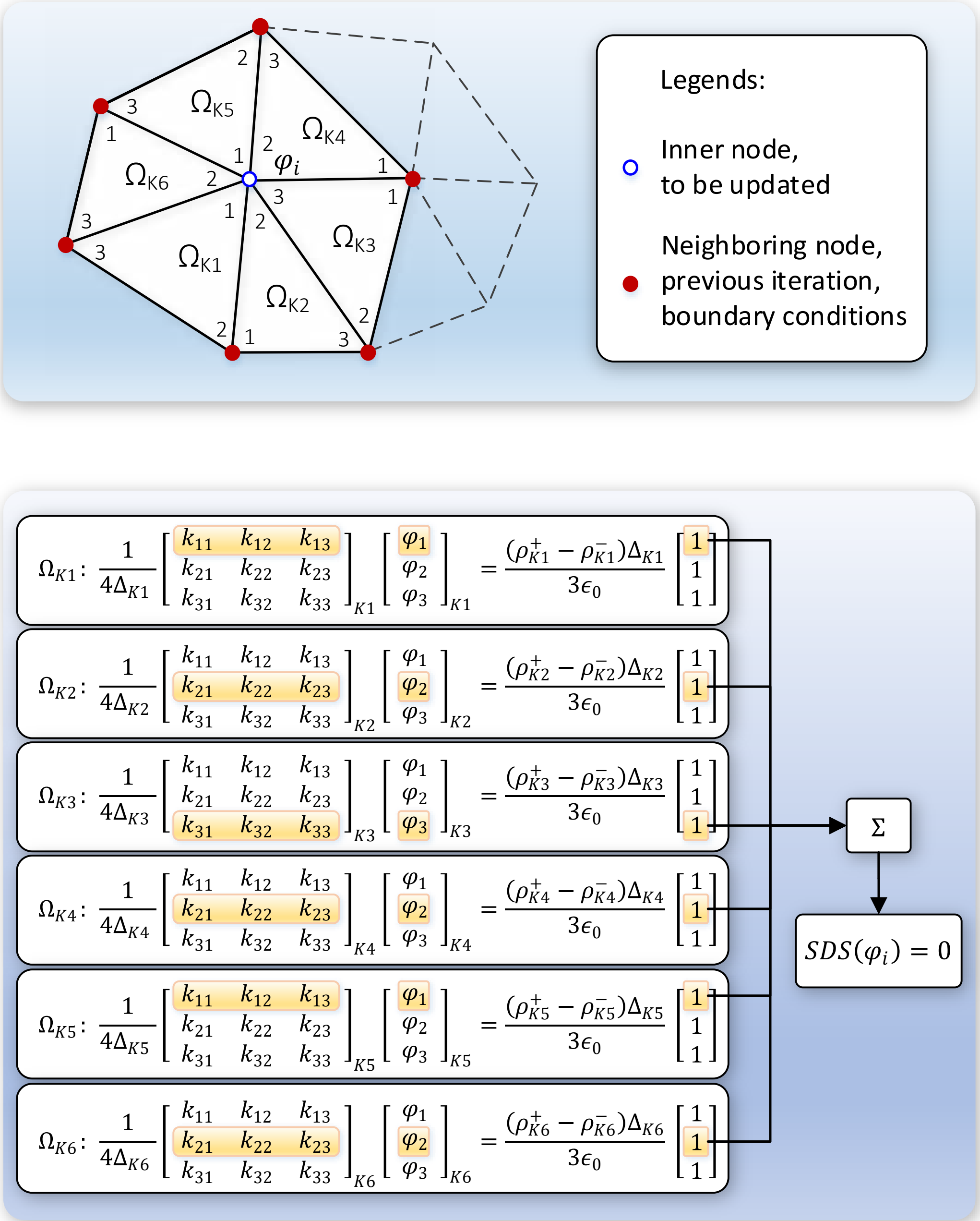

Legends:
Inner node, to be updated
Neighboring node, previous iteration, boundary conditions
Ω_{K1}
Ω_{K2}
Ω_{K3}
Ω_{K4}
Ω_{K5}
Ω_{K6}
φ_i
$\Omega_{K1}: \frac{1}{4\Delta_{K1}}\begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}_{K1}\begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \end{bmatrix}_{K1} = \frac{(\rho_{K1}^+ - \rho_{K1}^-)\Delta_{K1}}{3\epsilon_0}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$
$\Omega_{K2}: \frac{1}{4\Delta_{K2}}\begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}_{K2}\begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \end{bmatrix}_{K2} = \frac{(\rho_{K2}^+ - \rho_{K2}^-)\Delta_{K2}}{3\epsilon_0}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$
$\Omega_{K3}: \frac{1}{4\Delta_{K3}}\begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}_{K3}\begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \end{bmatrix}_{K3} = \frac{(\rho_{K3}^+ - \rho_{K3}^-)\Delta_{K3}}{3\epsilon_0}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$
$\Omega_{K4}: \frac{1}{4\Delta_{K4}}\begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}_{K4}\begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \end{bmatrix}_{K4} = \frac{(\rho_{K4}^+ - \rho_{K4}^-)\Delta_{K4}}{3\epsilon_0}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$
$\Omega_{K5}: \frac{1}{4\Delta_{K5}}\begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}_{K5}\begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \end{bmatrix}_{K5} = \frac{(\rho_{K5}^+ - \rho_{K5}^-)\Delta_{K5}}{3\epsilon_0}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$
$\Omega_{K6}: \frac{1}{4\Delta_{K6}}\begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}_{K6}\begin{bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \end{bmatrix}_{K6} = \frac{(\rho_{K6}^+ - \rho_{K6}^-)\Delta_{K6}}{3\epsilon_0}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$
Σ
$SDS(\varphi_i) = 0$

Figure 3.2: Details of a sub-domain solver for the NDD methodology.

shape function, global system matrix formation by assembling, the imposition of boundary conditions and solution of the linear system of equations. Same as Galerkin FEM, domain decomposition and elemental formulation are required by NDD. However, NDD avoids the assembly in Galerkin FEM and therefore no huge matrix is involved in NDD and it is a matrix-free method. The system of equations for each element in NDD is generated by integrating the product of the residual and weight function over each element and setting the integral to zero. The elemental equations when linear interpolation (assume

φ distribution in element (*e*) can be expressed by $\varphi_{(e)} = N_1\varphi_{1,(e)} + N_2\varphi_{2,(e)} + N_3\varphi_{3,(e)}$ where N_1, N_2, and N_3 are interpolation functions) is used are shown in (3.9).

$$K_{(e)} \begin{bmatrix} \varphi_{1,(e)} \\ \varphi_{2,(e)} \\ \varphi_{3,(e)} \end{bmatrix} = \frac{(\rho^+_{(e)} - \rho^-_{(e)})\Delta_{(e)}}{3\epsilon_0} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \tag{3.9}$$

where $K_{(e)}$ is the coefficient matrix and the calculation of matrix entries for the linear element (triangle) is shown in (3.10). $\Delta_{(e)}$ represents the area of the element (e).

$$k_{ij,(e)} = -\iint_{\Omega_{(e)}} (\nabla N_i) \cdot (\nabla N_j) dxdy \quad (i, j = 1, 2, 3). \tag{3.10}$$

Note that equation (3.9) can also be expressed by a system of equations although matrix form is used here to keep neat. NDD is a matrix-free method because matrix manipulation or calculation is not involved during the solution process. According to [49], one linear equation including the to-be-updated inner node and all its neighboring nodes can be obtained (in Fig. 3.2). The values for all the neighboring nodes can be considered as known boundary conditions when updating the inner node. The linear equation for each node is completely independent, which can be solved by GPUs at the same time. Massive parallelism can be achieved by NDD to solve (3.1). It takes 30 ms to solve Poisson's equation (3.1) for a FEM problem with 8184 nodes by NDD.

3.3.2 Upwind Nodal Charge Conservation

The nodal charge conservation (NCC) guarantees charge conservation law at each finite element node. Besides, each node can be projected to one CUDA core in GPUs to enable solving current continuity equations at each node simultaneously. As a result, the computation efficiency can be improved.

Since the central difference-based iterative scheme usually brings undesired instability [50], the upwind method is utilized. The scenario of the ions migrating in an electric field is quite similar to the situation where a leaf is blown by the wind. The upwind can be defined as the direction the leaf is coming from and so the wind forces the leaf to move downwind. Similarly, in the electric field, we can define the upwind for positive charges or negative charges as the direction the positive or negative charges are coming from. In other words, the upwind is the opposite direction of the charges' velocities.

Derived from the upwind concept, the gradient of positive or negative charge density at one node is mainly determined by the gradient of the positive or negative charge density in the upwind direction. Then, for one specific node, we can define the element in the upwind direction for positive charges (negative charges) as the upwind element for positive charges (negative charges) (Fig. 3.3).

With $\nabla\cdot\boldsymbol{W}(t) = 0$ and equations (3.1) – (3.5), nodal charge conservation equations (3.11) and (3.12) are obtained by the following three steps. One subscript *i* can be added to each

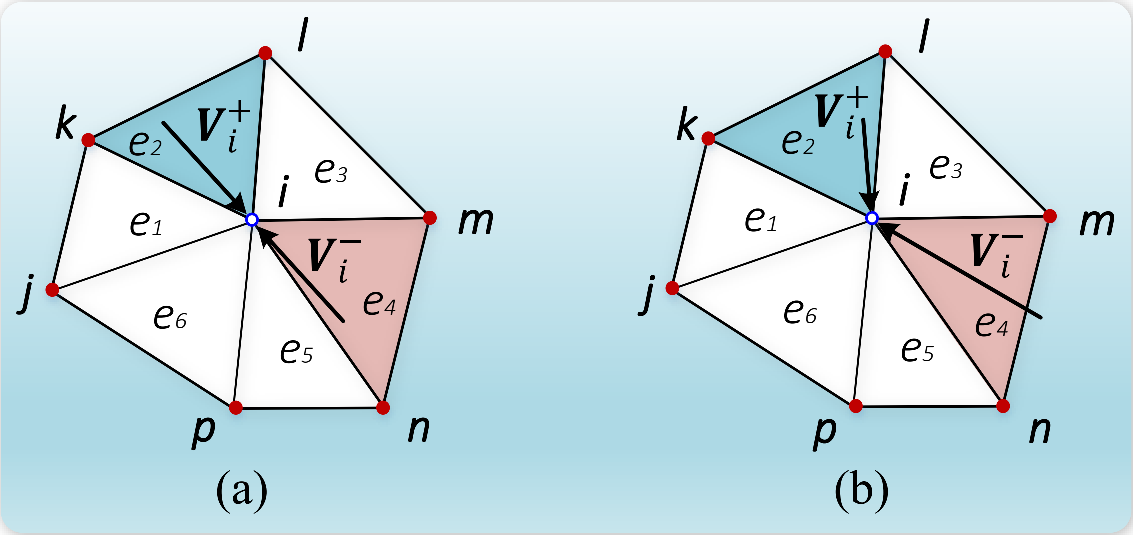


Figure 3.3: Upwind element (e_2) at node i for positive charge density with velocity $\boldsymbol{V}_i^+$ and upwind element (e_4) at node i for negative charge density with velocity $\boldsymbol{V}_i^-$. (a) The wind vector $\boldsymbol{W}$ is zero vector. (b) The wind vector $\boldsymbol{W}$ is nonzero vector.

parameter in (3.2) and (3.3) to represent the corresponding parameter for each node and the validity of the equations remains. Then, the first-order discretized forms of the left-hand side of (3.2) and (3.3) are used to approximate the partial differentials. At last, the upwind difference method is applied to approximate the gradients of the positive and negative charge densities. Subscript *ue* represents the upwind element for the corresponding node's positive charge or negative charge [51].

$$\begin{aligned}\frac{\rho_i^+(t_{n+1})-\rho_i^+(t_n)}{\Delta t} &= -\nabla\cdot\big(\rho_i^+(t_n)\boldsymbol{v}_i^+(t_n)\big)-\frac{R\rho_i^+(t_n)\rho_i^-(t_n)}{e}\\ &= -\nabla\rho_{i,ue}^+(t_n)\boldsymbol{v}_i^+(t_n)-\frac{k^+}{\epsilon_0}(\rho_i^+(t_n))^2+(\frac{k^+}{\epsilon_0}-\frac{R}{e})\rho_i^+(t_n)\rho_i^-(t_n),\end{aligned}\tag{3.11}$$

$$\begin{aligned}\frac{\rho_i^-(t_{n+1})-\rho_i^-(t_n)}{\Delta t} &= \nabla\cdot\Big(\rho_i^-(t_n)\big(-\boldsymbol{v}_i^-(t_n)\big)\Big)-\frac{R\rho_i^+(t_n)\rho_i^-(t_n)}{e}\\ &= -\nabla\rho_{i,ue}^-(t_n)\boldsymbol{v}_i^-(t_n)-\frac{k^-}{\epsilon_0}(\rho_i^-(t_n))^2+(\frac{k^-}{\epsilon_0}-\frac{R}{e})\rho_i^+(t_n)\rho_i^-(t_n).\end{aligned}\tag{3.12}$$

In (3.11) and (3.12), the first term of the right-hand sides consists of the gradient of the positive charges or gradient of negative charges and this is where the upwind concept is applied. Note that the upwind element for positive charges or negative charges are determined by the direction of the charges' velocities, which means knowing the charges' velocities is the prerequisite to figuring out the upwind element. In (3.4) and (3.5), it is obvious that the charges' velocity has two components: the velocity caused by the electric field and the velocity provided by the wind. The wind vector will be a predefined wind distribution. However, the electric field at each node may be discontinuous because of the fact that electric field $\boldsymbol{E}(t)$ is the gradient of $\varphi(t)$ which is numerically estimated by the linear interpolation.

To provide a reasonable electric field $\boldsymbol{E}(t)$ value at each node, the different averaging scheme has been examined, including the angle-weighted scheme, area-weighted scheme, and unweighted-average scheme. It turns out that the unweighted-average scheme gives the best match with the COMSOL Multiphysics$^{\text{TM}}$ result.

From (3.11) and (3.12), the computation of charge density for all the nodes in the domain is independent, which makes upwind NCC achievable. With upwind NCC, the oscillation is removed and the parallelism is guaranteed. Solving current continuity equations for each node at the same time significantly improves the computation efficiency.

3.3.3 Applying Boundary Conditions

This algorithm is suitable to solve the ionized field with both Dirichlet conditions and Neumann conditions. The Dirichlet conditions can be applied by assigning the known desired value to φ at the corresponding nodes but not updating the values at these nodes in the computation process.

However, Neumann conditions cannot be applied directly. Generally, Neumann conditions are described as in (3.8) with the projection of φ in the normal direction. To apply these conditions to the proposed algorithm, the partial differential equation (3.8) is expanded as a product of two vectors in (3.13): one of the vectors is the gradient of parameter φ at the boundary and the other is the outward unit normal vector of the boundary. On the other hand, according to the Galerkin approach to the method of weighted residual, (3.14) can be obtained from (3.1) for each element [52]. In (3.14), it is obvious that the second term on the right-hand side is related to the boundary conditions. When only Dirichlet conditions are applied on the boundary, the integral of the second term is always zero. However, this integral is nonzero when the Neumann condition is applied and the integrated element (*e*) is a boundary element. The boundary element here means the element that has at least one side coinciding with boundary Γ (the global boundary for the entire problem domain). Assuming boundary condition in (3.13) is applied and a boundary element (e_1) has only one side between local node 1 and local node 2 coinciding with global boundary Γ, then the integral relevant to boundary conditions in (3.14) can be simplified as (3.15).

$$\begin{aligned}\frac{\partial\varphi}{\partial n}\bigg|_{\Gamma} =&(\frac{\partial\varphi}{\partial x}\hat{a}_x + \frac{\partial\varphi}{\partial y}\hat{a}_y)\cdot\hat{a}_n\\ =&\frac{\partial\varphi}{\partial x}n_x + \frac{\partial\varphi}{\partial y}n_y\\ =&E_0,\end{aligned} \tag{3.13}$$

where $\hat{a}_x$ and $\hat{a}_y$ are a set of base vectors in the problem domain and they are in the direction of x-axis and y-axis respectively. $\hat{a}_n$ is the outward vector with unit length which is normal to the boundary and $\hat{a}_n = n_x\hat{a}_x + n_y\hat{a}_y$.

$$\begin{aligned}&\epsilon_0 \iint_{\Omega_{(e)}} \frac{\partial N_i}{\partial x}\Big(\sum_{j=1}^{3}\varphi_{j,(e)}\frac{\partial N_j}{\partial x}\Big) + \frac{\partial N_i}{\partial y}\Big(\sum_{j=1}^{3}\varphi_{j,(e)}\frac{\partial N_j}{\partial y}\Big)dxdy \\ &= \iint_{\Omega_{(e)}} N_i(\rho^+_{(e)} - \rho^-_{(e)})dxdy + \epsilon_0 \oint_{\Gamma_{(e)}} N_i(\frac{\partial \varphi}{\partial x}n_x + \frac{\partial \varphi}{\partial y}n_y)dl \qquad (i = 1, 2, 3),\end{aligned} \tag{3.14}$$

where N_i and N_j are the interpolation functions. Since linear interpolation is used, there are three interpolation functions for each element. $\Gamma_{(e)}$ represents the boundary of element (*e*). n_x and n_y are the projections of vector $\hat{a}_n$ normal to the boundary in x and y directions.

$$\oint_{\Gamma_{(e_1)}} N_i(\frac{\partial \varphi}{\partial x}n_x + \frac{\partial \varphi}{\partial y}n_y)dl = \begin{cases} -\frac{E_0 l_{12}}{2} & (i = 1) \\ -\frac{E_0 l_{12}}{2} & (i = 2) \\ 0 & (i = 3) \end{cases}, \tag{3.15}$$

where l_{12} is the length of the side between local node 1 and local node 2 for element (e_1). $\Gamma_{(e_1)}$ is the boundary of element (e_1).

3.4 Case Study and Results

The hybrid ionized field around HVDC and HVAC transmission lines whose transmission tower is configured as in Fig. 3.4 is analyzed. The three-phase 380 kV AC lines are on the left of the tower ordered by phase A, phase B, and phase C from top level downwards. ABC phase sequence is applied and therefore time-varying AC voltages are $v_A = 310.269cos(\omega t)\ kV$, $v_B = 310.269cos(\omega t - 120°)\ kV$ and $v_C = 310.269cos(\omega t + 120°)\ kV$. The bipolar 500 kV DC lines are on the right of the tower ordered by positive polarity, neutral polarity and negative polarity from top level downwards. The voltages of these polarities are $V_P = +500\ kV$, $V_{Neutral} = 0\ kV$ and $V_N = -500\ kV$.

Because of the gravity, conductors will sag between transmission towers, which means the distance between the conductors and ground may change – specifically shorten – in the pathway of transmission lines. Short distance signifies a stronger electric field when the stressed voltage is the same. In this perspective, transmission lines where the largest sag happens have the most severe influence on the environment and thus analyzing the ionized field in this location will be more valuable and instructive. Occupational Safety and Health Administration (OSHA), the minimum clearance distance for transmission lines up to 500 kV is 35 feet (around 10.668 m) [53]. In this case, assuming that 10.8 m above the ground is the height of the lowest-level transmission lines with the largest sag. Then, the heights of the three levels of transmission lines after sag are 10.8 m, 18.8 m, and 26.8 m

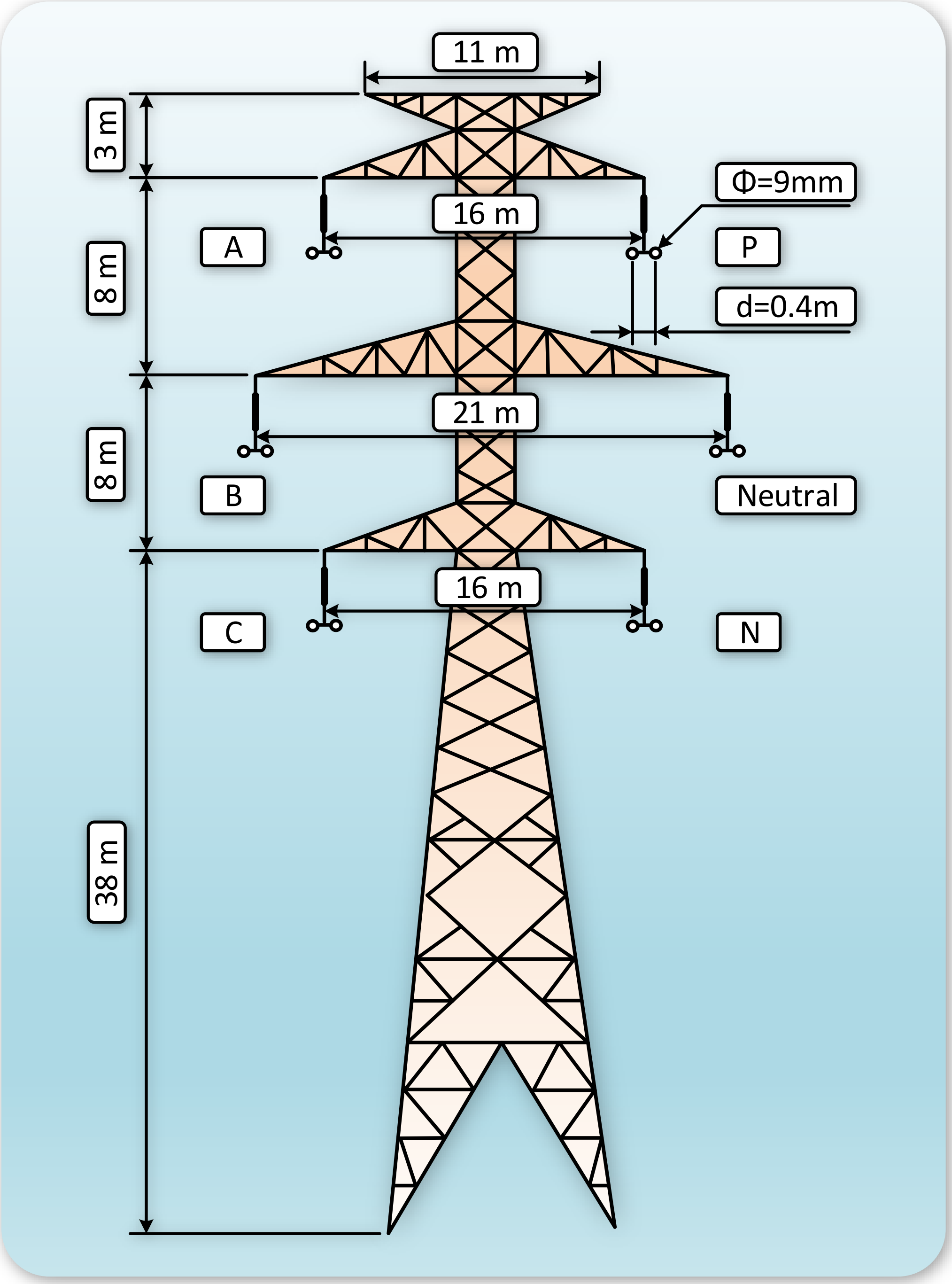


Figure 3.4: Geometry of hybrid AC/DC transmission tower.

(as shown in Fig. 3.5). The area of 100 m * 80 m (W * H) is chosen as problem domain in Fig. 3.5 [54].

To solve the hybrid ionized field in the problem domain, the mesh grid was generated by COMSOL Multiphysics$^{\mathrm{TM}}$ v5.4. By importing the mesh information into the proposed algorithm which was deployed into GPUs with the CUDA Toolkit 9.1 [55], the case can

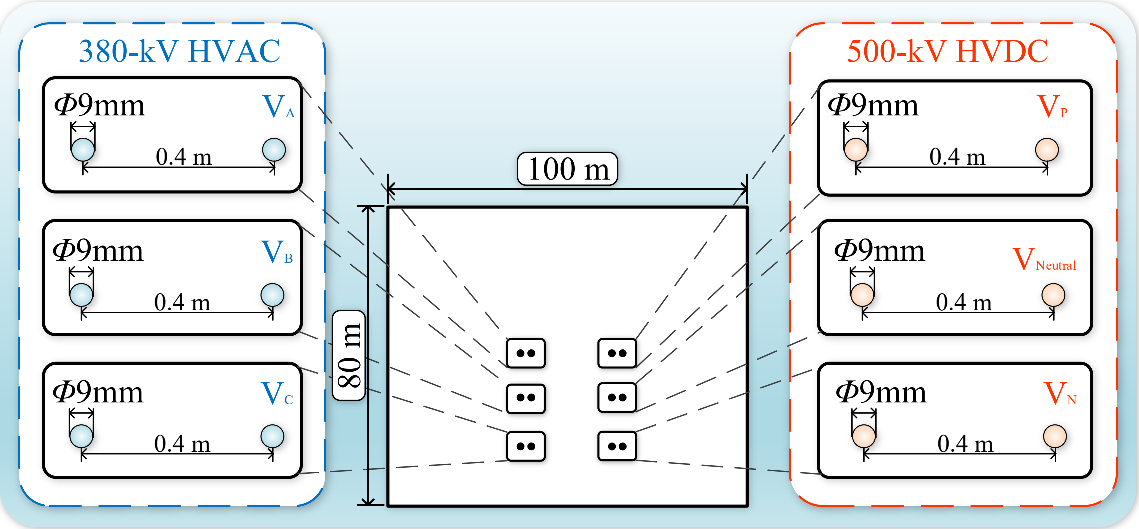


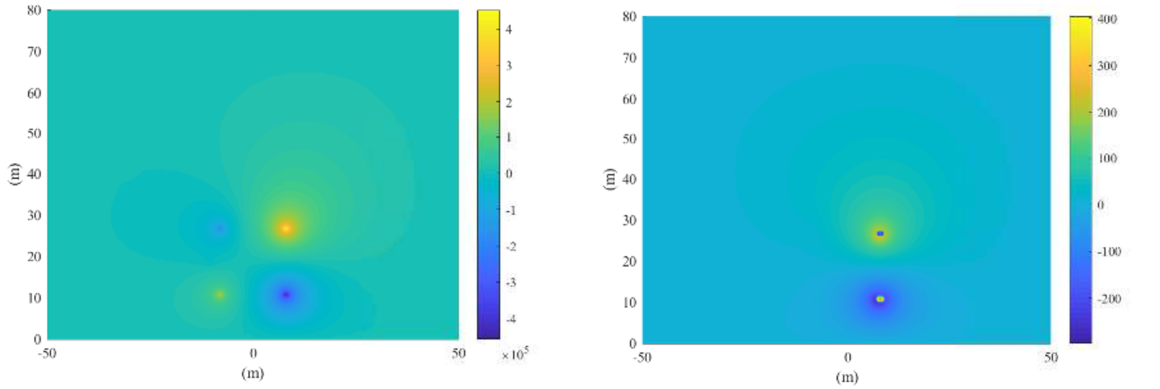
Figure 3.5: Problem domain and line configurations.

be solved. A parallel workstation composed of multicore CPUs and many-core GPUs was utilized to solve this case. The CPUs are dual Intel Xeon E5-2698 v4 CPUs@2.2 GHz, 20 cores each with 128 GB RAM. The GPU is NVIDIA® Tesla® V100-PCIE-16 GB with 5120 CUDA cores [25]. In this case, the time-step Δt is set to 5 μs in order to assure the stability of the algorithm. The stability condition of this algorithm is that the time-step must not exceed the traveling time for both positive and negative ions in one element [51]. In addition, (3.16) is applied to update the charge density on the conductor surfaces [56] and E^+_{onset} (onset electric field on the positive conductor) and E^-_{onset} (onset electric field on the negative conductor) are calculated based on (3.17) [57]. The parameter r in the equations is the radius of the conductor.

$$\begin{aligned} \rho_c^+(t_{n+1}) =& \rho_i^+(t_n)\Big(1+\mu\frac{E_{max}-E^+_{onset}}{E_{max}+E^+_{onset}}\Big), \\ \rho_c^-(t_{n+1}) =& \rho_i^-(t_n)\Big(1+\mu\frac{E_{max}-E^-_{onset}}{E_{max}+E^-_{onset}}\Big), \end{aligned} \tag{3.16}$$

$$\begin{aligned} E^+_{onset} =& 30(1+\sqrt{(9.06/r)}*10^{-2})*10^2\ [kV/m], \\ E^-_{onset} =& (31+0.813/\sqrt{r})*10^2\ [kV/m]. \end{aligned} \tag{3.17}$$

To measure the accuracy and efficiency of the proposed algorithm, the performance of solving Poisson's equation with known charge density distribution by COMSOL Multiphysics™ MUMPS solver and the proposed algorithm is compared. The result comparison of φ distribution at each node is demonstrated in Fig. 3.6 and the mean relative error for all nodes is 0.07%. It turns out that the proposed algorithm is able to provide sufficient accuracy when solving Poisson's equation. In view of speed, the proposed algorithm is 17 times faster than COMSOL Multiphysics™ MUMPS solver when using 40 cores. It is worth mentioning that COMSOL Multiphysics$^{\text{TM}}$ MUMPS solver is state-of-the-art [58]. For this 8184-node FE problem, the proposed algorithm achieves to take 30 ms to solve Poisson's equation because of the massive parallelism.



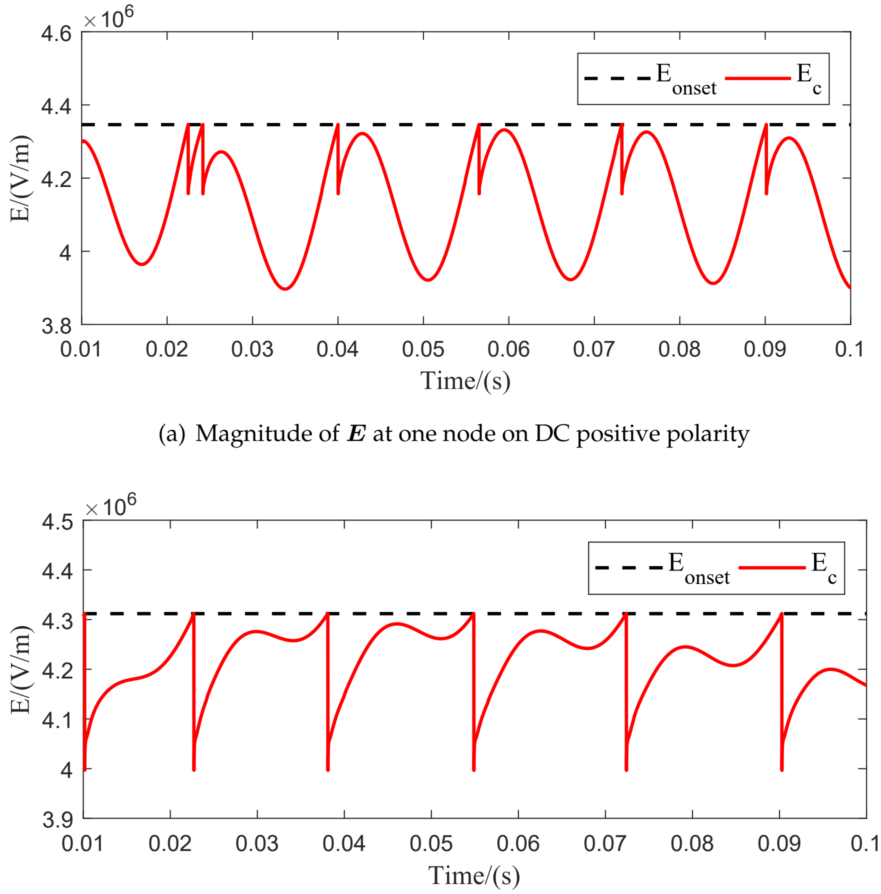
(a) φ distribution calculated by the proposed method

(b) Error map between result from COMSOL$^{\text{TM}}$ and result from the proposed method

Figure 3.6: Result comparison between COMSOL$^{\text{TM}}$ and the proposed method

Fig. 3.7 shows the magnitude of the electric field at one node on the DC positive polarity conductor surface and that of another node on the DC negative polarity conductor surface with respect to time, which can be valuable information when selecting transmission line conductors and designing protection schemes. Obviously, Kaptzov's condition is guaranteed because either of the magnitudes of $\boldsymbol{E}$ does not exceed the corresponding onset value. Fig. 3.7 also illustrates the multiple occurrences of corona on the DC conductors. Each time the magnitude of $\boldsymbol{E}$ reaches the onset value, the corona will happen and charges will be generated around the conductors. These charges have a strong effect on weakening the $\boldsymbol{E}$ around conductors because of the proximity and repellency of like charges and therefore a steep drop of $\boldsymbol{E}$ occurs. Due to the diffusion effect, the charges around the conductors go farther away from the conductors and thus the effect of weakening $\boldsymbol{E}$ becomes weaker, then the $\boldsymbol{E}$ on conductors increases slowly. The corona will not happen until the $\boldsymbol{E}$ reaches the onset value for the next time. Fig. 3.8 demonstrates the same process. We can see two yellow rings in almost all the subplots. One is close to the conductors and the other is far away from the conductors. The second yellow ring is due to the second occurrence of the corona. Besides, Fig. 3.7 demonstrates that the $\boldsymbol{E}$ strength at DC conductor surfaces is composed of DC component and AC component, which is consistent with the statement in [13].

To test the condition with the existence of wind, vector $\boldsymbol{W}$ in equations (3.4) and (3.5) is set to 5 m/s and 10 m/s upward to get the charge density distribution in Fig. 3.8. Refer to (3.4) and (3.5), the wind vector will have the same effect on the speed of the charges no matter it is positive or negative. When the wind with 5 m/s upward speed is applied, the



(a) Magnitude of $\boldsymbol{E}$ at one node on DC positive polarity

(b) Magnitude of $\boldsymbol{E}$ at one node on DC negative polarity

Figure 3.7: Magnitude of $\boldsymbol{E}$ on DC positive and negative polarity

charge distributions for both positive and negative charges are not symmetric anymore. The speed of charges moving upward is enhanced while the speed of charges moving downward is restrained. This effect is more significant when the speed increases to 10 m/s. Based on (3.2) – (3.5), if the speed of charges changes after applying wind, the current density vectors ($\boldsymbol{J}^+$ and $\boldsymbol{J}^-$) will vary and further the space charge density distributions will be influenced as well.

3.5 Summary

In this chapter, the massively parallel processing algorithm consisting of fine-grained NDD and upwind NCC successfully improves the efficiency of solving the time-varying hybrid ionized field which is computationally burdened with the coupling of the Poisson's equation and current continuity equations and repetitive FEM computation. Time-varying distributions of $\varphi(t)$, positive charge density $\rho^+(t)$ and negative charge density $\rho^-(t)$ can be obtained at each time-step to describe the hybrid ionized field around transmission lines

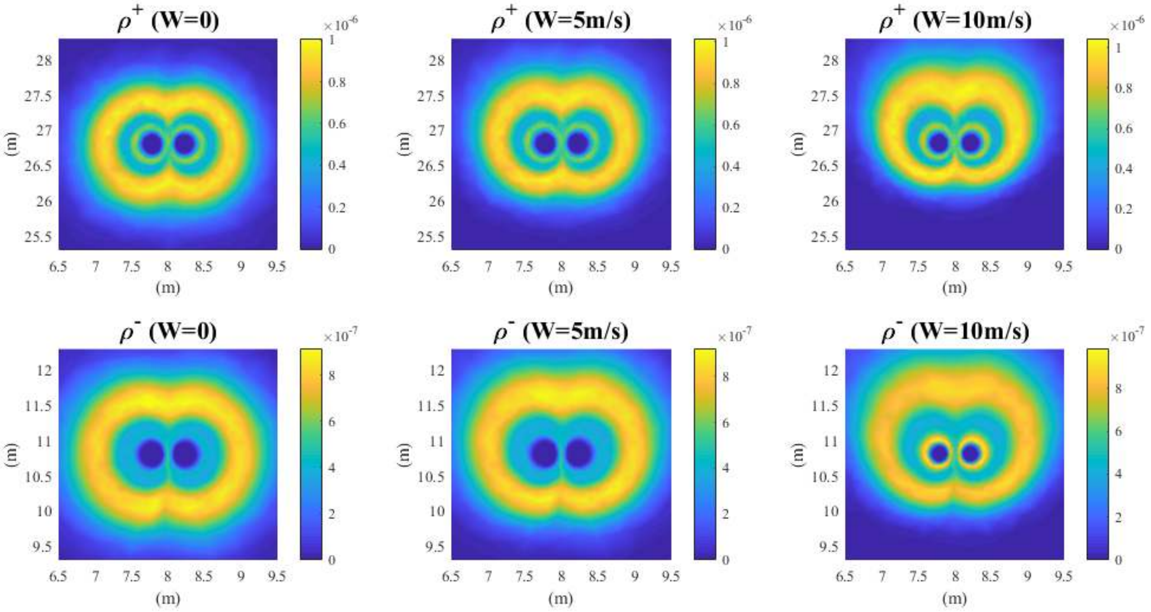


Figure 3.8: Space charge density distribution with different wind velocity $\boldsymbol{W}$.

under different wind conditions. With the intrinsic attribute of NDD and NCC to calculate φ and ρ for each node independently, the computing power of GPUs can be fully explored to accelerate computation speed. It turns out that the Poisson's equation with 8184-node in the case study can be solved in 30 ms, which is 17 times faster than solver in COMSOL Multiphysics$^{\text{TM}}$, implying the execution time for 10^5 time-steps is reduced to 50 mins instead of 14 hours. Besides, the accuracy of the proposed method is also validated against the commercial software with a mean relative error of 0.07%. This efficient algorithm will be helpful to evaluate both the environmental effect and biological effect of the hybrid ionized field even within a short development cycle. Also, the detailed results from this algorithm is valuable for designing conductor selection, tower configuration, and appropriate protection schemes. The wind will impel or suppress the movement of charges when the wind is blowing towards the same direction as the movement of charges or wind is blowing towards the opposite direction of the movement of charges. This algorithm is also capable of performing simulation under different weather conditions to guarantee the robustness of the design.

In the future, this algorithm can be tuned to model other configurations in power systems. In addition, this algorithm can be applied to improve the efficiency of solving similar equations in the semiconductor simulation field [59]. Furthermore, the transient finite-element analysis of semiconductors can be solved by the proposed algorithm efficiently.

4
Time-Stepped Finite-Element Modeling of Three-Phase Transformer for Electromagnetic Transient Emulation on FPGA

4.1 Introduction

In this chapter, a solver which can achieve real-time emulation for a three-phase transformer modeled by finite elements with a time step $70\mu s$ is proposed. An integration between this FE model solver and a field-circuit coupling is also discussed. The accuracy of the transformer solver is validated against the commercial FE simulation tool.

Real-time digital simulation of transformers is essential in order to design an energy-efficient transformer, appropriate protection scheme, and a better control system. The electromagnetic transient study as one of the major perspectives to demonstrate power transformers' performance [60] is mandatory to analyze, otherwise, severe damage might occur to the transformer parts because of inrush current, unexpected harmonics, and overvoltages. Finite elements (FEs) in appropriate element type are generally utilized to discretize a large analysis domain, which helps resolve the geometric irregularity of a transformer and provide a detailed and precise study for the magneto dynamic field [61].

Field-programmable gate arrays (FPGAs) have been investigated and employed in the electrical and electronics field for about three decades since its inception. FPGA was primarily used for prototyping application-specific integrated circuits (ASICs) in the commercial market because of its reprogrammability with no recurring expense and due to its flexibility for fast prototyping with less cost resulted from mistakes. In academia, scholars have successfully explored the potential of FPGA in many fields, such as industrial control systems, power equipment modeling, and so on [62–64]. Benefitting from Moore's law which states that the size of the transistor will shrink exponentially, the FPGA has become much more compact so that it contains millions of gates and high bandwidth memory, which enlarge its capacity by more than a factor of 10000 since its introduction [65]. With the increasing demand for high-speed computation and the growing recognition of hardware acceleration, FPGA becomes favored in real-time simulation [66–68] nowadays due to its huge potential in hardware parallelism and pipelining of user designs.

The conventional lumped models, such as topology-based models and admittance matrix-based models [69, 70], have been widely used in the electromagnetic transient simulation. However, these models are not able to provide information about eddy current distribution and field distribution. Combining Maxwell's equations with the FE model for the transformer, a very detailed analysis can be conducted. But the increased computational burden is non-negligible. Because of its inherent parallel architecture which can implement multiple tasks simultaneously leading to strong computing power, FPGA demonstrates the potential to achieve a real-time simulation for transformer modeled by FEs. Although graphical processing units (GPUs) also have parallel architecture, FPGA outperforms GPUs in the real-time simulation since the developers have access to work much closer to silicon and thus achieve more flexibility and less latency on the FPGA platform. Detailed real-time simulation for a transformer promotes the design and testing of the control system without the need for actual prototyping, leading to cost reduction and higher design reliability.

To simulate the behaviors of the transformer efficiently and accurately, an algorithm taking full advantage of FPGA architecture is required. In this paper, the adaptive transmission-line modeling (TLM) method which decouples the nonlinearity from the linear network and requires fewer iterations to alleviate computation cost, and, the preconditioned conjugate gradient (PCG) which solves a matrix equation in a parallel manner will work together to offer a real-time solution for the FE model of a three-phase transformer and analyze the magneto dynamic field around the transformer with a small time step (70 μs). In addition, high-level synthesis (HLS) technology is employed to mitigate the common concern about hardware difficulties when developing on FPGA which may insinuate a longer development cycle [64].

This chapter is organized as follows: the problem is described, and the governing equations are generated in Section 4.2.1. The proposed adaptive TLM with PCG solver is explained in detail in Section 4.2.2. In Section 4.3, the hardware emulation with deep data pipelining on FPGA is presented. The case studies of the emulation of a three-phase power transformer with and without field-circuit coupling are conducted in Section 4.4. At last, Section 4.5 gives a summary.

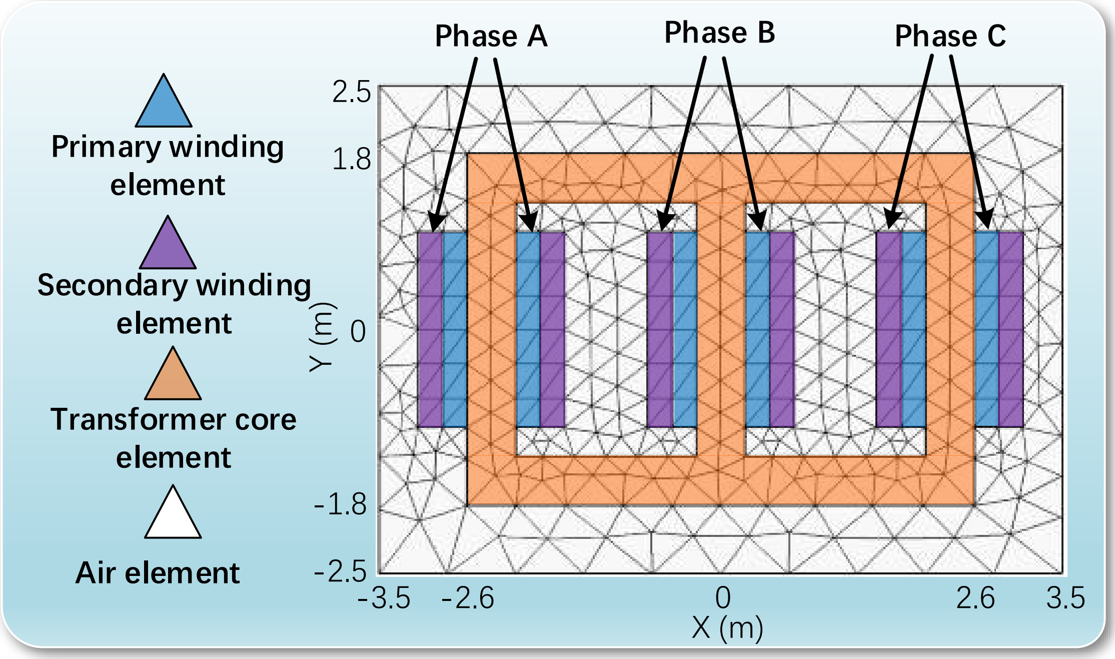


Figure 4.1: 2-D FE model of a three-phase transformer.

4.2 Finite Element Formulation and Solver

4.2.1 Problem Formulation

Consider a magnetic dynamic problem defined on a 2-D domain shown in Fig. 4.1, which is filled by air and a three-phase transformer model. By applying Ampere-Maxell law to obtain the magnetic vector potential distribution at any time in this domain, the following diffusion equation is adopted, in which magnetic vector potential $\mathbf{A}$ and impressed current density $\mathbf{J}_z$ only have z-component in a 2-D problem:

$$\nabla \cdot (\nu \nabla \mathbf{A}) = \sigma \frac{\partial \mathbf{A}}{\partial t} - \mathbf{J}_z, \tag{4.1}$$

where ν is the field-dependent magnetic reluctivity, σ is the electrical conductivity, $\mathbf{J}_z$ is impressed current density and is zero except in the winding zones. The windings in the transformer are modeled by winding zones with constant impressed current density through them.

The FEM is generally utilized to analyze the magneto dynamic field around transformers. Due to greater versatility and ease of implementation, the Galerkin approach becomes popular when deriving finite element equations. First, we subdivide the domain into triangular elements as shown in Fig. 4.1. A^e, the magnetic vector potential over element Ω^e, can be written as:

$$A^e = N_1 A_1^e + N_2 A_2^e + N_3 A_3^e, \tag{4.2}$$

where N_1, N_2, N_3 are shape functions and A_1^e, A_2^e, A_3^e are nodal values of magnetic vector potential at vertices of element Ω^e (see Fig. 4.2). Since the Galerkin approach is a special case of the method of weighted residual, we formulate the residual $\mathbf{R}$ by moving all terms of the differential equation on one side:

$$\mathbf{R} = \nabla \cdot (\nu \nabla \mathbf{A}) - \sigma \frac{\partial \mathbf{A}}{\partial t} + \mathbf{J}_z. \tag{4.3}$$

To seek a satisfying numeric solution in a weighted-integral sense, the integral of the product of the residual and the weighting function over one element is forced to be zero. After simplifying this integral by partial integral and applying natural boundary conditions, the integral of weighted residual over element Ω^e can be expressed as [71]:

$$\int_{\mathbf{\Omega}^e} \nu^e \nabla \mathbf{W}^e \cdot \nabla \mathbf{A}^e d\mathbf{\Omega} + \int_{\mathbf{\Omega}^e} \sigma^e \frac{\partial \mathbf{A}^e}{\partial t} \mathbf{W}^e d\mathbf{\Omega} = \int_{\mathbf{\Omega}^e} \mathbf{J}_z \mathbf{W}^e d\mathbf{\Omega}. \tag{4.4}$$

According to the Galerkin method, A^e is substituted by (4.2) and the weighting functions are set to be the same as shape functions, respectively. As a result, three equations with three unknown magnetic vector potentials at vertices of element Ω^e as elemental equations are obtained [72]:

$$\frac{\nu^e}{4\Delta^e} \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \begin{bmatrix} A_1^e \\ A_2^e \\ A_3^e \end{bmatrix} + \frac{\sigma^e \Delta^e}{12} \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} \frac{\partial A_1^e}{\partial t} \\ \frac{\partial A_2^e}{\partial t} \\ \frac{\partial A_3^e}{\partial t} \end{bmatrix} = \frac{J_z^e \Delta^e}{3} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \tag{4.5}$$

where

$$\begin{aligned} & k_{11} = b_1 b_1 + c_1 c_1, k_{12} = k_{21} = b_1 b_2 + c_1 c_2, \\ & k_{22} = b_2 b_2 + c_2 c_2, k_{23} = k_{32} = b_2 b_3 + c_2 c_3, \\ & k_{33} = b_3 b_3 + c_3 c_3, k_{31} = k_{13} = b_1 b_3 + c_1 c_3. \end{aligned} \tag{4.6}$$

4.2.2 Adaptive TLM with PCG Solver

4.2.2.1 Adaptive Transmission-Line Modeling Method

To deal with the nonlinear relationship between the permeability of the core material and the magnetic flux density, an iterative method such as Newton-Raphson method was traditionally integrated with FEM to solve matrix equations for each time step. Due to the observation that the nonlinearity causes a varying coefficient matrix on the left-hand side of (4.5), a new linear system of equation is formed in each iteration and needs to be solved with the updated Jacobian matrix. The process of calculating new entries in this varying coefficient matrix, assembling this matrix, and solving each new linear system in each iteration makes this integration time-consuming and resource-consuming. Due to the analogy between the node-admittance matrix relative to the equivalent TLM network and finite element matrix, the TLM method is able to solve nonlinear magneto dynamic problems by

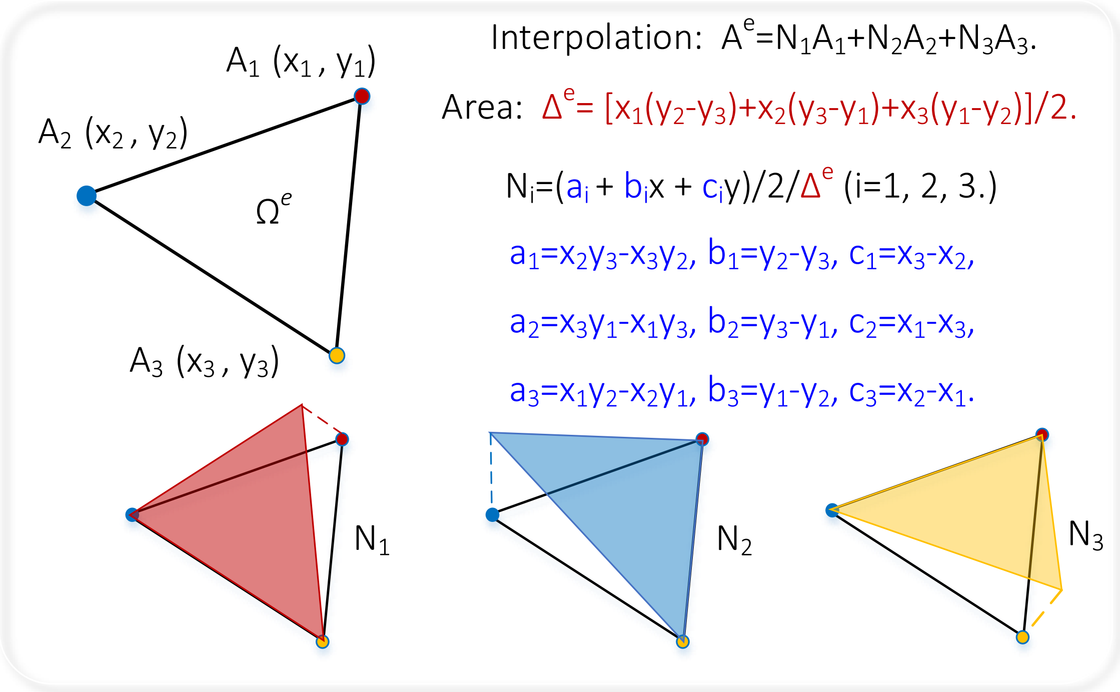

Interpolation: $A^e=N_1A_1+N_2A_2+N_3A_3$.
Area: $\Delta^e= [x_1(y_2-y_3)+x_2(y_3-y_1)+x_3(y_1-y_2)]/2$.
$N_i=(a_i + b_ix + c_iy)/2/\Delta^e$ (i=1, 2, 3.)
$a_1=x_2y_3-x_3y_2$, $b_1=y_2-y_3$, $c_1=x_3-x_2$,
$a_2=x_3y_1-x_1y_3$, $b_2=y_3-y_1$, $c_2=x_1-x_3$,
$a_3=x_1y_2-x_2y_1$, $b_3=y_1-y_2$, $c_3=x_2-x_1$.
A_1 (x_1, y_1)
A_2 (x_2, y_2)
A_3 (x_3, y_3)
Ω^e
N_1
N_2
N_3

Figure 4.2: FE and the interpolation functions for the Galerkin FEM.

decoupling the nonlinear elements from linear system [73]. With the application of TLM method, the equation (4.5) can be solved without changing coefficient matrices which can be significantly easier to implement.

TLM method was a popular technique to analyze and simulate wave propagation with Huygens' principle as a foundation at first [74]. In 1980, the TLM method was introduced as a new, accurate, and efficient approach to solving nonlinear lumped electrical networks [75]. Decoupling of linear components and nonlinear components was achieved by TLM technique in this chapter and thus the computation cost was reduced. The TLM method was employed by three steps:

1. Create a linear network with all linear components inside and connected externally to nonlinear components via ports;

2. Replace capacitors and inductors by TLM stubs and connecting nonlinear resistors to the linear network by TLM links;

3. Connect a Thevenin equivalent circuit composed of a voltage generator whose voltage equals to twice the value of the incident pulses on the transmission line and a resistor with the characteristic resistance of the corresponding transmission line to each port after removing TLM links and stubs.

After this modeling, a nodal admittance matrix $\mathbf{Y}$, a current vector of independent current generators $\mathbf{J}_s$, a voltage vector of voltage generators $\mathbf{E}_s$, and a vector of incident pulses at time t is $\mathbf{V}_i(t)$ in (4.7) can be obtained and then the vector of the nodal voltages $\mathbf{V}_d(t)$ in the whole network can be solved. Because of the fact that the nodal voltage must be equal to the sum of incident pulse and reflected pulse at the node, and the known relationship that the incident pulse in the next time step is the product of reflection coefficient K_c and the reflected pulse in this time step, the incident pulse at one node can be updated according to (4.8). Note that the incident pulses are injected into the linear network and thus propagate towards the ports. By injecting new incident pulses and solving (4.7), nodal voltages at the next time step can be obtained. Due to the extraction of the nonlinearity in this network, computing the inversion of the admittance matrix once is sufficient when the algorithm advances to the next time step, and the computation burden can be significantly diminished.

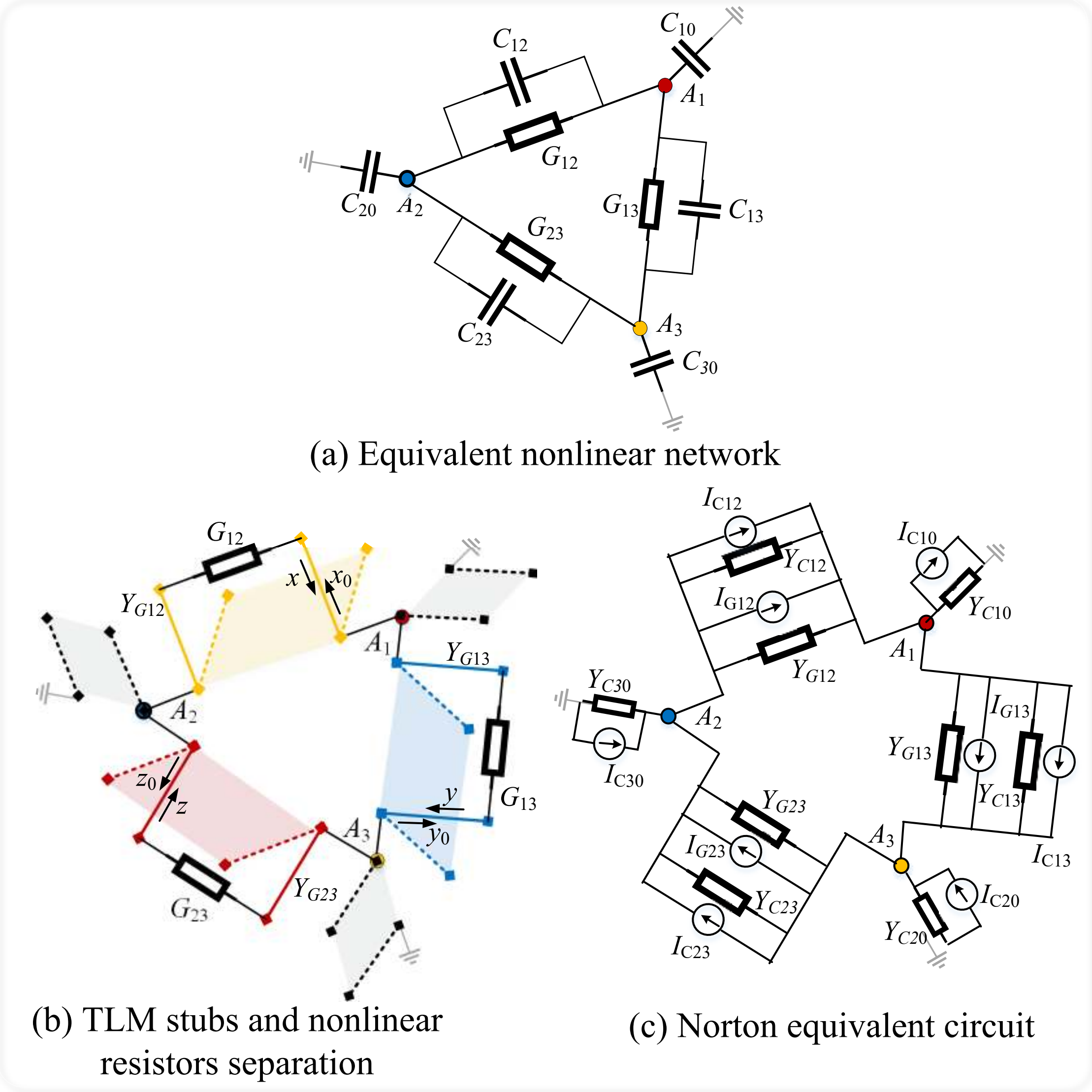
$$\mathbf{Y}\mathbf{V}_d(t) = \mathbf{J}_s - \mathbf{Y}\mathbf{E}_s - 2\mathbf{Y}\mathbf{V}_i(t), \tag{4.7}$$

$$V_i(t+\Delta t) = K_c(V_d(t) - V_i(t)). \tag{4.8}$$

After unveiling the analogy between the nodal admittance matrix and a finite element matrix, Deblecker and Lobry successfully applied the conventional TLM method into FEM for nonlinear eddy-current problems [73]. Comparing (4.5) with (4.7), it appears that the elemental equation (4.5) defines an electrical network with nonlinear resistors, capacitors, and the same topology as a triangle when magnetic vector potential is modeled as electric potential in Fig. 4.3 (a). Following the process of using the TLM method, these nonlinear resistors and capacitors are separated from linear components by transmission lines shown in Fig. 4.3 (b) and then the Norton equivalent circuit can be derived in Fig. 4.3 (c). For ease-of-use and consistency, Norton equivalent circuit instead of Thevenin equivalent circuit is used in FEM problem. The values of all these components in Fig. 4.3 are described in (4.9).

$$\begin{aligned}
G_{12} &= -\frac{\nu^e}{4\Delta^e}(b_1b_2 + c_1c_2), Y_{G12} = -\frac{\nu_g^e}{4\Delta^e}(b_1b_2 + c_1c_2),\\
G_{13} &= -\frac{\nu^e}{4\Delta^e}(b_1b_3 + c_1c_3), Y_{G13} = -\frac{\nu_g^e}{4\Delta^e}(b_1b_3 + c_1c_3),\\
G_{23} &= -\frac{\nu^e}{4\Delta^e}(b_2b_3 + c_2c_3), Y_{G23} = -\frac{\nu_g^e}{4\Delta^e}(b_2b_3 + c_2c_3),\\
C_{12} = C_{13} = C_{23} &= -\frac{\sigma^e\Delta^e}{12},\\
Y_{C12} = Y_{C13} = Y_{C23} &= -\frac{\sigma^e\Delta^e}{6\Delta t},\\
C_{10} = C_{20} = C_{30} &= \frac{4\sigma^e\Delta^e}{12},\\
Y_{C10} = Y_{C20} = Y_{C30} &= \frac{4\sigma^e\Delta^e}{6\Delta t},
\end{aligned} \tag{4.9}$$

where ν^e is not priori known for nonlinear resistors. ν_g^e is a guess value for the reluctivity for element e, which should be as close as possible to the values taken by the nonlinear



(a) Equivalent nonlinear network

(b) TLM stubs and nonlinear resistors separation

(c) Norton equivalent circuit

Figure 4.3: Process of applying TLM method to one finite element.

resistor. ν^e_g, the guess value for ν^e is used to assembly the global matrix on the left-hand side of (4.5). The characteristic impedance of transmission line for a capacitor with capacitance C is $\frac{\Delta t}{2C}$ with Δt for simulation time step [75]. The characteristic impedance of transmission line for a nonlinear resistor is an arbitrary guess value and the closer the guess value is, the fewer the required TLM iteration is [75].

With initial incident pulses and components with values in (4.9), the nodal voltages can be solved by (4.5) using the conventional TLM method. However, multiple TLM iterations are generally required before convergence due to the mismatch of ν^e and ν^e_g. It is also worth mentioning that (4.8) is not applicable here since the reflection coefficient cannot be determined without known resistance of the nonlinear resistor. Another equation is utilized to discover the incident pulses for the next time step. For a nonlinear resistor governed by the equation: $I = f(U)$. The following relationship is guaranteed based on the transmission line theory:

$$\frac{V_r(t) - V_i(t + \Delta t)}{Z_0} = f(V_r(t) + V_i(t + \Delta t)), \tag{4.10}$$

where V_r and V_i represents the reflected pulses and injected pulses for the network, respectively.

Therefore, three coupled equations need to be solved to update incident pulses in each element in FEM:

$$\begin{aligned} Y_{G12}(x_0 - x) &= G_{12}(x + x_0), \\ Y_{G13}(y_0 - y) &= G_{13}(y + y_0), \\ Y_{G23}(z_0 - z) &= G_{23}(z + z_0), \end{aligned} \tag{4.11}$$

where x, y, z represents the incident pulses and x_0, y_0, z_0 are reflected pulses.

Taking a closer look at (4.5) and (4.7), it is obvious that the coefficient matrix on the left-hand side in (4.5) remains unchanged in (4.7) after utilizing this conventional TLM method; meanwhile, there is one more term determined by the incident pulses on the right-hand side of (4.7). Instead of updating the Jacobian matrix in the Newton-Raphson method, the term related to the incident pulses will vary during each TLM iteration, which requires a much simpler operation.

Two main advantages can be gained with the application of TLM method: the nonlinear elements are decoupled from a linear system and solved individually; the equation (4.5) can be solved without changing admittance matrices. Please note that computing the inverse of admittance matrices once is sufficient with the unchanged admittance matrices. However, too many times of TLM iteration and the unguaranteed stability are commonplace concerns about the conventional TLM method. To overcome this bottleneck, an adaptive TLM method is considered [76]. The main idea of the adaptive TLM method is to use a closer guess value for the unknown to alleviate the number of TLM iteration.

In this transformer model, ν^e is time-varying, since the time-varying excitation current in the transformer produces a time-varying field and the ν^e of the nonlinear material in the transformer is field-dependent. Therefore, the difference between $\nu^e(t)$ and constant initial guess ν_g^e will fluctuate and the number of TLM iteration before convergence will vary from tens to hundreds (see Fig. 4.4).To follow the adaptive TLM method and make a closer guess, the value of ν^e at time t, which can be extracted from the solution at time t, is utilized to guess the real value of ν^e at time $(t + \Delta t)$. In Fig. 4.4, the required number of TLM iterations is significantly decreased because the real value of ν^e at time t is a much closer guess for ν^e at time $(t + \Delta t)$ than the constant initial guess. This adaptive TLM method is successfully utilized in this work to diminish the number of TLM iterations.

The adaptive TLM method not only possesses the merit of a simplified solving process and less intensive computation caused by decoupling nonlinear components from the network, but also requires fewer TLM iterations. But it is worth mentioning that the fewer

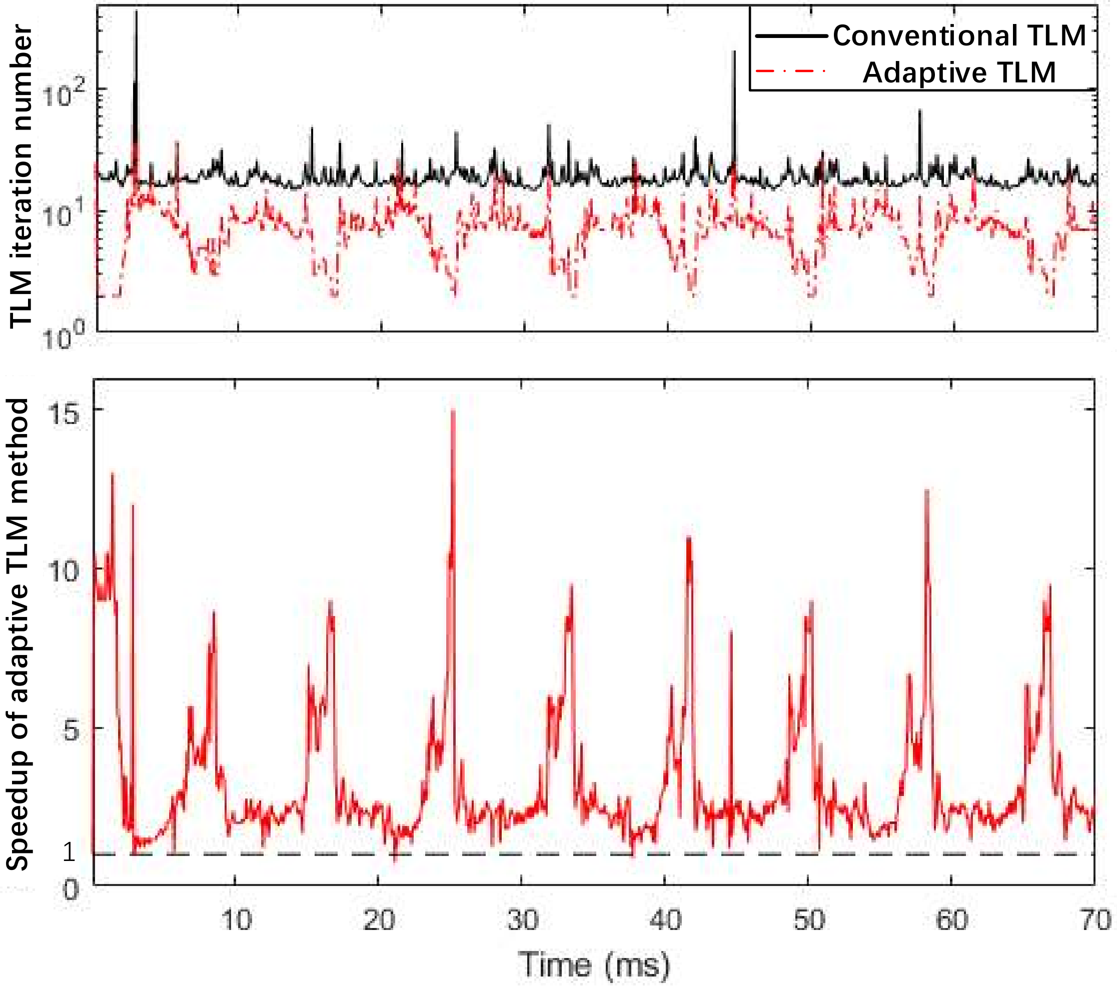


Figure 4.4: By injecting the same magnetizing current, the required number of TLM iterations with unchanged admittance matrix, and required number of TLM iterations with updated adaptive admittance matrix.

TLM iterations are at the cost of reassembling the admittance matrix before the first TLM iteration for each timestep. As a result, one inverse operation will be required before each timestep if the matrix equation (4.5) is solved by direct methods. The computation burden because of matrix reassembly and multiple inverse operation may jeopardize the efficiency of the adaptive TLM method. To guarantee the efficiency of the adaptive TLM method, a comprehensive selection of matrix equation solver is important.

4.2.2.2 Preconditioned Conjugate Gradient Algorithm

While achieving space discretization by FEs, backward Euler method is also exploited in (4.5) in order to discretize time. Then, (4.12) in element Ω^e with only three unknowns ($A_1^e(t+\Delta t)$, $A_2^e(t+\Delta t)$, and $A_3^e(t+\Delta t)$) at time $(t+\Delta t)$ is obtained. Note that the values of $A_1^e(t)$, $A_2^e(t)$, and $A_3^e(t)$ in this equation are acquired from the solution at time t and they are known before computing (4.12):

$$\begin{aligned}\frac{\nu^e}{4\Delta^e}\begin{bmatrix}k_{11} & k_{12} & k_{13}\\ k_{21} & k_{22} & k_{23}\\ k_{31} & k_{32} & k_{33}\end{bmatrix}\begin{bmatrix}A_1^e(t+\Delta t)\\ A_2^e(t+\Delta t)\\ A_3^e(t+\Delta t)\end{bmatrix}&\\ +\frac{\sigma^e\Delta^e}{12}\begin{bmatrix}2 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 2\end{bmatrix}\begin{bmatrix}A_1^e(t+\Delta t)\\ A_2^e(t+\Delta t)\\ A_3^e(t+\Delta t)\end{bmatrix}&\\ =\frac{J_z^e(t+\Delta t)\Delta^e}{3}\begin{bmatrix}1\\ 1\\ 1\end{bmatrix}+\frac{\sigma^e\Delta^e}{12}\begin{bmatrix}2 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 2\end{bmatrix}\begin{bmatrix}A_1^e(t)\\ A_2^e(t)\\ A_3^e(t)\end{bmatrix}&. \qquad (4.12)\end{aligned}$$

With (4.12) for each element in a discretized subdomain, a global matrix system with the size of the number of nodes in the whole domain is formed after the assembly. So far, this whole domain is modeled as a matrix system and the solution for this problem can be solved by matrix equation solvers.

The approaches to solving matrix equations can be categorized into two groups: direct methods and iterative methods [77]. The direct methods such as Gaussian elimination and LU decomposition solve one matrix equation by a set of sequential operations, which makes it unsuitable to be implemented on FPGA. On the contrary, the Conjugate Gradient (CG) algorithm which is one of the iterative methods can be highly parallel processed because of the potential of parallelly implementing matrix-vector multiplication, the core operation in the CG algorithm. The research in [33] demonstrates that a notable speedup can be achieved after implementing CG on FPGA. Besides, a high volume of digital signal processing blocks (DSPs) capable of efficiently processing the multiplier-accumulator operation that constitutes a great proportion of computation operations in CG algorithm are integrated into many modern FPGA. Without losing the high parallelism of the CG algorithm, a PCG algorithm with a featured matrix to increase the convergence rate is utilized to optimize the simulation process.

4.3 Hardware Implementation on Multiple FPGAs

Although the increasing complexity and capability result in an increment in design efforts, FPGA vendors have taken efforts to ease this stress from the designers. Developers can use optimization directives to alter the default behavior of the internal logic and modify data access patterns by HLS. The optimized hardware implementation is deployed on the Xilinx® Virtex® UltraScale+ HBM VCU128-ES1 board with the XCVU37P-fsvh2892-2L-e FPGA.

Based on the adaptive TLM method and PCG algorithm, a solver is designed and implemented on FPGAs. The detailed real-time hardware emulation is demonstrated in Fig. 4.5 and Fig. 4.6 with a state diagram for this finite state machine and block connections.

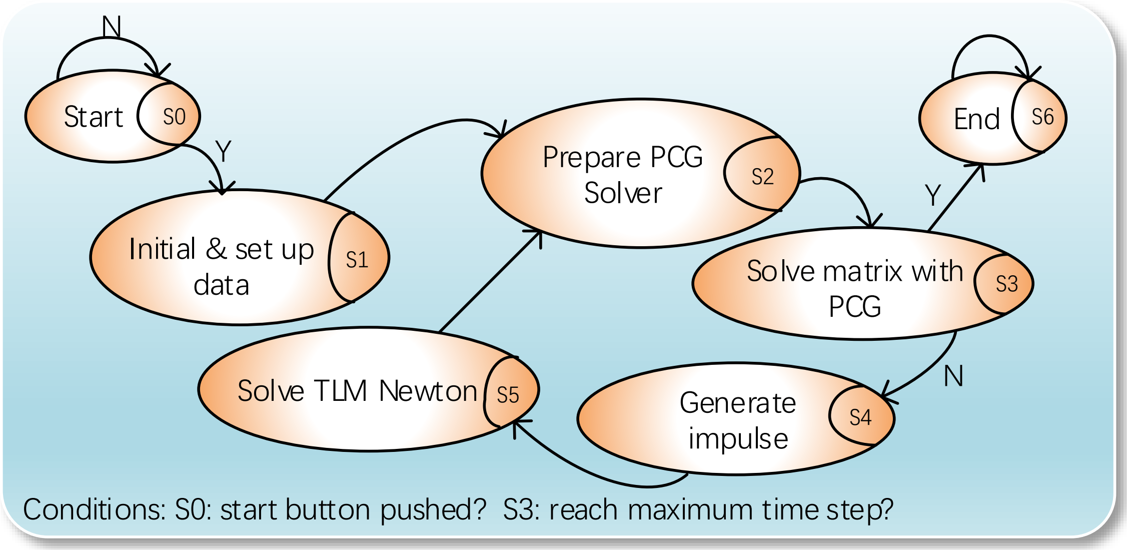


Figure 4.5: Finite state machine of the adaptive TLM with PCG solver.

In Fig. 4.6, there are eight hardware blocks optimized by directives such as loop unrolling, pipelining, and array partitioning. One more note about Fig. 4.6 is that four blocks including TLM Current Term, Excitation Current Term, Eddy Current Term, and Admittance Matrix Updater are started and processed simultaneously due to the data independence. Further details about these blocks are given as follows. The parallelism and pipelining of FPGA are fully utilized with the consideration of the available hardware resources and problem size.

1. *TLM Current Term*: The current caused by the incident pulses in the TLM method for each nonlinear element are assembled to each node. The inputs and the static matrices based on mesh information in the block are partitioned to improve access to the data. The loop unrolling and pipelining are applied to different levels of loops to achieve hardware parallelism and balance between computation time and resource allocation.

2. *Excitation Current Term*: This block is used to calculate the first vector on the right-hand side of (4.12). With the known excitation current in three primary windings, the projected excitation current on each node is obtained in this block as the output. The loop unroll is utilized to create multiple collections of operations to compute with greater hardware parallelism and the static arrays in the block are partitioned to support the unrolled loop.

3. *Eddy Current Term*: This block generates the result of the second term in (4.12), which is the product of the magnetic vector potentials solved at the last time step and the prior known matrix stored as a static array. A matrix-vector multiplication is imple-

mented, and the sum is acquired by using the tree adder algorithm with a complexity of $O(log_2 N)$. Besides, high-speed but low-resource utilization is guaranteed by pipelining.

4. *RHS Generator*: The right-hand side (RHS) of the matrix equation is determined by the current generated by incident pulses in the TLM method, the excitation current, and the eddy current. By summing up all the aforementioned currents for each node, the RHS can be calculated. The sum for each node is computed by an unrolled loop.

5. *Admittance Matrix Updater*: This is one of the critical blocks to implement the adaptive TLM method. Based on the calculated field-dependent magnetic reluctivity ν^e for each element at the previous time step, a new admittance matrix, and an updated preconditioned matrix are generated from this block and the matrices are transmitted in a sparse manner.

6. *PCG Solver*: This is a block to realize an efficient and high-performance PCG algorithm. The inputs are from the Admittance Matrix Updater block and the RHS Generator block. The static arrays containing the mesh information are partitioned to supply more efficient read operation. A block that accomplishes an efficient summation using the tree structure is built for and reused in this PCG Solver block. Besides, the loop unrolling and pipelining are the main optimization directives in this block. The magnetic vector potentials at the current time step are calculated by this block.

7. *Impulse Generator*: According to the updated magnetic vector potentials and the incident pulses, the reflected pulses from the linear network to each nonlinear element can be developed. They are calculated in parallel in an unrolled loop.

8. *TLM Newton Solver*: The matched ν^e at the current time step and the incident pulses for the next time step can both be calculated in this block. For each nonlinear element, a 3×3 matrix equation is solved by Newton-Raphson method. Pipelining technique and unroll loops are exploited to treat multiple nonlinear elements efficiently.

Though an FPGA is very powerful nowadays, the resources on one FPGA may still be a restriction for a complex design. For this solver, the bottleneck is the available number of DSPs on one board due to the huge demand for addition and multiplication operations. To use the minimum number of boards while not jeopardizing the performance of the solvers, this solver is deployed on two Xilinx® Virtex® UltraScale+ HBM VCU128-ES1 boards connected via QSFP interfaces which provide high-speed transmission. To utilize two boards efficiently and in balance, the resource utilization on each board, the amount of transmission data, and communication process are all considered to decide how to allocate blocks on these two boards.

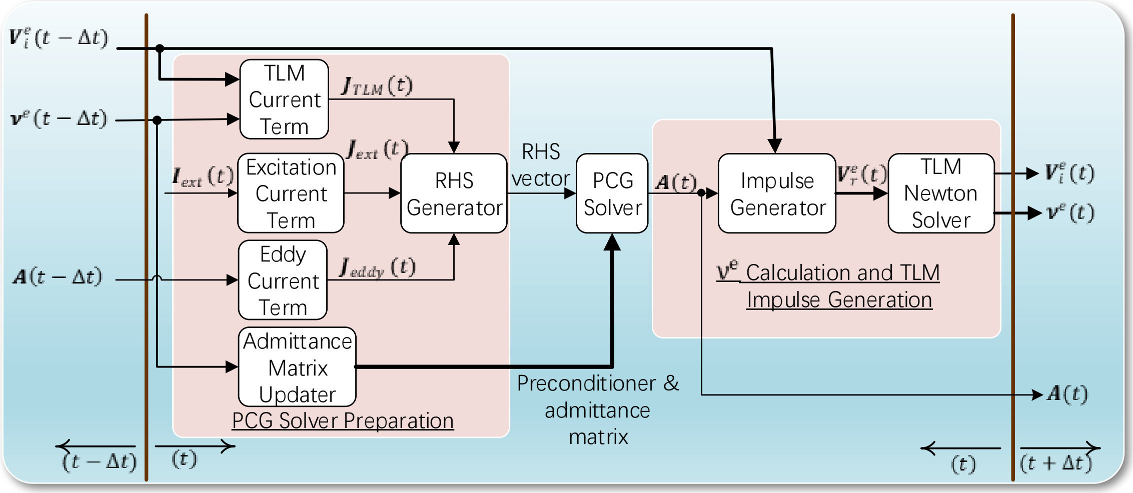

$\boldsymbol{V}_i^e(t-\Delta t)$
$\boldsymbol{v}^e(t-\Delta t)$
$\boldsymbol{A}(t-\Delta t)$
TLM Current Term
$\boldsymbol{J}_{TLM}(t)$
$\boldsymbol{I}_{ext}(t)$
Excitation Current Term
$\boldsymbol{J}_{ext}(t)$
Eddy Current Term
$\boldsymbol{J}_{eddy}(t)$
Admittance Matrix Updater
PCG Solver Preparation
RHS Generator
RHS vector
PCG Solver
Preconditioner & admittance matrix
$\boldsymbol{A}(t)$
Impulse Generator
$\boldsymbol{V}_r^e(t)$
TLM Newton Solver
$\boldsymbol{V}_i^e(t)$
$\boldsymbol{v}^e(t)$
$\mathrm{v^e}$ Calculation and TLM Impulse Generation
$\boldsymbol{A}(t)$
$(t-\Delta t)$
(t)
(t)
$(t+\Delta t)$

Figure 4.6: Paralleled and pipelined Hardware architecture for adaptive TLM with PCG solver.

4.4 Hardware Emulation Scenarios

4.4.1 Scenario I: Current-Excited Emulation

A three-phase power transformer rated at 40 kV/200 kV was modeled to verify the proposed solver. The geometrical parameters are given in Appendix A. The FE mesh consisting of 385 nodes and 728 elements for this transformer is shown in Fig. 4.1. The time-varying impressed current on the primary windings from left to right are $I_a(t) = 1000sin(120\pi t)A$, $I_b(t) = 1000sin(120\pi t - 120°)A$, and $I_c(t) = 1000sin(120\pi t + 120°)A$ respectively. The B-H curve of iron core expressed as equation (4.13) is employed and $\Delta t = 1s/60/238 \approx 70\mu s$ is used, implying a 238*60 = 14280 sampling frequency and harmonics in frequency as high as 7.14 kHz could be captured:

$$H = \begin{cases} 800B & \text{if } 0 < B < 0.6, \\ 800B + 10^5(B-0.6)^3 & \text{if } B > 0.6. \end{cases} \tag{4.13}$$

The hardware emulation of this transformer is performed on two Xilinx® Virtex® UltraScale+ HBM VCU128-ES1 boards connected by two QSFP interfaces (see Fig. 4.7), each of which has four channels. The main hardware resources of the Xilinx® XCVU37P FPGA are as follows: 9024 DSP48E slices, 2607360 flip-flops, 1303680 look-up tables, and 4032 BRAMs. The hardware utilization and the latency for each block are shown in Table 4.1. By fitting the available hardware on each board, balancing computation workload, and minimizing the amount of transmitted data, the blocks RHS Generator, PCG Solver, and Impulse Generator are allocated on Board1 and the other blocks are allocated on Board2. The subtotals of resource utilization and execution time for each board are also given in Table 4.1. As a small time step $70\mu s$ is utilized, one TLM iteration is sufficient for reasonable

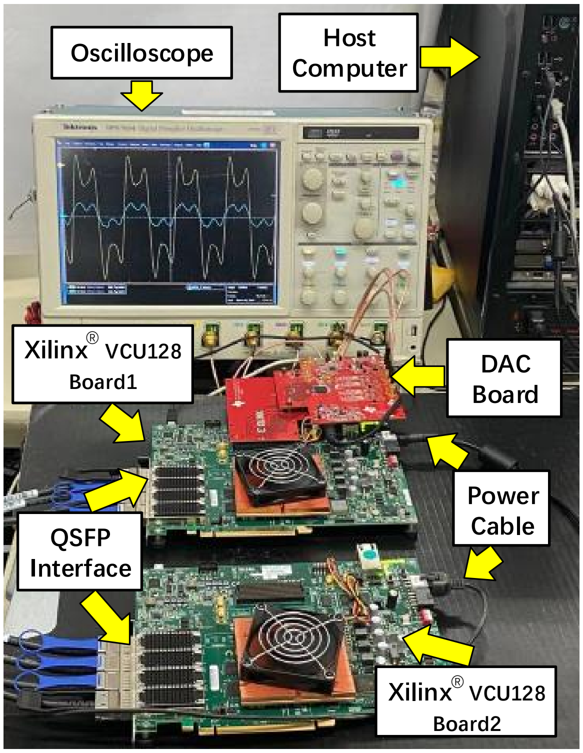


Figure 4.7: Hardware configuration for the transient emulation by proposed algorithm on multiple FPGAs.

accuracy. To make sure the design works after all these hardware blocks are interconnected, 100 MHz is selected as the clock frequency. The data transmission delay between two boards has been measured and it is about $17\mu s$. According to Table 4.1, the execution time for one time step is $(616 + 3 + 2820 + 112 + 1185)/100M + 17\mu s = 64.36\mu s$ which is smaller than one time step and therefore, a real-time execution is achieved. Comparisons between Comsol® results and real-time solver results are shown in Fig. 4.8 for field quantities.

As a critical block to achieve the adaptive TLM method, the Admittance Matrix Updater block uses up almost half of DSP resources on FPGA Board2, which is the third-most DSP-consuming block in this solver. Considering the DSP resource is very precious in this solver, how does the adaptive TLM method better than the conventional TLM method? The advantage of using the adaptive TLM method will be obvious after the following calculation. In Fig. 4.4, the average TLM iteration number is about 20 while the average adaptive TLM iteration number is about 7, which leads to 3× speedup on average. Assume we use the conventional TLM method, the execution time for one TLM iteration will be about 5 μs less, but with the increased number of TLM iterations, the execution time of conventional TLM is still significantly longer than the proposed adaptive TLM method. As a result, the conventional TLM method will be unable to achieve real-time simulation. The trade-off between resource utilization and execution time has been carefully considered in

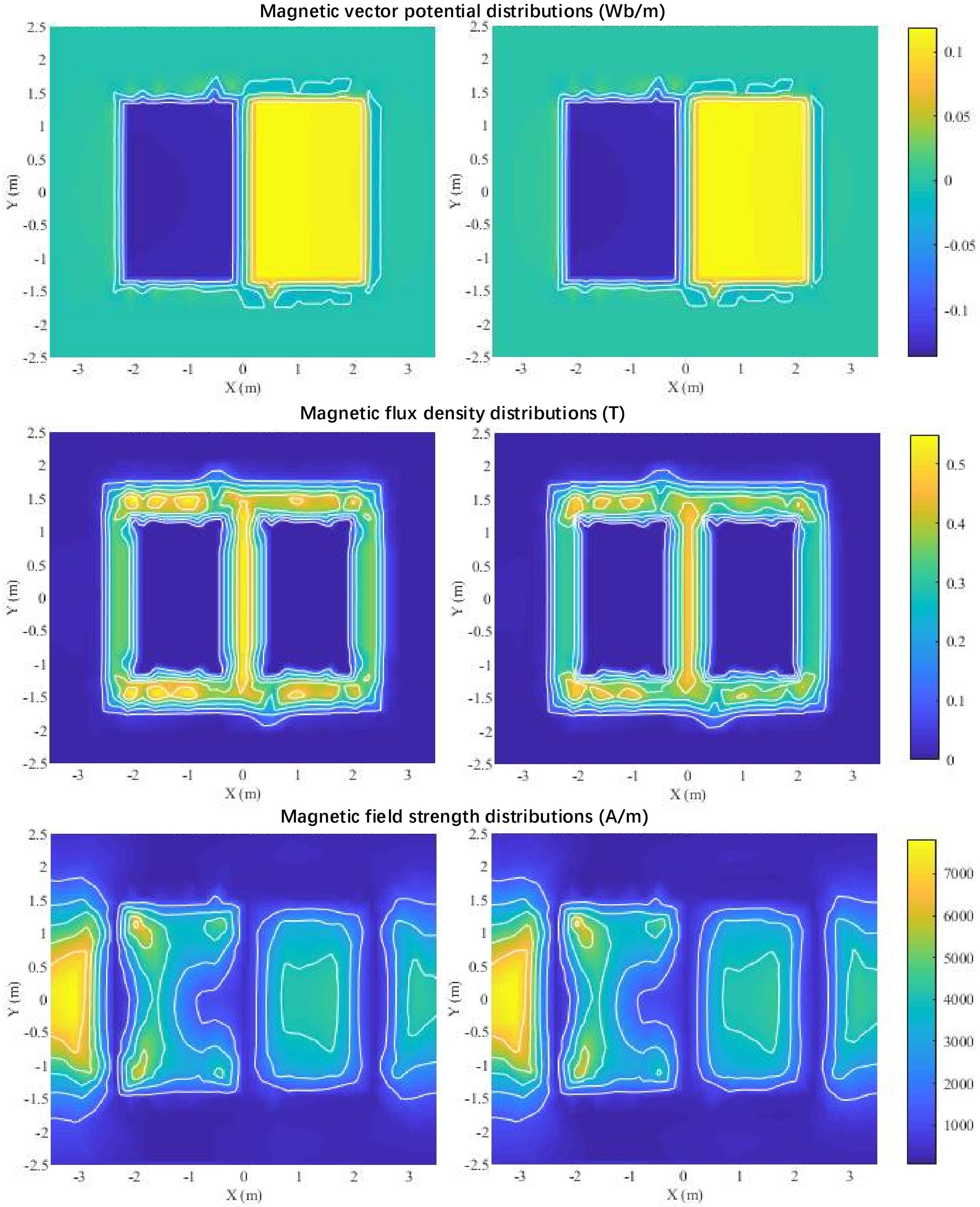


Figure 4.8: Magnetic vector potential distributions, magnetic flux density distributions, and magnetic field strength distributions at time $t = 37.82ms$ are shown respectively. All the distributions on the left are the results from Comsol® and all the distributions on the right are the real-time emulation results.

Table 4.1: Hardware resource utilization and timing report

FPGA Device	Module	Resource Utilization				Latency (clock cycles)
		BRAM	DSP	FF	LUT	
Xilinx Virtex UltraScale+ xcvu37p-fsvh2892-2L-e (Board2)	TLM Current Term	0	41	70965	119984	616
	Eddy Current Term	0	43	42686	102223	401
	Excitation Current Term	0	128	9685	9668	5
	Admittance Matrix Updater	0	4052	581950	507265	416
	TLM Newton Solver	76	4123	677572	407419	1185
	Subtotal	2%	94%	53%	88%	2623
Xilinx Virtex UltraScale+ xcvu37p-fsvh2892-2L-e (Board1)	RHS Generator	0	770	80469	98587	3
	PCG Solver	0	7312	1075172	768384	2820
	Impulse Generator	0	24	3613	11413	112
	Subtotal	0	90%	45%	68%	2935

this work.

4.4.2 Scenario II: Voltage-Excited Emulation

To interface the current-excited three-phase transformer FE model with the external circuit, a field-circuit coupling is required. The coupling approaches have been categorized into two groups and they are direct coupling methods and indirect coupling methods [78]. The direct coupling methods intuitively solve a large matrix combined by the drive circuit and the FE model of the device. However, the symmetry of the FE matrix may be broken, resulting in increased computation workload. On the contrary, the indirect coupling methods solve the circuit and the FE model individually. One indirect coupling method proposed in [79] which can guarantee accurate results even under a strong eddy current is adopted in this work.

To begin with, the relationship between the magnetic vector potential and the electromotive force (EMF) generated by coils can be established. As stated by Faraday's law, the EMF is proportional to the rate of change of the magnetic flux and the number of turns of coil. Besides, the magnetic flux is determined by the integral of magnetic flux density over the surface and the magnetic flux density is the curl of the magnetic vector potential. With the Kelvin-Stokes theorem, the integral of magnetic flux density over a surface can be rewritten as the line integral around the surface. As a result, an equation linking the discretized magnetic vector potential and the EMF in each FE is generated as follows:

$$V^e = \sum_{i=1}^{3} \frac{\partial A_i}{\partial t} \oint_{\Gamma^e} (N_w \cdot N_i \cdot \hat{n}) dl = \mathbf{A_w} \cdot \frac{\partial \mathbf{A}}{\partial t}, \tag{4.14}$$

where A_1^e, A_2^e, and A_3^e are the values at vertices of element Ω^e, N_w is the number of turns in a winding, and $\hat{n}$ is the unit vector of the wire direction. The row vector $\mathbf{A_w}$ can be considered as the weight vector for A at each node.

The EMF generated at each FE is connected in series, and the sum of these EMFs provides the total EMF created by the transformer. Gathering (4.5) and (4.14) together, the connection between the EMF over the transformer and the current through the windings can be discovered after replacing the vector of derivative of A with the product of the inverse of $\mathbf{A_w}$ and the EMF over the transformer. Note that the current also occurs on the right-hand side of (4.5).

Six equations for six circuits with six windings can be written according to Kirchhoff's voltage circuit law as follows:

$$\mathbf{V_{ex}} + \mathbf{V_{coil}} = 0, \tag{4.15}$$

where $\mathbf{V_{ex}}$ represents the external voltages and $\mathbf{V_{coil}}$ represents the voltages across the coils. Both of them are 6×1 vectors. Note that each entry in $\mathbf{V_{ex}}$ only depends on the current in its own circuit. For instance, entries in $\mathbf{V_{ex}}$ are denoted by $\mathbf{V_{ex}}[1-6]$, likewise, the currents through circuits are $\mathbf{I}[1-6]$. The derivative of $\mathbf{V_{ex}}[1]$ with respect to $\mathbf{I}[1]$ is nonzero while the derivative of $\mathbf{V_{ex}}[1]$ with respect to other currents are all zero. However, each entry in $\mathbf{V_{coil}}$ is the function of all six currents through the coils due to the energy exchange by the transformer between the electric field and the magnetic field. Based on the aforementioned relationship between the EMF and the current, the Jacobian matrix can be calculated by solving the same FEM equation as (4.5) with different excitation current vector on the right-hand side and making a weighted sum with (4.14) as explained in [79].

With (4.14) and the determined relationship between the EMF and the current, these six coupled equations can be solved by Newton-Raphson method after two to four iterations [79]. Even though the number of iterations for the field-circuit coupling is not large, the involved computation workload at each iteration is quite heavy. At the beginning of each iteration, the magnetic vector potentials need to be solved by adaptive TLM with PCG solver. Then, the weighted summations of EMF at each winding, the solution of matrix equations which has the size of the number of nodes and is used to generate the Jacobian matrix, and the process of generating the right-hand side of Jacobian matrix equations will all be implemented six times. Besides, twenty-one entries need to be resolved to compose the symmetric Jacobian matrix. At the end of each iteration, the updated 6×6 matrix will be computed to obtain the current increments. A data flow for this field-circuit coupling technique is presented in Fig. 4.9.

Each phase of the three-phase transformer was connected to an external circuit as

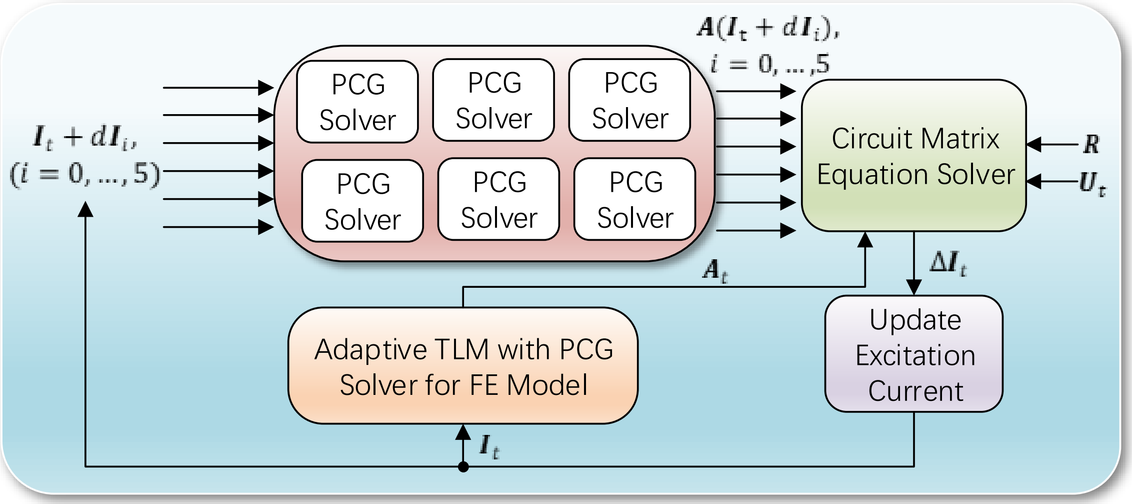


Figure 4.9: Dataflow for field-circuit coupling technique.

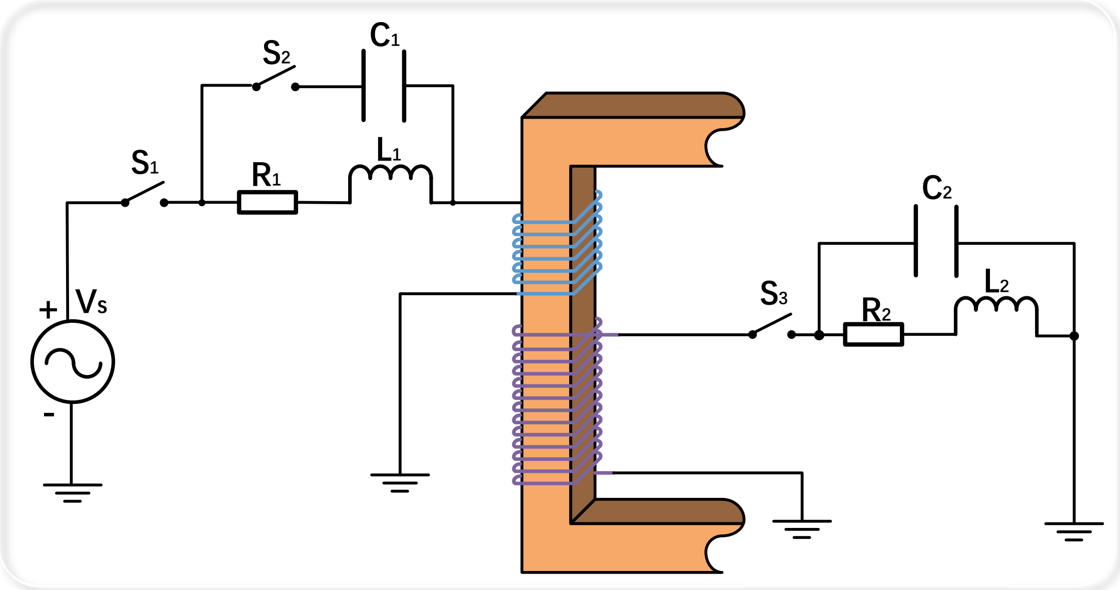


Figure 4.10: Schematic of the external circuit for one phase of the transformer.

shown in Fig. 4.10. The parameters of the external circuit are given in Appendix A. At the time $t = 0$, switch S_1 is turned on to energize the transformer meanwhile the secondary windings are open-circuited. After $100ms$, switch S_2 and S_3 are turned on and a load is added to each phase. After another $100ms$, the third and the fifth harmonics are injected into the voltage source V_s for each phase. The currents and voltages for the coil of the transformer phase A are shown in Fig. 4.11. The harmonic waveforms in the dashed box emulated by the FPGAs can also be seen displayed on the oscilloscope in Fig. 4.7. In addition, an analysis in the frequency domain was also compared in Fig. 4.12.

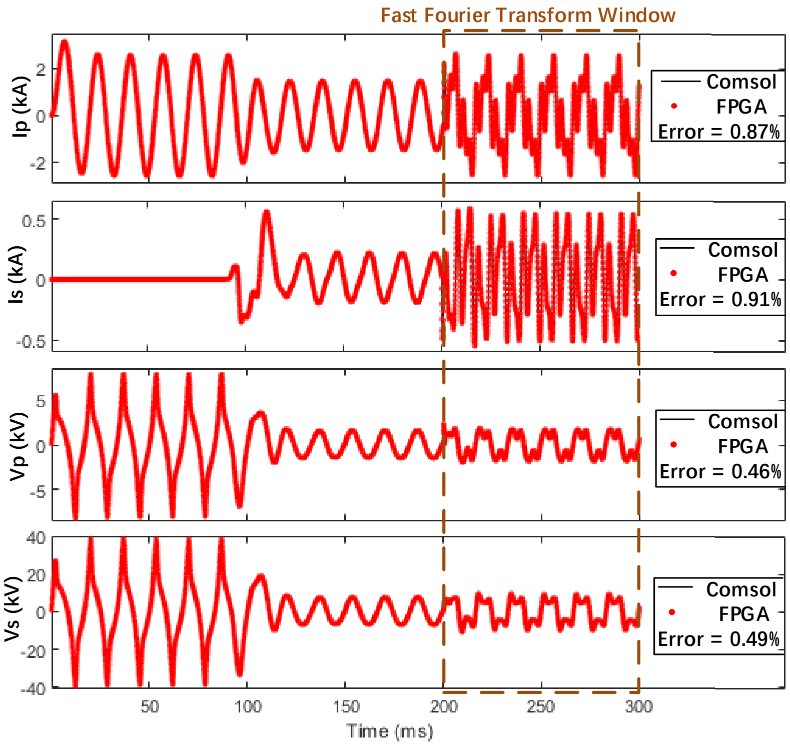


Figure 4.11: Comparison of results from emulation on FPGA and Comsol® off-line simulation.

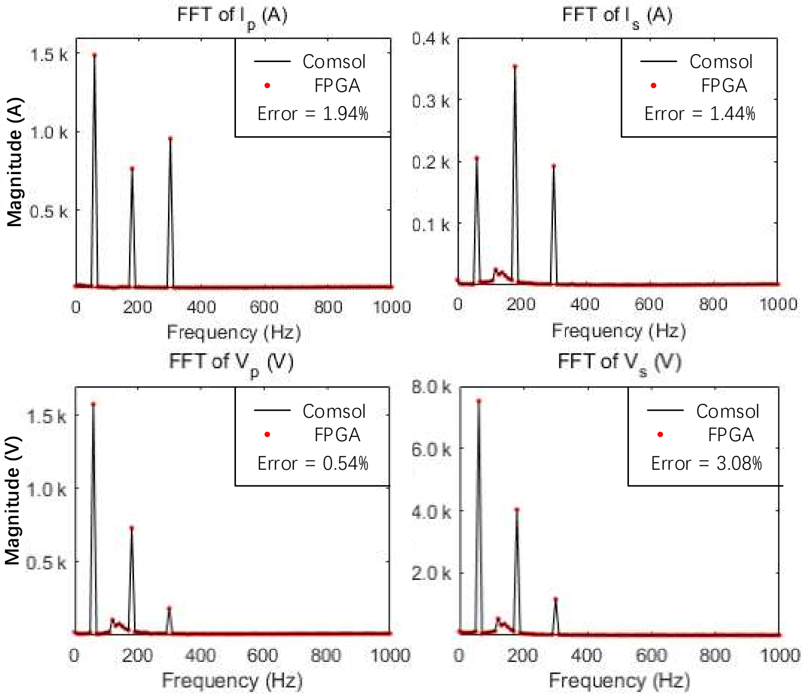


Figure 4.12: Emulation results on FPGA analyzed in frequency-domain and compared with Comsol® off-line simulation results.

Research on this field-circuit coupling technique on FPGA has been investigated and the approximate execution time for the process of applying this field-circuit coupling technique to solve a circuit and a FE system is 539.55μs. With such a heavy computation workload, it is obvious that this circuit and the FE system has not be simulated in real-time in this work, especially because of the limited resources on FPGAs. To be specific, the solution of matrix equations to prepare for the Jacobian matrix has to be calculated in series to accommodate the limited resources, which raises the execution time dramatically. However, the more available resource on FPGA will still be beneficial to the execution time; should enough resources become available in the future, it is even possible to achieve real-time emulation of this circuit and FE system.

4.5 Summary

In this chapter, a real-time emulation of the nonlinear eddy current problem with finite-elements is proposed for a three-phase transformer with guaranteed accuracy. The adaptive TLM method and PCG algorithm are integrated to construct the solver for the matrix equations that describes this problem. In this algorithm, the TLM method isolated the nonlinearity from the network successfully and the adaptive TLM method as a modified TLM is chosen to balance the resource utilization and the execution time. The PCG algorithm is implemented with deep data pipelining on FPGA to outperform other matrix equation solvers. The hardware design of this algorithm employs two FPGA boards because the limited resource on one single board is not sufficient to satisfy the demand. Furthermore, the combination between this adaptive TLM with PCG solver and a precise field-circuit coupling is discussed. With the growth of capability on FPGAs, a simulation of this combination will be accomplished efficiently in the future.

5 Conclusions and Future Works

The catastrophic impact and the prevalence of EMTs in a power system make EMT study a crucial step to design a reliable power system. Nevertheless, a detailed analysis such as FEM insinuates a longer design cycle because of the complexity of the model and the inefficient algorithm. This thesis has explored efficient algorithms to analyze EMTs in both ionized field problems, as an example of transmission tower configuration, and power transformer, as an example of power equipment. As the importance of high-performance computing, the computing power is improved by integrating more and more computing cores and utilizing them as many as possible concurrently. Following this hardware development trend, an algorithm which solves a large-scale FE model at the node-level in a massively parallel manner is proposed for ionized field problem around hybrid AC/DC lines. This algorithm can be intuitively accommodated for parallel architecture such as GPU. It is worth mentioning that no nonlinearity is involved in this ionized field problem because this model is filled by air only. However, nonlinearity is an inevitable difficulty for detailed EMT analysis about a power transformer. An algorithm that decouples the nonlinearity in a power transformer and can be calculated in a parallel fashion is implemented on FPGAs whose intrinsic parallelism and pipeline architecture enable more efficient computation. A real-time simulation is obtained as well. Besides, the field-circuit interface to link the finite element model with the external network for EMT studies is discussed.

5.1 Contributions of Thesis

The main contributions of this thesis are summarized as follows:

- An algorithm with node-level parallelism and without any computation burden for large global matrix assembly is proposed to compute a hybrid ionized field without the restriction of Deutsch's assumption. Each sub-domain solver is a module with the perfect size to showcase single instruction multiple data programming and hence deployed on many-core GPU to obtain massive parallelism and computation efficiency improvement. This is also a time-marching algorithm to obtain the dynamics of many quantities such as electric potential, positive space charge density, and negative space charge density. The dynamics for ion immigration are consistent with the corona phenomenon around HVDC transmission lines. The ion immigration under different wind conditions is also discussed and illustrated. The results of the comparison between this algorithm and commercial software Comsol™ demonstrate a speedup of 17 times retaining a high accuracy (mean relative error is about 0.07%).

- The transmission line decoupling technique is applied to deal with the nonlinear finite elements in a transformer modeled by FEs. This technique uses transmission lines to separate nonlinear components from a linear network and then the nonlinear components can be solved individually. By integrating the adaptive transmission-line modeling method, which is an example of the transmission line decoupling technique, with the preconditioned conjugate gradient method, a real-time solver is achieved on FPGAs. This solver is implemented on two FPGAs in a parallel and deeply pipelined fashion. High accuracy and computational efficiency are provided based on the comparison between these real-time solver results and the commercial software results.

- Integration between an indirect field-circuit coupling scheme and the real-time finite element model is proposed to a system of transformer and drive circuits. With this field-circuit coupling technique, the FE model and circuit are solved individually and thus the symmetry of the FEM matrix is not lost. Also, this technique can reuse the hardware design for preconditioned conjugate gradient method, which diminishes the stress caused by resource limitation on FPGA.

5.2 Future Research Topics

The proposed research topics for future work include but are not limited to the following:

- Homogenization approach was used to model windings in the transformer model. Some other models of a coil such as distributed modeling can be used to obtain more details in the future.

- The analysis domain in the ionized field problem was considered as air-filled space, which generates a linear FE model. In the future, the research can focus on other complicated configurations which may inset nonlinear materials into the analysis domain.

- A transformer model with 728 elements was simulated by two FPGAs in this work. To model a transformer with a finer-grained mesh, more FPGAs can be connected in a dedicated topology for more detailed simulation.

- In this work, a system to solve FE problems was built up by either GPU or FPGAs. A hybrid system consisting of different parallel computing platforms and hardware can be explored to acquire the advantages of different architecture.

Bibliography

[1] N. Watson and J. Arillate, *Power systems electromagnetic transients simulation*. The Institution of Engineering and Technology, London, U. K., 2003.

[2] H. W. Dommel, "Techniques for analyzing electromagnetic transients," *IEEE Comput. Appl. Power*, vol. 10, no. 3, pp. 18–21, Jul. 1997, ISSN: 1558-4151.

[3] H. Hill, A. Capon, O. Ratz, P. Renner, and W. Schmidt, *Transmission line reference book HVDC to +-600kV*. Jan. 1976.

[4] O. Stanojev, J. Garrison, S. Hedtke, C. Franck, and T. Demiray, "Benefit Analysis of a Hybrid HVAC/HVDC Transmission Line: A Swiss Case Study," Jun. 2019, pp. 1–6.

[5] V. L. Chartier, R. D. Stearns, and A. L. Burns, "Electrical environment of the uprated pacific NW/SW HVDC intertie," *IEEE Trans. Power Del.*, vol. 4, no. 2, pp. 1305–1317, 1989.

[6] B. A. Clairmont, G. B. Johnson, and L. E. Zaffanella, "The effect of hvac-hvdc line separation in a hybrid corridor," *IEEE Trans. Power Del.*, vol. 4, no. 2, pp. 1338–1350, 1989.

[7] J. S. Townsend, "The potentials required to maintain currents between coaxial cylinders," *The London, Edinburgh, and Dublin Philos. Mag. J. Sci.*, vol. 28, no. 163, pp. 83–90, 1914.

[8] M. P. Sarma and W. Janischewskyj, "Analysis of corona losses on DC transmission lines: I - unipolar lines," *IEEE Trans. Power App. Sys.*, vol. PAS-88, no. 5, pp. 718–731, 1969.

[9] ——, "Analysis of corona losses on DC transmission lines part II - bipolar lines," *IEEE Trans. Power App. Syst.*, vol. PAS-88, no. 10, pp. 1476–1491, 1969.

[10] W. Janischewskyj and G. Cela, "Finite element solution for electric fields of coronating DC transmission lines," *IEEE Trans. Power App. Syst.*, vol. PAS-98, no. 3, pp. 1000–1012, 1979.

[11] V. L. Chartier, S. H. Sarkinen, R. D. Stearns, and A. L. Burns, "Investigation of corona and field effects of AC/DC hybrid transmission lines," *IEEE Trans. Power App. Syst.*, vol. PAS-100, no. 1, pp. 72–80, 1981.

[12] I. C. A.F. Penner M.R. Raghuveer, "Finite element computation of corona injected current in AC conductors of hybrid AC/DC transmission," *Int. J. Energy Syst.*, vol. 12, no. 3, pp. 124–127, 1992.

[13] T. Zhao, S. A. Sebo, and D. G. Kasten, "Calculation of single phase AC and monopolar DC hybrid corona effects," *IEEE Trans. Power Del.*, vol. 11, no. 3, pp. 1454–1463, 1996.

[14] J. A. Martinez and B. A. Mork, “Transformer modeling for low- and mid-frequency transients - a review,” *IEEE Trans. Power Del.*, vol. 20, no. 2, pp. 1625–1632, 2005.

[15] V. Brandwajn, H. W. Dommel, and I. I. Dommel, “Matrix representation of three-phase n-winding transformers for steady-state and transient studies,” *IEEE Power Eng. Rev.*, vol. PER-2, no. 6, pp. 27–28, 1982.

[16] *Alternative Transients Program (ATP): Rule Book*. Heverlee, Belgium,Leuven EMTP Center, 1992.

[17] X. Chen, “Negative inductance and numerical instability of the saturable transformer component in emtp,” *IEEE Trans. Power Del.*, vol. 15, no. 4, pp. 1199–1204, 2000.

[18] T. Henriksen, “How to avoid unstable time domain responses caused by transformer models,” *IEEE Trans. Power Del.*, vol. 17, no. 2, pp. 516–522, 2002.

[19] G. R. Slemon, “Equivalent circuits for transformers and machines including non-linear effects,” *Proc. Inst. Elect. IV*, vol. 100, pp. 129–143, 1953.

[20] B. A. Mork, F. Gonzalez, D. Ishchenko, D. L. Stuehm, and J. Mitra, “Hybrid transformer model for transient simulation—part I: Development and parameters,” *IEEE Trans. Power Del.*, vol. 22, no. 1, pp. 248–255, 2007.

[21] P. Liu and V. Dinavahi, “Real-Time Finite-Element Simulation of Electromagnetic Transients of Transformer on FPGA,” *IEEE Trans. Power Del.*, vol. 33, no. 4, pp. 1991–2001, 2018.

[22] A. Polycarpou, *Introduction to the Finite Element Method in Electromagnetics*. 2006, pp. 1–126.

[23] *Supercomputer Fugaku specifications*. [Online]. Available: `https://www.fujitsu.com/global/about/innovation/fugaku/specifications/`.

[24] *NVIDIA Tesla V100 GPU accelerator*. [Online]. Available: `https://images.nvidia.com/content/technologies/volta/pdf/tesla-volta-v100-datasheet-letter-fnl-web.pdf`.

[25] *NVIDIA Tesla V100 GPU architecture*. [Online]. Available: `https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf`.

[26] A. Raj, *FPGA based Embedded System Developer's Guide*. 2019.

[27] J. Cong and Y. Ding, “An optimal technology mapping algorithm for delay optimization in lookup-table based FPGA designs,” in *ICCAD*, 1992.

[28] *The why and how of pipelining in FPGAs*. [Online]. Available: `https://www.allaboutcircuits.com/technical-articles/why-how-pipelining-in-fpga/`.

[29] J. Cong, B. Liu, S. Neuendorffer, J. Noguera, K. Vissers, and Z. Zhang, “High-Level Synthesis for FPGAs: From prototyping to deployment,” *IEEE Trans.Comput.-Aided Design of Integr. Circuits Syst.*, vol. 30, no. 4, pp. 473–491, 2011.

[30] T. A. Davis, *Direct Methods for Sparse Linear Systems*. 2006, pp. 1–217.

[31] Y. Saad, *Iterative Methods for Sparse Linear Systems*. 2000, pp. 1–447.

[32] E. Jajaga and J. Jajaga, “MPI parallel implementation of Jacobi,” 2012.

[33] G. Wu, X. Xie, Y. Dou, and M. Wang, "High-performance architecture for the conjugate gradient solver on FPGAs," *IEEE Trans. Circuits Syst., II, Exp. Briefs*, vol. 60, no. 11, pp. 791–795, 2013.

[34] M. M. Dehnavi, D. M. Fernández, and D. Giannacopoulos, "Enhancing the performance of conjugate gradient solvers on Graphic Processing Units," *IEEE Trans. Magn.*, vol. 47, no. 5, pp. 1162–1165, 2011.

[35] T. Okimura, T. Sasayama, N. Takahashi, and S. Ikuno, "Parallelization of finite element analysis of nonlinear magnetic fields using GPU," *IEEE Trans. Magn.*, vol. 49, no. 5, pp. 1557–1560, 2013.

[36] Y. Takahashi, K. Fujiwara, T. Iwashita, and H. Nakashima, "Parallel finite-element analysis of rotating machines based on domain decomposition considering nonconforming mesh connection," *IEEE Trans. Magn.*, vol. 52, no. 3, pp. 1–4, 2016.

[37] W. Wang, R. Xu, H. Li, Y. Liu, X. Guo, Y. Xu, H. Li, H. Zhou, and W. Yin, "Massively parallel simulation of large-scale electromagnetic problems using one high-performance computing scheme and domain decomposition method," *IEEE Trans. Electromagn. Compat.*, vol. 59, no. 5, pp. 1523–1531, 2017.

[38] *Vivado design suite user guide High-Level Synthesis*.

[39] Q. Li, Y. Xiang, Q. Mu, X. Zhang, X. Li, and G. He, "Exploration of FPGA-based electromagnetic transient real-time simulation system design using High-Level Synthesis," *J. Eng.*, vol. 2019, no. 16, pp. 1217–1220, 2019.

[40] M. P. Bahrman and B. K. Johnson, "The ABCs of HVDC transmission technologies," *IEEE Power Energy Mag.*, vol. 5, no. 2, pp. 32–44, 2007.

[41] P. S. Maruvada and S. Drogi, "Field and ion interactions of hybrid AC/DC transmission lines," *IEEE Trans. Power Del.*, vol. 3, no. 3, pp. 1165–1172, 1988.

[42] A. A. Elmoursi and G. S. P. Castle, "Modeling of corona characteristics in a wire-duct precipitator using the charge simulation technique," *IEEE Trans. Ind. Appl.*, vol. IA-23, no. 1, pp. 95–102, 1987.

[43] T. Takuma, T. Ikeda, and T. Kawamoto, "Calculation of ion flow fields of HVDC transmission lines by the finite element method," *IEEE Trans. Power App. Syst.*, vol. PAS-100, no. 12, pp. 4802–4810, 1981.

[44] T. Lu, H. Feng, Z. Zhao, and X. Cui, "Analysis of the electric field and ion current density under Ultra High-Voltage Direct-Current transmission lines based on finite element method," *IEEE Trans. Magn.*, vol. 43, no. 4, pp. 1221–1224, 2007.

[45] Y. Zhen, X. Cui, T. Lu, X. Zhou, and Z. Luo, "High efficiency FEM calculation of the ionized field under HVDC transmission lines," *IEEE Trans. Magn.*, vol. 48, no. 2, pp. 743–746, 2012.

[46] H. Yin, J. He, B. Zhang, and R. Zeng, "Finite volume-based approach for the hybrid ion-flow field of UHVAC and UHVDC transmission lines in parallel," *IEEE Trans. Power Del.*, vol. 26, no. 4, pp. 2809–2820, 2011.

[47] Y. Takahashi, K. Fujiwara, T. Iwashita, and H. Nakashima, "Parallel finite-element analysis of rotating machines based on domain decomposition considering nonconforming mesh connection," *IEEE Trans. Magn.*, vol. 52, no. 3, pp. 1–4, 2016.

[48] W. Wang, R. Xu, H. Li, Y. Liu, X. Guo, Y. Xu, H. Li, H. Zhou, and W. Yin, "Massively parallel simulation of large-scale electromagnetic problems using one high-performance computing scheme and domain decomposition method," *IEEE Trans. Electromagn. Compat.*, vol. 59, no. 5, pp. 1523–1531, 2017.

[49] P. Liu and V. Dinavahi, "Matrix-free nodal domain decomposition with relaxation for massively parallel finite-element computation of EM apparatus," *IEEE Trans. Magn.*, vol. 54, no. 9, pp. 1–7, 2018.

[50] A. N. Brooks and T. J. Hughes, "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations," *Comput. Methods in Appl. Mech. Eng.*, vol. 32, no. 1, pp. 199 –259, 1982, ISSN: 0045-7825.

[51] W. Li, B. Zhang, J. He, R. Zeng, X. Li, and Q. Wang, "Calculation of the ion flow field of AC-DC hybrid transmission lines," *IET Gener.*, vol. 3, no. 10, pp. 911–918, 2009.

[52] A. C. Polycarpou, "Two-dimensional boundary-value problems," in *Introduction to the Finite Element Method in Electromagnetics*, 1. Morgan & Claypool, 2006, pp. 61–124.

[53] Occupational Safety and Health Administration, *Power line safety (up to 350 kv)– equipment operations*. [Online]. Available: `https://www.osha.gov/laws-regs/regulations/standardnumber/1926/1926.1408`.

[54] Z. M. Al-Hamouz, "Adaptive finite-element ballooning analysis of bipolar ionized fields," *IEEE Trans. Ind. Appl.*, vol. 32, no. 6, pp. 1266–1277, 1996.

[55] *CUDA Toolkit 9.1*. [Online]. Available: `https://developer.nvidia.com/cuda-91-download-archive`.

[56] Y. Tian, X. Huang, W. Tian, Y. Zhu, L. Zhao, and Y. Zhang, "Study on the hybrid ion-flow field of HVDC and HVAC transmission lines by the nodal discontinuous Galerkin time-domain method," *IET Gener.*, vol. 11, no. 1, pp. 209–217, 2017.

[57] Z. M. Al-Hamouz, "Corona power loss, electric field, and current density profiles in bundled horizontal and vertical bipolar conductors," *IEEE Trans. Ind. Appl.*, vol. 38, no. 5, pp. 1182–1189, 2002.

[58] PARASOL, *MUMPS:a parallel sparse direct solver*. [Online]. Available: `http://mumps.enseeiht.fr/`.

[59] A. Gungor, J. Smajic, F. Moro, and J. Leuthold, "Time-domain coupled full Maxwell- and drift-diffusion-solver for simulating scanning microwave microscopy of semiconductors," in *2019 PhotonIcs Electromagnetics Research Symposium - Spring (PIERS-Spring)*, 2019, pp. 4071–4077.

[60] J. A. Martinez, R. Walling, B. A. Mork, J. Martin-Arnedo, and D. Durbak, "Parameter determination for modeling system transients-part III: Transformers," *Trans. Power Del.*, vol. 20, no. 3, pp. 2051–2062, 2005.

[61] A. Narang and R. H. Brierley, "Topology based magnetic model for steady-state and transient studies for three-phase core type transformers," *IEEE Trans. Power Syst.*, vol. 9, no. 3, pp. 1337–1349, 1994.

[62] E. Monmasson and M. N. Cirstea, "FPGA design methodology for industrial control systems—a review," *IEEE Trans. Ind. Electron.*, vol. 54, no. 4, pp. 1824–1842, 2007.

[63] J. Liu and V. Dinavahi, "Nonlinear magnetic equivalent circuit-based real-time sen transformer electromagnetic transient model on FPGA for HIL emulation," *IEEE Trans. Power Del.*, vol. 31, no. 6, pp. 2483–2493, 2016.

[64] Y. Chen and V. Dinavahi, "FPGA-based real-time EMTP," *IEEE Trans. Power Del.*, vol. 24, no. 2, pp. 892–902, 2009.

[65] S. M. Trimberger, "Three ages of FPGAs: A retrospective on the first thirty years of FPGA technology," *Proc. IEEE Proc. IRE*, vol. 103, no. 3, pp. 318–331, 2015.

[66] A. Myaing and V. Dinavahi, "FPGA-based real-time emulation of power electronic systems with detailed representation of device characteristics," *IEEE Trans. Ind. Electron.*, vol. 58, no. 1, pp. 358–368, 2011.

[67] K. Sridharan and T. K. Priya, "The design of a hardware accelerator for real-time complete visibility graph construction and efficient FPGA implementation," *IEEE Trans. Ind. Electron.*, vol. 52, no. 4, pp. 1185–1187, 2005.

[68] Y. Chen and V. Dinavahi, "Digital hardware emulation of universal machine and universal line models for real-time electromagnetic transient simulation," *IEEE Trans. Ind. Electron.*, vol. 59, no. 2, pp. 1300–1309, 2012.

[69] H. W. Dommel, *EMTP Theory Book*. Porland, OR, USA: BPA, 1985.

[70] V. Brandwajn, H. W. Dommel, and I. I. Dommel, "Matrix representation of three-phase n-winding transformers for steady-state and transient studies," *IEEE Trans. Power App. Syst.*, vol. PAS-101, no. 6, pp. 1369–1378, 1982.

[71] O. Deblecker, J. Lobry, and C. Broche, "Novel algorithm based on transmission-line modeling in the finite-element method for nonlinear quasi-static field analysis," *IEEE Trans. Magn.*, vol. 39, no. 1, pp. 529–538, 2003.

[72] P. Liu and V. Dinavahi, "Matrix-free nodal domain decomposition with relaxation for massively parallel finite-element computation of EM apparatus," *IEEE Trans. Magn.*, vol. 54, no. 9, pp. 1–7, 2018.

[73] O. Deblecker, J. Lobry, and C. Broche, "Use of transmission-line modelling method in FEM for solution of nonlinear eddy-current problems," *IEE Proceedings - Science, Measurement and Technology*, vol. 145, no. 1, pp. 31–38, 1998.

[74] C. Christopoulos, *The Transmission-Line Modeling (TLM) Method in Electromagnetics*. 2006, pp. 1–123.

[75] P. B. Johns and M. O'Brien, "Use of the transmission-line modelling (t.l.m.) method to solve non-linear lumped networks," *Radio and Electronic Engineer*, vol. 50, pp. 59–70, 1980.

[76] P. Liu, J. Li, and V. Dinavahi, "Matrix-free nonlinear finite-element solver using transmission-line modeling on GPU," *IEEE Trans. Magn.*, vol. 55, no. 7, pp. 1–5, 2019.

[77] R. Larson, B. H. Edwards, and D. C. Falvo, *Elementary Linear Algebra*, Fifth Edition. Houghton Mifflin College Div, 2003.

[78] B. Asghari, V. Dinavahi, M. Rioual, J. A. Martinez, and R. Iravani, "Interfacing techniques for electromagnetic field and circuit simulation programs IEEE task force on interfacing techniques for simulation tools," *IEEE Trans. Power Del.*, vol. 24, no. 2, pp. 939–950, 2009.

[79] J. Li, P. Liu, and V. Dinavahi, “Matrix-free edge-domain decomposition method for massively parallel 3-d finite element simulation with field-circuit coupling,” *IEEE Trans. Magn.*, vol. 56, no. 10, pp. 1–9, 2020.

[80] X. Zhou, X. Cui, T. Lu, Y. Zhen, and Z. Luo, “A time-efficient method for the simulation of ion flow field of the AC-DC hybrid transmission lines,” *IEEE Trans. Magn.*, vol. 48, no. 2, pp. 731–734, 2012.

[81] J. Qiao, J. Zou, J. Yuan, J. B. Lee, and M. Ju, “Calculation of ion-flow field of hvdc transmission lines in the presence of wind using finite element-finite difference combined method with domain decomposition,” *IEEE Trans. Magn.*, vol. 52, no. 3, pp. 1–4, 2016.

[82] J. Lobry, J. Trecat, and C. Broche, “The transmission line modeling (TLM) method as a new iterative technique in nonlinear 2-d magnetostatics,” *IEEE Trans. Magn.*, vol. 32, no. 2, pp. 559–566, 1996.

A

A.1 Simulation Parameters in Chapter 4

- Transformer parameters: The limb length is 2.6 m, yoke length is 5.2 m, and the coil size is 0.25 m * 2 m, power rating is 85 MVA, 40kV/200kV. The number of coil turns is 40 for the primary side and 200 for the secondary side. The σ^e is 1000.

- Case study parameters: $V_a = 40\sqrt{2}sin(120\pi t)kV$, $V_b = 40\sqrt{2}sin(120\pi t - 120°)kV$, $V_c = 40\sqrt{2}sin(120\pi t + 120°)kV$, $R_1 = R_2 = 10\Omega$, $L_1 = L_2 = 46mH$, and $C_1 = C_2 = 93\mu F$. The magnitude of the injected third and fifth harmonics are 6.67kV and 4kV respectively.