Design of Machine Learning based Framework for Semiconductor Device Analysis

A

Thesis submitted

for the award of the degree of

DOCTOR OF PHILOSOPHY

By

GAURAV KUMAR

DEPARTMENT OF ELECTRONICS AND ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781 039, ASSAM, INDIA

MAY 2019

TH-2607_126102004

TH-2607_126102004

Certificate

This is to certify that the thesis entitled "**Design of Machine Learning based Framework for Semiconductor Device Analysis**", submitted by **GAURAV KUMAR** (126102004), a research scholar in the *Department of Electronics and Electrical Engineering, Indian Institute of Technology Guwahati*, for the award of the degree of **Doctor of Philosophy**, is a record of an original research work carried out by him under our supervision and guidance. The thesis has fulfilled all requirements as per the regulations of the institute and in our opinion has reached the standard needed for submission. The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Dated:
Guwahati.

Dr. Gaurav Trivedi
Associate Professor
Dept. of Electronics and Electrical Engg.
Indian Institute of Technology Guwahati
Guwahati - 781 039, Assam, India.

Dated:
Guwahati.

Prof. Anil Mahanta
Professor
Dept. of Electronics and Electrical Engg.
Indian Institute of Technology Guwahati
Guwahati - 781 039, Assam, India.

TH-2607_126102004

TH-2607_126102004

To My Family

TH-2607_126102004

TH-2607_126102004

Acknowledgements

Foremost, I would like to express my sincere gratitude to my thesis advisors **Dr. Gaurav Trivedi** and **Prof. Anil Mahanta** for their continuous support, motivation, patience, and immense knowledge during my Ph.D. study and related research. Their guidance has helped me in all the time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my Ph.D. study. Besides my advisors, I would like to thank the rest of my doctoral committee: Prof. Roy Paily, Prof. H. Nemade, Prof. Pratima Agarwal, and Prof. Anup Gogoi, for their insightful comments and encouragements which incented me to widen my research from various perspectives.

A very special gratitude goes out to the Govt. of India for helping and providing funding for the work.

I would like to acknowledge my colleagues and beloved friends Deepak, Satyabrata, Karam, Sunil, Ashok, Sushanta, Sukanta, Mohit and Amit Singh for their significant help during the research and creating the best workplace to do the research. You guys have been an endless source of encouragement and inspiration, and it would be difficult to find such a wonderful group elsewhere.

I would like to thank Shri Surinder Singh, Shri H. S. Jatana and Shri Ashutosh Yadav from Semiconductor Laboratory, Chandigarh, India for providing technical support and facilities.

I can't thank enough to my parents, sister, and wife for their altruistic love, support, sacrifice and encouragements throughout the thesis. I thank God almighty for this wonderful experience.

And finally, last but by no means least, also to everyone in IIT Guwahati community, it was great sharing a beautiful campus with all of you during the timeline of my thesis.

Thanks for all your encouragement!

Gaurav Kumar

TH-2607_126102004

TH-2607_126102004

Abstract

The quest for making sub-nano regime devices is posing several challenges to the electronics industry, such as high leakage current, high DIBL, hot electron effect and other short channel effects. Technology Computer Aided Design (TCAD) simulations can help overcome all these issues in the best possible manner, provided it gives reliable and accurate results. The solution accuracy of the TCAD simulations depends mainly on two main factors; 1) Transport model employed for the analysis, and 2) Numerical solution of the transport model which basically depends on the type of discretization scheme used for the approximation. In this thesis, the two main challenges of the TCAD; *Accuracy* and *Computation time*, have been addressed by incorporating different discretization methodologies, such as Finite Element Method (FEM), Discontinuous Galerkin Finite Element Method (DG-FEM), Streamline Upwind Petrov Galerkin (SUPG) based Finite Element Method (SUPG-FEM) and Element Free Galerkin (EFG) to improve the solution accuracy, and Random Walk and Machine Learning based methodologies to reduce the computation time of the analysis. This thesis presents the design and development of a parallel and scalable framework, **VEDA** (**V**ery **E**fficient **D**evice **A**nalyzer), for the analysis of semiconductor devices. The proposed methodologies are employed to discretize fundamental device equations, a set of Partial Differential Equations (PDEs), governing the flow of carriers inside the device, and later solved numerically using a suitable methodology to compute the solution.

DG-FEM and SUPG-FEM are two variants of FEM based on two different approaches to improve the solution accuracy. In DG-FEM, the basis functions employed for the discretization are discontinuous piecewise polynomials which aid to conserve the flux efficiently and are capable of handling complex geometries. In VEDA, SUPG method is

TH-2607_126102004

employed to calculate flux and flow of charge in the device. It is observed that the SUPG stabilization technique is better than the classical Scharfetter-Gummel method to solve a convection-diffusion equation. On the other hand, EFG is an element free or meshfree method to discretize the problem domain. In EFG a set of nodes are scattered throughout the domain to formulate linear algebraic equations which represent carrier transport in semiconductor devices without employing mesh for domain discretization. The solution provided by our proposed method is compared with other discretization techniques including the methodology adopted by commercial TCAD simulator, *Sentaurus*. The accuracy of the solution obtained by the proposed method is approximately $10\times$ better as compared to the solution provided by *Sentauras*.

Since the solution provided by EFG is accurate but not efficient in terms of computational time, we employ random walk method to speed up the solution of device equations. As we know that the numerical solution of differential equations depends on the initial guess, therefore, we propose to improve it by using random walk method. In this method, equivalent electrical circuits of fundamental device equations are formulated and are analyzed self-consistently in a coupled manner to perform device analysis. The random walk method is based on the probability distribution calculated from the weights at each node. The parallelizable aspect of random walk method has also been explored to improve the computation time of the analysis. The proposed method exhibits up to 20% faster solution with the maximum error of 4% in the initial guess as compared to the final solution. Although the improvement in the computational time obtained using random walk method is not significant, it motivates us to explore other methods such as machine learning algorithms and its application to the semiconductor device analysis. Therefore, different machine learning algorithms along with Artificial Neural Network (ANN) have been employed for the analysis of semiconductor devices for a given set of parameters. It is observed that the use of ANN has helped not only in improvising computation time but also providing nearly accurate solution. We have observed a minimum speedup of $1.86\times$ (13 iterations to 7 iterations) and maximum speedup of $3.5\times$ (21 iterations to 6 iterations)

TH-2607_126102004

for a different set of parameters by using the proposed methodology. The implementation of ANN has enabled us to explore a new paradigm in the domain of TCAD. The ANNs mimicking the electrical characteristics of semiconductor devices are also employed to perform circuit analysis. We have designed different circuits using these ANN based semiconductor devices, and their performances are found to be within the acceptable accuracy limits. The methodology employed for the analysis of circuits having ANN based semiconductor devices can easily be parallelized enabling quick turnout of the solution. This approach helps us to overcome issues related to the semiconductor device modeling without developing a semiconductor device model in the early stage of the device design. It eliminates the need for semiconductor device modeling and the efforts taken in the development of compact device models of semiconductor devices including novel devices (forthcoming devices whose device models are not yet known). The machine learning based semiconductor device and circuit analysis is found to be efficient in terms of accuracy and computation time as compared to the lookup table based methods employed for TCAD and circuit analysis. This begins a new era in the area of VLSI for semiconductor device design, modeling and circuit analysis.

TH-2607_126102004

TH-2607_126102004

Contents

1 Introduction **1**

1.1 Introduction . . . 2

1.2 Thesis Motivation and Objective . . . 7

1.3 Thesis Contributions . . . 8

1.4 Thesis Organization . . . 9

2 Fundamentals of TCAD in Semiconductor Device Analysis **13**

2.1 Introduction . . . 14

2.2 Physical Models for Device Analysis . . . 17

2.2.1 Drift-Diffusion Model . . . 19

2.2.1.1 Convergence Constraints on Drift-diffusion model . . . 20

2.2.2 Mesh Formation . . . 21

2.2.3 Boundary Conditions . . . 22

2.2.3.1 Dirichlet Boundary Condition . . . 22

2.2.3.2 Neumann Boundary Condition . . . 23

2.2.3.3 Robin Boundary Conditions . . . 23

2.3 A Survey of Different Discretization Methods to Implement Transport Models . . . 24

2.4 Formulation of Basic Device Equations . . . 26

2.5 Numerical Solution of Linear Algebraic Equations . . . 29

2.6 Summary . . . 30

3 Implementation of FEM and its Variants for Device Analysis **31**

3.1 TCAD Framework using Finite-Element Discretization . . . 32

3.1.1 Step-by-Step FEM Formulation of a Second Order 1D PDE . . . 35

TH-2607_126102004

3.1.2 Computational Analysis of FEM 39
3.1.3 Implementation Details and Simulation Results 40
3.1.3.1 PN Junction Diode 42
3.1.3.2 MOS Capacitor 43
3.1.3.3 MOSFET 44
3.2 TCAD Framework VEDA using DG-FEM 46
3.2.1 Implementation Details and Simulation Results 49
3.2.1.1 PN Junction Diode 49
3.3 TCAD Framework VEDA using SUPG Stabilization Technique 51
3.3.1 Formulation of Drift-Diffusion equation using SUPG method 51
3.3.2 Simulation Results 52
3.3.2.1 PN Junction Diode 52
3.4 Quantum Simulations with Schrödinger-Poisson Solver 54
3.4.1 GaAs/AlGaAs Heterostructure 55
3.4.2 MOS Capacitor 56
3.5 Performance Analysis of Different Discretization Schemes 58
3.6 Summary 58

4 An Accurate and Adaptive Framework based on EFG Method 61

4.1 Proposed Implementation of EFG Method for Device Analysis 62
4.2 Element-Free Galerkin Method 64
4.2.1 MLS Approximation 68
4.2.2 The Weight Function 70
4.3 Application of EFG method for the Analysis of Semiconductor Devices 71
4.3.1 Discretization of Poisson Equation 74
4.3.2 Discretization of Continuity Equation 80
4.4 Parametric Analysis of EFG 81
4.5 Implementation Details and Results 85
4.5.1 L_2-Norm Error Analysis 85

TH-2607_126102004

4.5.2 Computational Complexity 87
4.6 Summary 88

5 Random Walk and Machine Learning Algorithms based Analysis of Semiconductor Devices 91

5.1 Proposed TCAD framework using Random Walk Algorithm 93
5.1.1 Discretization of PDEs and their Equivalent Electrical Circuit 95
5.2 Implementation Details and Simulation Results 99
5.2.1 PN Junction Diode 101
5.3 Proposed TCAD Framework using Machine Learning 103
5.3.1 Logistic Regression 103
5.3.2 Decision Tree 105
5.3.3 Random Forest 106
5.3.4 Support Vector Machine (SVM) 106
5.3.5 Naive Bayes 107
5.3.6 Deep Learning with Artificial Neural Network 108
5.4 Circuit Analysis Using Machine Learning 111
5.5 Summary 118

6 Conclusion and Future Aspects 121

TH-2607_126102004

TH-2607_126102004

List of Figures

1.1 The process flow of a typical TCAD framework for device analysis. 4

2.1 Role of TCAD in manufacturing process. 15

2.2 The manufacturing process of a typical semiconductor device. 16

2.3 Shows the hierarchy of various transport models [1] 18

2.4 Non-Uniform mesh formation in case of MOSFET [2] 21

2.5 Steps to solve fundamental device equation on each individual node. 22

2.6 Different boundary conditions for the analysis of semiconductor devices. 23

2.7 Venn diagram showing relationship between different methods. 25

3.1 Comparison of different discretization schemes employed for (a) FDM and (b) FEM. 33

3.2 A tetrahedron shape function to be used as an element in FEM 33

3.3 A simple 1D geometry to understand the implementation of FEM. 35

3.4 (a) Uniform mesh over PN junction diode (b) Non-uniform mesh over PN junction diode 42

3.5 Potential profile of a 2D PN junction diode at equilibrium 43

3.6 Electric filed inside a PN diode at equilibrium 43

3.7 I-V characteristic of a forward bias PN junction diode 43

3.8 Non-uniform meshing for a 3D PN junction diode 44

3.9 Potential profile of a 3D PN junction diode at equilibrium 44

3.10 Potential profile inside a 3D MOS capacitor 45

3.11 Non-uniform mesh formation inside a 3D MOSFET 45

3.12 Doping profile of a 3D MOSFET 46

TH-2607_126102004

3.13 Comparison of output characteristics for a n-Channel MOSFET. 46
3.14 DG basis function with one degree of freedom per element. 47
3.15 DG basis function with two degree of freedom per element. 47
3.16 Basis function as used in Finite Element Method. 47
3.17 Basis function as used in DG-Finite Element Method. 47
3.18 Uniform mesh over PN junction diode. 50
3.19 Non-uniform mesh over PN junction diode. 50
3.20 Potential profile of a 2D PN junction diode at equilibrium. 50
3.21 Forward biased current-voltage characteristics of a PN junction diode. 50
3.22 Error analysis of finite element method with different degree of Lagrange elements. . 53
3.23 Accuracy COnvergence Product. 53
3.24 Structure of a GaAs/AlGaAs used in simulation. 55
3.25 Meshing used in the simulation. 55
3.26 Electron density profile inside GaAs/AlGaAs heterostructure. 56
3.27 Potential profile. 56
3.28 Wavefunction $k = 1$ for first eigen value. 56
3.29 Wavefunction $k = 2$ for second eigen value. 56
3.30 MOS-Capacitor used for simulation 57
3.31 CV characteristics of a MOS with $3nm$ oxide thickness. 57
3.32 CV characteristics of a MOS with $8nm$ oxide thickness. 57
3.33 Inversion Charge for a MOS with $3nm$ oxide thickness. 57
3.34 Inversion Charge for a MOS with $8nm$ oxide thickness. 57

4.1 Different discretization schemes employed for (a) FEM and (b) Element Free Method 64
4.2 Flowchart of EFG and FEM implementation 64
4.3 Local support domains used in the MFree method to construct shape functions. . . . 65
4.4 Comparison of different solvers with analytical solution of Poisson's equation. 86
4.5 Comparison of carrier concentration computed using EFG and Sentaurus. 86

TH-2607_126102004

4.6 Comparison of normalized error for different sizes of support domain. 87
4.7 Comparison of normalized error for different Gaussian points of integration. 87
4.8 Comparison of normalized error for different number of nodes with MLS method. . . 88
4.9 Comparison of normalized error for different number of nodes with RPIM method. . 88
4.10 Comparison of computation time for different number of nodes. 89

5.1 Random walk by a drunkard in a city. 94
5.2 Process flow of proposed TCAD framework VEDA with random walk method 95
5.3 Equivalent electrical circuit of 2D Poisson's equation for a single node 96
5.4 Equivalent electrical circuit for a Dirichlet boundary condition 96
5.5 Equivalent electrical circuit for a Neumann boundary condition 97
5.6 Equivalent electrical circuit of the Poisson's equation over a rectangular geometry . . 98
5.7 Equivalent electrical circuit of a current equation for a single node 98
5.8 Equivalent electrical circuit of the Current equation over a rectangular geometry . . . 98
5.9 Computation time of various parallelization schemes employed in Random Walk . . 101
5.10 Comparison of computation time for CUDA and CUDA with MPI as problem size increases 101
5.11 Plot showing percentage of error with number of walks at different number of nodes . 102
5.12 Comparison of initial guess (using Random Walk) of Poisson's equation with its final solution 103
5.13 Process flow of the proposed framework VEDA using machine learning algorithm . . 104
5.14 Basic principle of decision tree algorithm. 105
5.15 A 2-dimensional data set in support vector machine 106
5.16 Basic principle of an artificial neural network 108
5.17 A deep neural network with multiple hidden layers 109
5.18 Two different activation functions used in model formation: Sigmoid and Rectifier . . 109
5.19 Comparison of potential profile at the depletion region for different doping concentrations 110

TH-2607_126102004

5.20 Comparison of potential profile at the depletion region for different doping concentrations . . . 111
5.21 Comparison of Id-Vds of a MOSFET for different Vg. . . . 113
5.22 Output Characteristics of a MOSFET using ANN. . . . 113
5.23 A simple MOSFET circuit. . . . 114
5.24 Comparison of different characteristics of a simple MOSFET circuit (a) Output Characteristics (b) Transfer Characteristics . . . 115
5.25 A simple cascaded MOSFET circuit . . . 116
5.26 Comparison of transfer characteristics of a simple Cascaded MOS circuit . . . 116
5.27 Comparison of transfer characteristics of a CMOS Inverter. . . . 117
5.28 A current mirror circuit. . . . 117
5.29 Comparison of output characteristics of a current mirror circuit. . . . 118

TH-2607_126102004

List of Tables

2.1 Characteristics analysis of different discretization scheme. 26

2.2 Various device parameters used for a PN diode. 29

3.1 List of normalization factors for the quantities of interest. 41

3.2 Device parameters and their values for a PN junction diode. 42

3.3 Device parameters and their values for a 3D MOS capacitor. 44

3.4 Various device parameters used for a PN diode. 50

3.5 Various device parameters used for a MOS Capacitor. 56

3.6 Comparison of computation time for different discretization schemes. 58

4.1 Method: EFG-MLS, Parameters: $k = 4$, $mm = 3$, $d_{max} = 2.5$ 82

4.2 Method: EFG-MLS, Parameters: $k = 4$, $mm = 3$, $d_{max} = 3.5$ 82

4.3 Method: EFG-MLS, Parameters: $k = 4$, $mm = 3$, $d_{max} = 4.5$ 82

4.4 Method: MWS-MLS, Parameters: $k = 4$, $mm = 3$, $d_{max} = 2.5$ 82

4.5 Method: MWS-MLS, Parameters: $k = 4$, $mm = 3$, $d_{max} = 3.5$ 83

4.6 Method: MWS-MLS, Parameters: $k = 4$, $mm = 3$, $d_{max} = 4.5$ 83

4.7 Method: EFG-RPIM, Parameters: $k = 4$, $mm = 3$, $d_{max} = 2.5$ 83

4.8 Method: EFG-RPIM, Parameters: $k = 4$, $mm = 3$, $d_{max} = 3.5$ 83

4.9 Method: EFG-RPIM, Parameters: $k = 4$, $mm = 3$, $d_{max} = 4.5$ 84

4.10 Method: MWS-RPIM, Parameters: $k = 4$, $mm = 3$, $d_{max} = 2.5$ 84

4.11 Method: MWS-RPIM, Parameters: $k = 4$, $mm = 3$, $d_{max} = 3.5$ 84

4.12 Method: MWS-RPIM, Parameters: $k = 4$, $mm = 3$, $d_{max} = 4.5$ 84

4.13 Device parameters of a PN junction diode used in the simulation. 85

TH-2607_126102004

5.1 Speedup Analysis for IBM benchmarks on Intel Xeon Phi 100

5.2 Various device parameters used for a PN diode 102

TH-2607_126102004

List of Acronyms

ANN	Artificial Neural Network
CAD	Computer Aided Design
CG-FEM	Continuous Galerkin Finite Element Method
DD	Drift-Diffusion
DG-FEM	Discontinuous Galerkin Finite Element Method
DIBL	Drain Induced Barrier Lowering
DNN	Deep Neural Network
DOF	Degree Of Freedom
EDA	Electronic Design Automation
EFG	Element Free Galerkin
FDM	Finite Difference Method
FEM	Finite Element Method
FVM	Finite Volume Method
GAAFET	Gate All Around Field Effect Transistor
GPU	Graphic Processing Unit
LUT	Look Up Table
ML	Machine Learning
MLS	Moving Least Square
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
PDE	Partial Differential Equation
RPIM	Radial Point Interpolation Method
RW	Random Walk

TH-2607_126102004

SUPG	Streamlined Upwind Pertrov-Galerkin
SVM	Support Vector Machine
VLSI	Very Large Scale Integration
TCAD	Technology Computer Aided Design

TH-2607_126102004

1

Introduction

Contents

1.1 Introduction 2
1.2 Thesis Motivation and Objective 7
1.3 Thesis Contributions 8
1.4 Thesis Organization 9

TH-2607_126102004

1.1 Introduction

The semiconductor industry has observed Moore's law in a pivotal role for the economic and technological advancements since last many decades. Efficient and economical product design is accomplished through transistor scaling, which enables the exponential growth of the semiconductor market by designing smaller, faster and ultra-low power circuits. This ever-growing demand to efficiently incorporate multiple diverse functionalities in a single electronic system has created a new dimension called *More than Moore* to sustain the growth of the semiconductor industry. Therefore, the VLSI community now endeavors to explore the possibilities of designing even smaller and faster devices with low power consumption, which leads to initiate the search for novel devices with the desired features.

In the present era, the semiconductor industry is capable of designing and fabricating electronic circuits at $10nm$ FinFET technology node and is aiming towards 7-$5nm$ FinFET fabrication technologies. At the same time, academia is exploring the feasibility of the semiconductor device design and fabrication at lower technology nodes using Silicon or other suitable materials, such as Silicene, Graphene, MoS_2, and WS_2. Transistors at this regime tend to lose gate control over the channel making it more susceptible to leakage current and abrupt switching modes. The quest for designing sub-nano regime devices poses several challenges, such as high leakage current, high drain induced barrier lowering (DIBL) and hot electron effect, etc. The introduction of new materials and their complex physical properties at sub $5nm$ poses another challenge at Technology Computer Aided Design (TCAD) for modeling their physical phenomena to analyze the electrical behavior of a device. To resolve these issues in designing electronic circuits, it becomes imperative to incorporate these effects in TCAD through mathematical modeling. As we know, TCAD has enabled the scientific community to take a big leap in semiconductor device technology within a short span of time. The given pace of semiconductor industry demands a highly accurate TCAD tool which can precisely analyze a semiconductor device close to the actual characterized results obtained after device fabrication. By employing TCAD tools to analyze semiconductor devices, realistic mathematical models may be developed if it provides reliable and accurate solution. Thus, TCAD helps in reducing the path of device

TH-2607_126102004

design and development of novel devices by skipping various cumbersome steps of the manufacturing process.

A TCAD tool is basically a framework which bundles a set of complex mathematical equations whose solution provides the desired outcome. The steps required to build this framework for device analysis is shown in Fig. 1.1, which explains the simulation process by incorporating several device models [3]. The simulation process is initiated by procuring all data related to device geometry, material parameters, doping profile and necessary boundary conditions. Device geometry, also referred as whole domain, is further divided into smaller domains or segments to solve PDEs locally to complete the process of discretization. Discretization generates a mesh throughout the whole geometry representing interconnections between neighboring nodes. Therefore, the discretization process converts a nonlinear partial differential equation to its linear algebraic equivalent which is analyzed iteratively until a stable solution is obtained. For finding the numerical solution, charge is computed using an initial guess of the solution. The charge calculated in the previous step is then used to analyze Poisson's and Continuity equations iteratively until the solution converges fulfilling threshold criterion. Gummel's and Newton-Raphson algorithms [3] are employed to linearize both Poisson's and continuity equations. Different electrical properties, such as current, electrical field, potential profile, etc. are calculated at the end of simulation process for the specified input parameters.

The five fundamental physics equations for Drift-Diffusion model to analyze semiconductor devices can be represented as follows:

(i) **Poisson's equation**

$$\bigtriangledown \cdot (-\epsilon \bigtriangledown \phi) = \rho \tag{1.1}$$

$$\rho = q[p - n + N_d - N_a]$$

(ii) **Current equations**

$$J_n = qn\mu_n \bigtriangledown \phi + qD_n \bigtriangledown n \tag{1.2}$$

$$J_p = qp\mu_p \bigtriangledown \phi - qD_p \bigtriangledown p \tag{1.3}$$

TH-2607_126102004

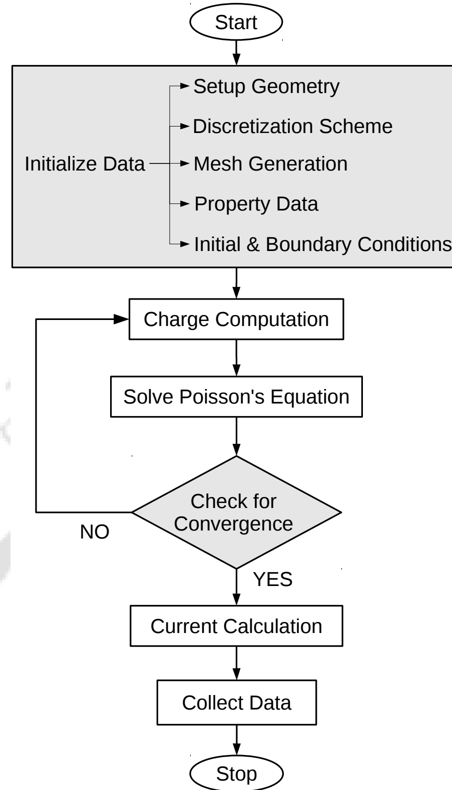


Figure 1.1: The process flow of a typical TCAD framework for device analysis.

(iii) **Continuity equations**

$$\frac{\partial n}{\partial t} = \frac{1}{q} \bigtriangledown \cdot J_n + R_n \tag{1.4}$$

$$\frac{\partial p}{\partial t} = -\frac{1}{q} \bigtriangledown \cdot J_p + R_p \tag{1.5}$$

where ϕ is potential; ϵ is dielectric Permittivity; ρ is space charge density; μ is mobility; D is diffusion coefficient; q is electrostatic charge; n and p specify electron and hole density in conduction and valance band respectively; R is net generation and recombination rate; N_a and N_d are acceptor and donor concentrations respectively; J is current density; and t denotes time.

The accuracy of TCAD framework mainly depends on two factors; 1) Transport models incorporated for the analysis, 2) Discretization scheme employed to analyze carrier transport models. Several mathematical models have been proposed in the literature [1,4] to analyze the transport mechanism of semiconductor devices, such as semi-classical drift-diffusion, quantum-ballistic, hydrodynamic models etc., which consist of a set of complex mathematical equations. For emulating the real behavior of

TH-2607_126102004

semiconductor devices, it becomes imperative to include second-order effects in the transport models as well, which further increase their computational complexity by manifolds. To obtain the solution of these equations in a reasonable time, we have no other choice except to trade off computational time with accuracy. As we know, the scientific community is exploring different ways to fabricate semiconductor devices at sub $5nm$ level, computational complexity for the accurate analysis of these devices has increased even without incorporating second order effects. The increased level of complexity constraints us to employ highly accurate discretization schemes for solving coupled PDEs while analyzing semiconductor devices.

Several discretization schemes have been presented in the literature, such as Finite Difference Method (FDM), Finite Element Method (FEM), Finite Volume Method (FVM) for the analysis of semiconductor devices [1, 3, 5–8]. These methods discretize partial differential equations (PDEs) over a given geometry to formulate a set of linear algebraic equations and obtain an approximate numerical solution with the help of predefined boundary conditions [9]. Since the last many decades, FDM is being employed to analyze differential equations efficiently in terms of computational time with a limited accuracy. It is widely used along with more efficient and robust methods, such as FEM and FVM. These methods provide an efficient solution in case of high order PDEs and unstructured domains and are considered for the analysis of semiconductor devices [7, 8, 10].

Many other methods have been employed in this work to further improve the accuracy of FEM, such as Discontinuous Galerkin method, Streamline Upwind Petrov-Galerkin Stabilization Technique etc. Although these methods improvise the solution, a numerical error still exists as compared to the analytical solution because of the predefined mesh structure. Discretization process divides the problem domain into finite small elements to form a mesh-like structure for establishing a relationship among nodes of the mesh. The dependency on a predefined mesh structure makes the discretization process susceptible to the numerical errors. For eliminating this dependency, meshfree or element-free Galerkin method (EFG) was introduced in [11–13]. The efficiency of the meshfree method has been illustrated in literature while analyzing mechanical structures in [14, 15]. Study of the EFG method provides impetus to extend its application for the analysis of semiconductor devices.

The improved accuracy of the TCAD simulations comes at a price of higher computation time,

TH-2607_126102004

which motivates us to develop a methodology for the accurate solution without sacrificing much time. This thesis deals with exploring various methods to reduce the computation time while maintaining the desired accuracy. *Initial guess* which is one of the most important parts in the device simulation is studied in the proposed work to improve computation time for the analysis of semiconductor devices. The impact of choosing an initial guess on the quality of final solution is studied in detail.

Two different approaches, *Random Walk* method [16, 17] and application of *Machine Learning* [18–20] in the analysis of semiconductor devices, are proposed in this thesis to improve the initial guess during semiconductor device analysis. In the case of random walk implementation, equivalent electrical circuits for all the involved PDEs required for the device analysis are proposed and analyzed using random walk method by utilizing its embarrassing scope of parallelization. The application of random walk method for the analysis of very large electrical circuits and power grid networks has been reported in the literature [21–24]. Further, the electrical analogy of various partial differential equations representing semiconductor device behaviour [25–28] motivated us to introduce application of random walk in the analysis of semiconductor devices.

The successful implementation of random walk in the analysis of semiconductor devices motivated us to explore non-traditional methods to improve initial guess during the analysis of semiconductor devices. In order to achieve the above mentioned goal, different machine learning algorithms have been studied for the analysis of semiconductor devices. The application of machine learning algorithms in the domain of yield enhancement during fabrication process of semiconductor devices has been reported in the literature [29–31], which has motivated us to incorporate it in the semiconductor device analysis as well. An attempt to model electrical properties of semiconductor devices using machine learning has already been reported [32–35] but in these works ANN is used to create a model and to peform study of interest, whereas in our proposed methodology to analyze semiconductor devices, machine learning is employed to analyze steps of semiconductor device analysis to speed it up. We have also developed a platform for ANN based semiconductor device modeling with a better accuracy and have extended its application for the analysis of electrical circuits composed of MOSFETs and basic devices. This proposed framework exploits ANN based semiconductor device models to analyze circuits and is validated against commercially available electronic design automa-

TH-2607_126102004

tion (EDA) tool *Spectre* [36]. It is observed that the proposed framework is very efficient in terms of memory utilization and computation time as compared with lookup table based approach [37–39]. This is to be reported that the ANN based semiconductor device models and machine learning based the circuit analysis have accuracy within 0.1% as compared to the BSIM models and commercially available circuit simulators.

It should also be noted, the work presented in this thesis addresses the issues related to the fast and accurate analysis of semiconductor devices. It also reduces dependency on the explicit development of models of novel semiconductor devices. This work paves the way to study the effect of parametric variation on the performance of electrical circuit composed of novel devices in the early stage of design cycle. It is envisaged that this would help in the development of next generation mature semiconductor devices. The present study also motivates us to eliminate the need of traditional semiconductor device modeling and, to aim towards ANN or Deep Neural Network (DNN) or Spiking Neural Network (SNN) powered semiconductor device models and machine learning based next generation electronic design automation tool development.

1.2 Thesis Motivation and Objective

Technology is steering the wheels of our rapidly growing needs and expectations. Advancement in the field of science or any other domain has been made possible only because of the aggressive nature of current technology, which is backed up by the enormous growth of the semiconductor industry. Transistor sizing is the most important paradigm for improving the speed and performance of semiconductor devices. Small size transistors have enabled us to house more number of devices in a given die area while improving the power and delay parameters per computation. The current semiconductor industry operates at a technology node of 14 nm-10 nm and thriving towards sub-5 nm devices. Over the last five decades, Moore's law has been the driving force of the semiconductor industry but it is currently facing challenges due to the prominence of second-order effects in the sub-nanometer regime devices. Transistors at this regime tend to lose gate control over the channel making it more susceptible to leakage current and abrupt switching modes, due to a phenomenon called Quantum tunneling. To improve these issues while continuing our path with Moore's law,

TH-2607_126102004

research is being carried out to employ Graphene, carbon nano tube and molybdenum disulfide in future transistors of sub-5 nm regime. The introduction of new materials and their complex physical properties at sub-5 nm poses another challenge at TCAD for modeling the physical phenomena to analyze the electrical behavior of a device. It is because of the use of TCAD only, we have been able to take such a big leap in technology in such a short span of time, and thus to continue with this pace we need to depend on the highly accurate TCAD tools which can precisely find solutions for a given device efficiently.

The main objective of this thesis is to design and develop an efficient framework for fast and accurate semiconductor device analysis. In order to achieve these goals, the main objectives which need to be accomplished are given below.

- Design of a framework based on FEM based discretization method
- Improvise the solution provided by FEM using different stabilization methods, such as DG-FEM and SUPG
- Implementation of Element-Free Galerkin method to improve accuracy of the solution
- Application of machine learning and random walk method to improve initial guess of the solution

1.3 Thesis Contributions

In this thesis, different methodologies have been proposed to design and develop a framework for the analysis of semiconductor devices. The two main challenges of TCAD, *accuracy* and *computation time*, have been the motivating force to contribute in this area. The main contribution of the thesis can be broadly classified into two parts. In the first part, different methodologies, such as discretization schemes and meshing techniques, are employed to improve the solution accuracy of the semiconductor device simulations without sacrificing much computation time. The second part of the thesis explores alternate methodologies to minimize the computation time enforcing accuracy to be as high as possible. The key contributions of the thesis are described below.

TH-2607_126102004

- A 2D/3D framework is developed to analyze semiconductor devices based on finite element method.
- DG-FEM and SUPG stabilization techniques are employed to improve accuracy of the solution as compared to FDM and FEM.
- A highly accurate and adaptive framework is developed to analyze semiconductor devices based on Element-Free Galerkin method.
- Equivalent electrical circuits of various PDEs are proposed and a circuit analysis methodology based on random walk is employed to analyze these circuits, which helps to reduce the computation time by providing improved initial guess.
- Different machine learning algorithms are applied to predict potential profile inside a device which is utilized as an initial guess to analyze the proposed device structure efficiently.
- A circuit analyzer is designed for the analysis of electrical circuits composed of semiconductor devices represented by ANNs.

1.4 Thesis Organization

The organization of rest of the thesis is presented below.

Chapter 2: Fundamentals of TCAD in Semiconductor Device Analysis

In this chapter, different aspects of TCAD, such as discretization schemes and meshing techniques are discussed. The study is extended to briefly describe the strength and weaknesses of different discretization schemes for a particular set of problems. Different meshing techniques to further improvise the solution accuracy are also introduced in this chapter. It is to be noted that the choice of a good discretization scheme and a meshing technique can improve the solution accuracy to a great extent.

TH-2607_126102004

Chapter 3: Implementation of FEM and its Variants for Device Analysis

In this chapter, a framework to analyze semiconductor devices is presented using FEM and its variants to discretize the PDEs arising from semiconductor device modeling. DG-FEM and SUPG-FEM are the variants of FEM which are described in this chapter. In DG-FEM, the basis functions used for discretization are discontinuous piecewise polynomials which aids to conserve the flux and handles the complex geometries efficiently. SUPG stabilization technique analyzes semiconductor devices more efficiently as compared to the classical Scharfetter-Gummel method. To validate the effectiveness of the proposed methods, analysis of different semiconductor devices, such as PN junction diode, MOS capacitor, and MOSFET are demonstrated.

Chapter 4: An Accurate and Adaptive Framework based on EFG Method

This chapter presents an implementation of a highly accurate and adaptive framework based on the Element-Free Galerkin method to analyze semiconductor devices. In the EFG method, a set of nodes are scattered throughout the domain to formulate algebraic equations representing carrier transport in semiconductor devices without employing mesh for domain discretization. The correlation among neighboring field nodes for mesh generation is not required apriori to formulate linear algebraic equations making it more suitable for adaptability. The primary difference between FEM and EFG lies in the construction of their shape functions. In FEM, elements are used to construct shape functions which are identical for all the similar elements. These shape functions can be chosen in the beginning of analysis by predetermining their values. However, in the EFG method, shape functions are constructed uniquely at each point of interest to achieve more accurate results. The solution provided by the proposed methodology is compared with other discretization techniques including the methodology adopted by commercial TCAD simulator, *Sentaurus* [2]. Accuracy of the solution obtained by the proposed methodology is $\approx 10\times$ better as compared to the solution provided by *Sentauras*. The effectiveness of the proposed method is validated by comparing the outcome of the analysis with the analytical solution and device TCAD *Sentauras*.

TH-2607_126102004

Chapter 5: Random Walk and Machine Learning Algorithms based Analysis of Semiconductor Devices

In this chapter, random walk and machine learning based approaches are proposed to speed up the computation time of the device analysis. In the case of the random walk method, equivalent electrical circuits representing fundamental device equations are analyzed in a suitable manner. Using the solution of this electrical circuit, solution of semiconductor device analysis is formulated. The analysis of semiconductor devices using the random walk method is validated by analyzing various semiconductor devices. In the machine learning based approach, artificial neural networks (ANNs) are utilized to create a model to predict the potential profile inside a device for a particular set of parameters. Later ANN based semiconductor device models are employed to analyze electrical circuits composed of semiconductor devices. Details of this analysis and the circuit analyzer developed specifically for the analysis of ANN based semiconductor devices are presented in this chapter.

Chapter 6: Conclusion and Future Aspects

In this chapter, a conclusion is drawn on the various methods and framework developed for the efficient and accurate analysis of semiconductor devices. As we have stated that for the novel devices, it is imperative to design and develop an efficient TCAD framework envisaging the complex behavior of these devices. Therefore, the future scope presented in this chapter incorporates a brief discussion on this issue along with different possible applications of the proposed methodologies.

TH-2607_126102004

2

Fundamentals of TCAD in Semiconductor Device Analysis

Contents

2.1 Introduction 14
2.2 Physical Models for Device Analysis 17
2.3 A Survey of Different Discretization Methods to Implement Transport Models 24
2.4 Formulation of Basic Device Equations 26
2.5 Numerical Solution of Linear Algebraic Equations 29
2.6 Summary 30

TH-2607_126102004

In this chapter, the fundamentals of TCAD for the analysis of semiconductor devices are discussed in detail. Various aspects of TCAD starting from device dimensions, its physical properties, discretization schemes and meshing techniques to different carrier transport models are discussed in order to find the accurate solution of the involved partial differential equations (PDEs). The study has been extended to briefly describe the strength and weaknesses of different discretization schemes for a particular set of problems. Different meshing techniques and its impact on the solution's accuracy has also been introduced in this chapter. The choice of a good discretization and meshing scheme can improve the solution accuracy to a great extent.

2.1 Introduction

Whenever we name the greatest innovations of the twentieth century, we undoubtedly call upon *The Electronics Industry* which is backed-up by the advancements in semiconductor devices. TCAD plays a vital role in bringing the electronic industry to the current level by reducing the design and development cycle of novel devices with different geometries and materials. **T**echnology **C**omputer **A**ided **D**esign, as the name suggests, it uses the processing power of high-end computers to efficiently design and analyze semiconductor devices to approximately predict their electrical behavior by implementing various device physics models, such as Drift-Diffusion, Hydrodynamics, Boltzmann (Monte Carlo), Quantum Corrected Boltzmann, and Non-equilibrium Greens Function [4] etc. These models comprise of a set of coupled non-linear partial differential equations to compute potential and charge-carrier density. All these models may target a different class of semiconductor devices, and varies with respect to a given accuracy and computation time. Drift-diffusion is the most widely used fundamental model to analyze sub-micron devices. A good TCAD framework must be able to incorporate all these models to design semiconductor devices with the desired level of accuracy.

The given pace of semiconductor industry demands a highly accurate TCAD framework which can precisely analyze a semiconductor device as compared to the actual characterized results obtained after device fabrication. The critical role of TCAD in the manufacturing process of semiconductor devices can be realized using Fig. 2.1. By employing TCAD tools to analyze semiconductor devices, realistic mathematical models may be developed if it provides a reliable and accurate solution. TCAD

TH-2607_126102004

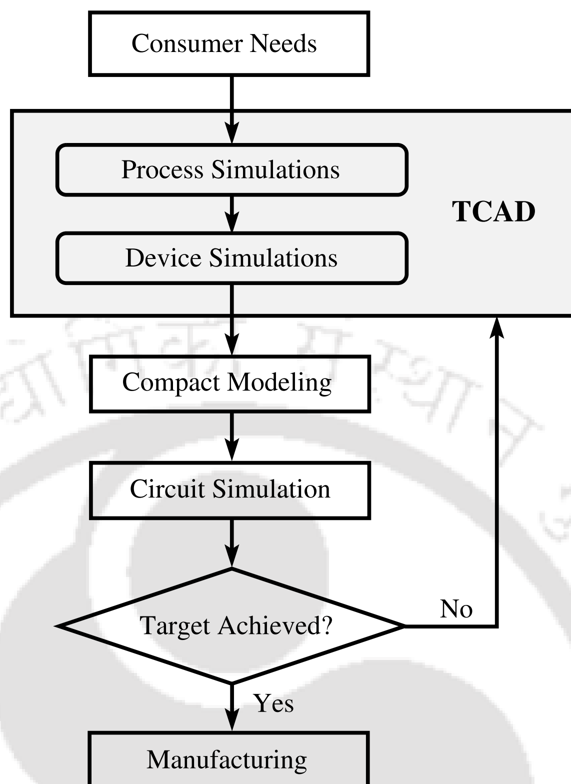


Figure 2.1: Role of TCAD in manufacturing process.

helps in reducing the cycle of design and development of novel devices by skipping various cumbersome steps of the manufacturing process as shown in Fig. 2.2 exhibiting various steps involved in the manufacturing process of a typical semiconductor device. It can be observed from Fig. 2.2, that the steps involved in the development of a device are time-consuming and irreversible. TCAD helps bypass all these steps and provide approximate guidelines for the best possible device geometry and materials of the future semiconductor devices.

An extensive survey of the literature has motivated author to design and develop an efficient framework for fast and accurate semiconductor device analyzer. It has been observed in the proposed study that the currently available TCAD frameworks are not enough to efficiently analyze complex and nano-regime devices, such as multi-gate and 2D devices. The analysis of such devices requires computations of higher order nonlinear differential equations, which further needs more robust framework to efficiently analyze their behavior. As we know, an increase in accuracy increases computation time and it is observed in the literature too that several parallelization techniques are used to reduce com-

TH-2607_126102004

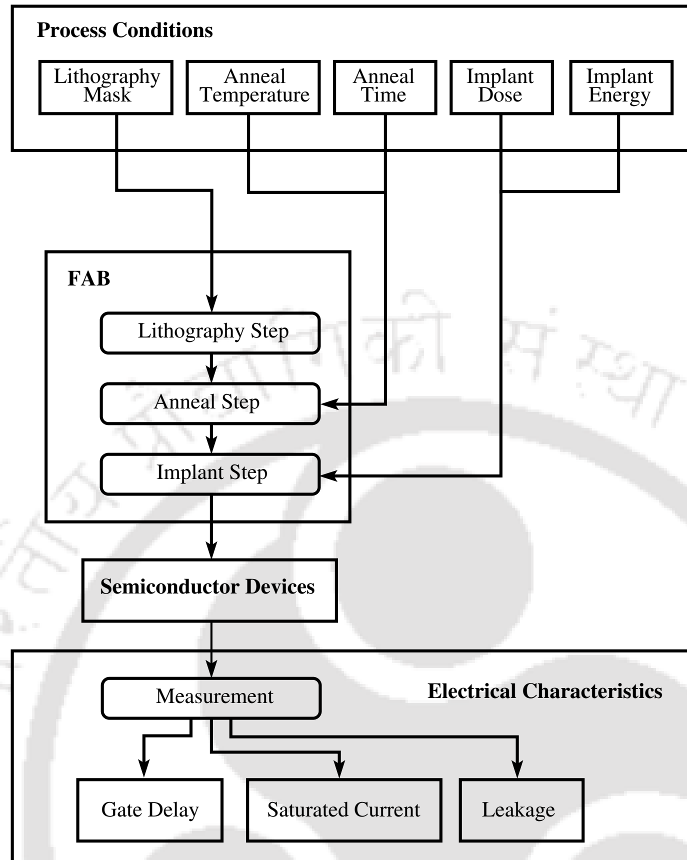


Figure 2.2: The manufacturing process of a typical semiconductor device.

putation time. Therefore, in the proposed work, author had incorporated novel techniques, such as Random Walk and Machine Learning to target computational aspect of analysis.

There are mainly two principle traits, *Accuracy* and *Computation Time*, of a TCAD simulation framework to define its efficiency. An ideal framework should provide the highest level of accurate with the least possible computation time, which is nearly impossible to achieve since there is always a trade-off between accuracy and computation time. This particular challenge of developing an ideal framework has intrigued us to work in this area. In the quest to develop such an ideal framework, different aspects of TCAD have been studied to find the best possible scope of improvement. Thus the work presented in this thesis can be broadly classified to target both *Accuracy* and *Computation Time* aspect of TCAD. In the pursuit to improve accuracy of the framework, its two primary factors; 1) Transport models incorporated for the analysis, 2) Discretization scheme employed to solve the transport models, are discussed in detail in this thesis. The primary aim of a transport model is to

TH-2607_126102004

capture the effect of every single physical phenomenon of a semiconductor device. Thus, a transport model basically bundles a set of complex mathematical equations. For modeling the real behavior of a device, it becomes imperative to include second-order effects of sub-nano regime devices in the transport models, which further increases their computational complexity by manifolds. This increased level of complexity constraints us to employ highly accurate discretization methods to solve transport models for the analysis of semiconductor devices. A brief overview of different transport models employed to analyze semiconductor devices is presented in the next section.

Since the highly accurate analysis is compute intensive, the second part of the thesis is aimed to improve the computation time of the analysis. Different aspects to speed up of the analysis have been studied and are implemented. The parallelization of the computations has been introduced wherever possible to harness computational power of the underlying hardware, and the corresponding results are presented in this thesis. Two different novel methods; *Random Walk* and *Machine Learning*, have also been proposed to speed up the analysis by predetermining a very good initial guess to the solution. This leads us to present different methods in this thesis to develop a very efficient TCAD framework for the analysis of semiconductor devices.

2.2 Physical Models for Device Analysis

Based on certain characteristics and phenomenons like channel length modulation, electron-electron scattering, electron-phonon scattering and other possible applications, device modeling can be categorized into multiple domains. Starting from the basic fundamental transport phenomenons to more complex ones, the hierarchy of different transport models can be presented as shown in Fig. 2.3. The semi-classical Boltzmann model for the transportation of charge carriers has been the mainstay of semiconductor technology from the very beginning. Most of the device simulations are based on the solution of Boltzmann equation (BTE) or its simplified versions, such as Drift-Diffusion (DD) model and Hydrodynamic (HD) transport model.

However, with the advent of technology with reference to down-sizing of device dimension into the nanoscale regime, many new interesting effects have become dominant, such as (i) velocity overshoot, (ii) hot-electron effects, (iii) ballistic transport, (iv) quantum transport, (v) hot-electron

TH-2607_126102004

thermionic emission, and (vi) interfaces, surfaces and metal boundary effects [1] etc. Therefore, it can be concluded that the semi-classical device models are no longer valid to work efficiently in the nanoscale devices. The introduction of quantum effects in the nano regime leads to lay the foundation of different theories or models that can very well treat quantum transport phenomenon between macroscopic and microscopic scale. Therefore, among the numerous available physical models for the analysis of semiconductor devices, most efficient and popular models are listed below [1], [40].

(i) Drift-Diffusion Model

(ii) Hydrodynamics Model

(iii) Thermodynamics Model

(iv) Ballistic Model

(v) Quantum Model

Approximate ↑ Easy, Fast

	Model	Improvements
Semiclassical approaches	Compact models	Appropriate for circuit design
	Drift-Diffusion equations	Good for device down to 0.5 μm
	Hydrodynamics equations	Velocity overshoot effect can be treated properly
	Boltzmann transport equation Monte Carlo/CA methods	Accurate up to classical limits
Quantum approaches	Quantum Hydrodynamics	Keep all classical hydrodynamics features + quantum corrections
	Quantum Monte Carlo/CA methods	Keep all classical features + quantum corrections
	Quantum-Kinetic equations (Liouville, Wigner-Boltzmann)	Accurate up to single particle description
	Green's function method	Includes correlation in both space and time domain
	Direct solution of the n-body Schrodinger equation	Can be solved only for small number of particles

Exact ↓ Difficult

Figure 2.3: Shows the hierarchy of various transport models [1]

In the proposed work, we use the fundamental drift-diffusion model to study various aspects of accuracy and computation time for the analysis of semiconductor devices. Since this model is incorporated with the commercial TCAD simulators as a basic model, comparison of our proposed methodologies using fundamental drift-diffusion model with the exiting TCAD simulators can be performed reliably.

2.2.1 Drift-Diffusion Model

As the name suggests, this fundamental model works on two different phenomenons, *drift* and *diffusion*, in order to devise mathematical equations for the transportation of charged carriers in a semiconductor device. The *drift* phenomenon corresponds to the movement of carriers under the effect of electric field., whereas, the movement of carriers with respect to density gradient is defined by *diffusion*. The popular drift-diffusion current equations can be easily derived employing Boltzmann transport (BTE) equation by considering moments of the BTE [41]. It is based on the five basic equations namely, (i) current equations, (ii) continuity equations and (iii) Poisson's equation, and are mentioned below.

(i) **Poisson's equation**

$$\triangledown \cdot (-\epsilon \triangledown \phi) = \rho$$

$$where \;\; \rho = q[p - n + N_d - N_a]$$

(ii) **Current equations**

$$J_n = qn\mu_n E + qD_n \triangledown n$$

$$J_p = qp\mu_p E - qD_p \triangledown p$$

(iii) **Continuity equations**

$$\frac{\partial n}{\partial t} = \frac{1}{q} \triangledown \cdot J_n + U_n$$
$$\frac{\partial p}{\partial t} = -\frac{1}{q} \triangledown \cdot J_p + U_p$$

where μ is mobility; D is the diffusion coefficient; U is net generation and recombination rate; and ϕ is potential.

TH-2607_126102004

The Poisson's equation computes the potential profile or electric filed inside a semiconductor device for a given amount of charge distribution, whereas, current equation calculates charge density using drift and diffusion gradients, and the concentration of charge carriers with respect to time is computed using the continuity equation.

2.2.1.1 Convergence Constraints on Drift-diffusion model

When the partial differential equations arising from the Drift Diffusion model are discretized using the finite difference method, the following constraints need to be enforced in order to get the system of equations converged to the final solution [42]

(i) The mesh size (Δh) is limited by Debye length.

(ii) The time step is limited by the dielectric relaxation time.

Debye length can be defined as the distance over which a significant charge separation can occur, therefore, the mesh size must be smaller than Debye length to resolve the issue of charge variation in space. Debye length, L_D, can be represented as,

$$L_D = \sqrt{\frac{\varepsilon k_B T}{q^2 N'}}$$

In GaAs and Si at room temperature, the Debye length is approximately only $50\ \mathring{A}$ when $N = 10^{18} cm^{-3}$ and increases to $400\ \mathring{A}$ when $N = 10^{16} cm^{-3}$.

The dielectric relaxation time is the characteristics time of the charge fluctuations to decay under the influence of its own field. It can be represented as,

$$t_{dr} = \frac{\varepsilon}{qN\mu}$$

As the mobility increases, dielectric relaxation time decreases rapidly. In GaAs at a mobility of $6000\ cm^2/V - s$ and $N = 10^{18} cm^{-3}$ the dielectric relaxation time is $10^{-15}\ s$.

TH-2607_126102004

2.2.2 Mesh Formation

The formation of mesh or meshing in a semiconductor device is the most fundamental process step in order to perform device simulation successfully. However, meshing is a discretization dependent scheme and is discussed extensively in the chapters 3 and 4. Here, the fundamental concept of meshing is discussed in brief. During meshing, the whole device domain is represented using several nodes and their interconnections as shown in Fig. 2.4. The fundamental device equations are solved simultaneously and iteratively on these nodes to compute the reliable electrical characteristics. The mesh size need not be of equal dimension in all directions and may depend upon the amount of accuracy needed on any particular device region. Fig. 2.4 shows the nonuniform mesh formation in case of a MOSFET. A dense mesh in a particular region means more number of nodes or refined meshes in that region to obtain higher order of accuracy for the parameters under study. On the other hand, in the regions where variation in the solution or parameters under study is not significant, a coarse meshing may be employed. Thus, a proper choice of dense and coarse meshing in different regions is desired for an efficient analysis while reducing computation time and increasing accuracy of the solution.

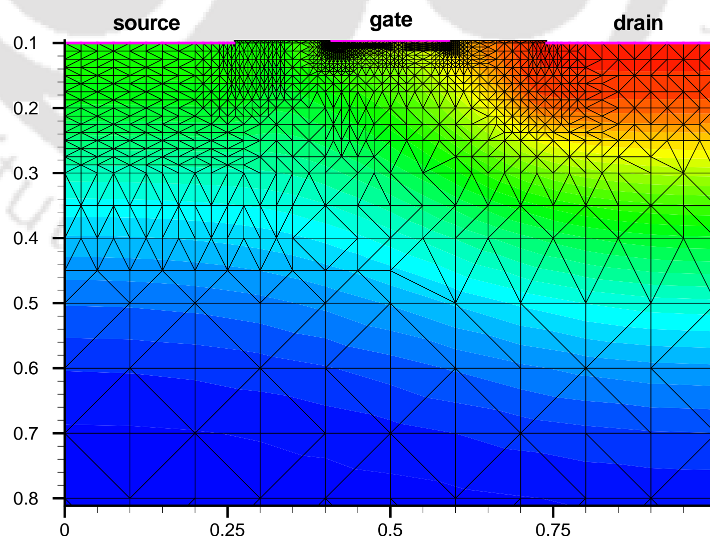


Figure 2.4: Non-Uniform mesh formation in case of MOSFET [2]

The process of assigning given parameters and numerically analyzing fundamental device equa-

TH-2607_126102004

tions on each node points of a mesh with definitive boundary conditions is demonstrated by Fig. 2.5. It represents mapping of the parameters from a Poisson's equation which is a second order partial differential equation and is analyzed numerically at each mesh node. The computational complexity of a 2 dimensional PDE increase with a fine meshing scheme to get more accurate results. The computational complexity increases further when we go for the multidimensional solution of a PDE.

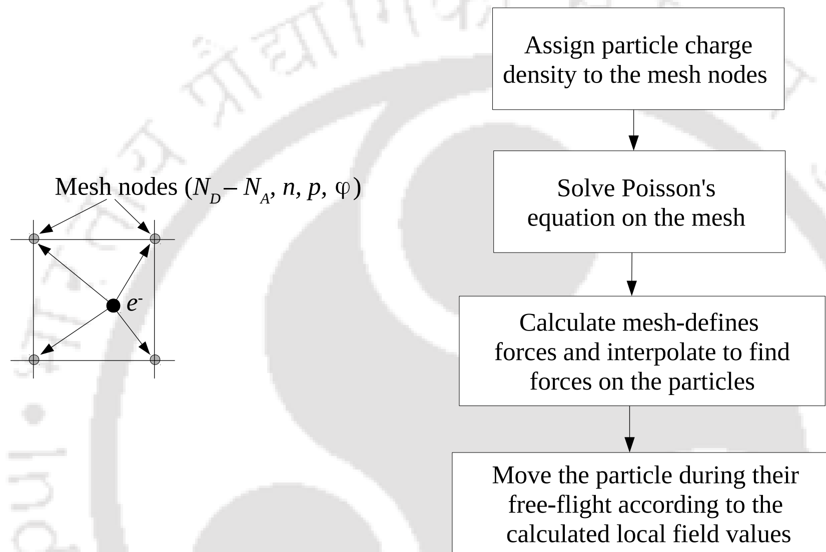


Figure 2.5: Steps to solve fundamental device equation on each individual node.

2.2.3 Boundary Conditions

The solution of a system of partial differential equations is computed numerically only with the help of different boundary conditions. A boundary condition defines a specific value to the PDE at the boundaries or at the interfaces. Three different boundary conditions have been studied for the proper analysis of PDEs in semiconductor devices as shown in Fig. 2.6, and are described below.

2.2.3.1 Dirichlet Boundary Condition

This boundary condition implies that the solution of the unknown function is exactly known at that particular boundary or interface. For a region bounded by $x \in [0, 1]$, Dirichlet boundary condition

TH-2607_126102004

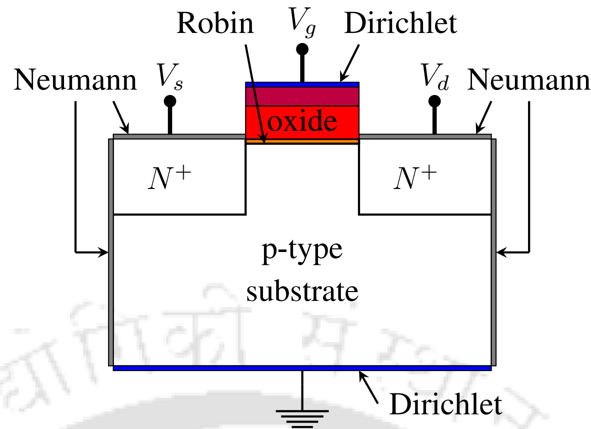


Figure 2.6: Different boundary conditions for the analysis of semiconductor devices.

can be represented by equation 2.2.

$$\frac{d^2(u)}{dx^2} = f \quad x \in [0,1] \tag{2.1}$$

$$\begin{aligned} u_{(x=0)} &= C_1 \text{ [Constant]} \\ u_{(x=1)} &= C_2 \text{ [Constant]} \end{aligned} \tag{2.2}$$

2.2.3.2 Neumann Boundary Condition

In the Neumann Boundary condition, the gradient of the unknown function is exactly known rather that the function itself and can be represented by equation 2.5.

$$\frac{d^2(u)}{dx^2} = f \quad x \in [0,1] \tag{2.3}$$

$$\frac{d(u)}{dx}\Big|_{(x=0)} = C_3 \text{ [Constant]} \tag{2.4}$$

2.2.3.3 Robin Boundary Conditions

This particular boundary condition is only implemented at the interface between two different materials where it is not possible to apply any of the two above mentioned conditions, as it can be seen from Fig. 2.6. Robin boundary condition is a weighted combination of both Neumann and

TH-2607_126102004

Dirichlet boundary condition, and can be represent as,

$$a\frac{d(u)}{dx} + bu \quad = \quad C_4 \tag{2.5}$$

2.3 A Survey of Different Discretization Methods to Implement Transport Models

The core of any transport model bundles a set of complex mathematical equations for its analysis, which in the case of semiconductor device analysis, are a set of several partial differential equations. This section discusses several methods to compute the solution of a PDE. It is well known that a PDE in its raw form is called the strong-form, and finding an exact solution of a strong-form PDE is computationally intensive for the practical engineering problems due to its complexity. The complex nature of the strong form of a PDE demands to first convert it to a weaker form to get the solution. As discussed in the earlier section, for the analysis of PDEs, three types of boundary conditions, Neumann, Dirichlet or Robin may be used. Finite difference method is a well-known numerical method suitable for the analysis of strong-form PDEs with Dirichlet boundary conditions for producing accurate results. However, in the case of Neumann or derivative boundary conditions, not only the accuracy of the solution deteriorates but also it becomes unstable. This is because of the error introduced in FDM due to ignoring higher order terms while discretizing PDEs into linear algebraic equations.

A sufficient degree of consistency is assumed to approximate the unknown function of a strong-form PDE, i.e., it should be differentiable up to the highest order of the given PDEs. On the other hand, weak-form PDEs require a weaker consistency of the approximate function. Discretization of weak form PDEs can produce a stable set of equations to achieve better solution accuracy. The central principle behind the creation of weak-form is the weighted residual method which is a more general and powerful mathematical tool developed to discretize system equations. Different methods have been reported in the literature based on the use of strong and weak forms of partial differential equations, as shown in Fig. 2.7.

As it is shown in Fig. 2.7, many well-known methods, such as FDM, FEM, FVM and Meshfree Weak Strong (MWS) etc. have been developed in the past for the solution of PDEs. All of these

TH-2607_126102004

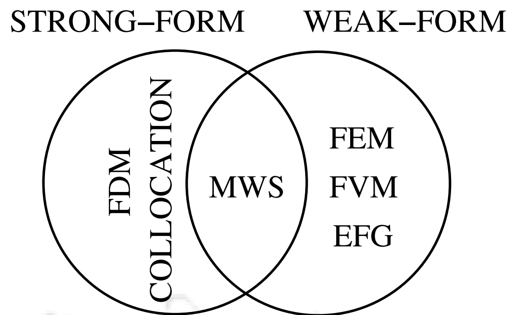


Figure 2.7: Venn diagram showing relationship between different methods.

methods depend on mesh generation for discretization, which makes them susceptible to numerical errors. These issues related to their dependency on mesh generation can be addressed by employing Element-free Galerkin method to formulate a system of linear algebraic equations. Due to the absence of finite elements in the discretization process or its non-dependency on mesh generation, it is called the element free method. It is to be noted that EFG and meshfree are used interchangeably in this thesis. A comparative study of EFG and other discretization schemes is presented in Table 2.1 to venture its dominance using four vital parameters termed as mesh generation, adaptability, range of problems and accuracy [43], [44]. The prime focus of the EFG method is to provide an accurate and more stable numerical solution for PDEs with all kinds of possible boundary conditions. Along with the accuracy, one more peculiar feature of EFG method is its ease of doing the adaptive simulation. The other methods, Collocation and MWS, are also based on the meshfree approach. The Collocation method depends on the strong-form formulation, therefore, produces a desirable level of accuracy with Dirichlet boundary conditions only [45], [46]. Meshfree Weak-Strong (MWS) method can be described as a hybrid method employing weak-form at boundaries with Neumann boundaries and strong-form in the rest of the domain.

In the search for an approximate solution of PDEs with boundary conditions, the unknown function to be determined is approximated using shape functions before discretizing PDEs into linear algebraic equations [47]. Different shape functions employed for the discretization have been discussed in details in later chapters 3 and 4. Two different shape functions, *Moving Least Squares* (MLS) and *Radial Point Interpolation Method* (RPIM) are used in the proposed work for the implementation of EFG method, and are described in Chapter 4.

TH-2607_126102004

Table 2.1: Characteristics analysis of different discretization scheme.

Properties	**Discretization Schemes**				
	FDM	**FEM**	**Collocation**	**MWS**	**EFG**
Complex geometry	Difficult	Easy	Difficult	Easy	Easy
Adaptability	Difficult	Difficult	Difficult	Easy	Easy
Stability of discretized equations	Stable	Stable	Not stable	Not stable	Stable
Crack growth	No	No	No	No	Yes
Solution	Less Accurate	Accurate	Moderately Accurate	Moderately Accurate	More Accurate

2.4 Formulation of Basic Device Equations

The main aim of semiconductor device modeling is to capture the effects of various physical phenomenons acting on a device in the form of mathematical equations. A collective form of these mathematical equations defines a transport model which can be analyzed consistently to perform device analysis. This section discusses certain phenomenons and their mathematical formulations which are taken as a base in this work to further improve the accuracy and computation time aspects of the device analysis.

As it is already described in Fig. 1.1, to build a framework for device analysis a specific procedure is followed [3] by incorporating several device models. At first, all the data related to material parameters, device geometry, doping profile and all the necessary boundary conditions are gathered. The whole domain (device geometry) is further divided into small domains in the form of a nonuniform mesh to solve the PDEs locally to complete the process of discretization. This step generates a mesh throughout the whole geometry to find the solution in terms of potential and charge distribution in the device. Later, the charge is calculated globally using an initial guess of the solution followed by analyzing Poisson's and carrier concentration PDEs in the coupled form. These coupled equations are solved iteratively until the solution converges fulfilling threshold criterion. The implementation of both Poisson's and continuity equations includes Gummels and Newton-Raphson algorithms [3]. At the end of the analysis, current flowing through the device is calculated for specified input parameters.

The self-consistent equations for a Drift-Diffusion model are already presented in Section 2.2.1. For analyzing these self-consistent equations, the mathematical expressions for n and p are formulated

TH-2607_126102004

in terms of other dependent variables, such as V, ϕ_n, ϕ_p. The definitions of these dependent variables arise from the concept of quasi-Fermi potentials. Here, ϕ_n and ϕ_p are the quasi-Fermi potentials for electrons and holes, respectively. Considering the case of a simple PN junction, the relationship of carrier concentration with the quasi-Fermi potential is represented below [14].

The concentration of electron in the conduction band in n region is given by,

$$n_0 = N_c \exp\left[\frac{-(E_c - E_F)}{kT}\right] \tag{2.6}$$

which can also be formulated as,

$$n_0 = n_i \exp\left[\frac{E_{Fn} - E_{Fi}}{kT}\right] \tag{2.7}$$

$$q\phi_{Fn} = E_{Fi} - E_{Fn} \tag{2.8}$$

Similarly in the p region, the hole concentration is given by

$$p_0 = N_v \exp\left[\frac{-(E_F - E_v)}{kT}\right] \tag{2.9}$$

$$p_0 = N_a = n_i \exp\left[\frac{E_{Fi} - E_{Fp}}{kT}\right] \tag{2.10}$$

$$q\phi_{Fp} = E_{Fi} - E_{Fp} \tag{2.11}$$

The built-in potential barrier, V_{bi} of a PN junction can be described in the form of quasi-fermi levels as,

$$V_{bi} = |\phi_{Fn}| + |\phi_{Fp}| \tag{2.12}$$

Using above mentioned equations of carrier concentration, the built-in potential, V_{bi}, can be calculated as,

$$V_{bi} = \frac{kT}{q} \ln\left(\frac{N_a N_d}{n_i^2}\right) \tag{2.13}$$

$$= V_t \ln\left(\frac{N_a N_d}{n_i^2}\right) \tag{2.14}$$

where E_c, E_v, E_F are the energy levels corresponding to the conduction band, valence band and

TH-2607_126102004

fermi level, respectively; E_{Fn} and E_{Fn} are the quasi-Fermi energy levels corresponding to electrons and holes, respectively; N_a and N_d are the acceptor and donor doping concentrations, respectively; N_c and N_v denote the effective density of states for conduction and valence bands, respectively; V_t = kT/q and is defined as the thermal voltage.

The Space-Charge width for a PN junction can also be described as,

$$x_p = \frac{N_d \times x_n}{N_a} \tag{2.15}$$

$$x_n = \left\{ \frac{2\epsilon_s V_{bi}}{q} \left[\frac{N_a}{N_d} \right] \left[\frac{1}{N_a + N_d} \right] \right\}^{1/2} \tag{2.16}$$

$$x_p = \left\{ \frac{2\epsilon_s V_{bi}}{q} \left[\frac{N_d}{N_a} \right] \left[\frac{1}{N_a + N_d} \right] \right\}^{1/2} \tag{2.17}$$

where x_n, x_p are the junction depletion widths in n and p region, respectively.

The total depletion width (W) is the sum of x_n and x_p and can be expressed as,

$$W = x_n + x_p \tag{2.18}$$

$$W = \left\{ \frac{2\epsilon_s V_{bi}}{q} \left[\frac{N_a + N_d}{N_a N_d} \right] \right\}^{1/2} \tag{2.19}$$

Based on the above equations, analytical model can be formulated to extract the electrical behavior of a device for a specific set of inputs. For analyzing electrical properties of the devices accurately, it is also necessary to model carrier generation-recombination and mobility instead of keeping them constant. *Shockley-Read-Hall Recombination* model is used to model generation-recombination phenomenon and the corresponding mathematical formulation can be described as,

$$U = \frac{np - n_i^2}{\tau_p \left(n + n_i \exp^{\left(\frac{E_t - E_i}{V_T}\right)} \right) + \tau_n \left(p + n_i \exp^{\left(\frac{E_i - E_t}{V_T}\right)} \right)} \tag{2.20}$$

$$\tau_n = \frac{\tau_n^0}{1 + \frac{N_A + N_D}{N_n^{ref}}}, \qquad \tau_p = \frac{\tau_p^0}{1 + \frac{N_A + N_D}{N_p^{ref}}} \tag{2.21}$$

where τ_n and τ_p denotes carrier lifetime for electrons and holes, respectively and E_t is trap energy

TH-2607_126102004

level. Table 2.2 shows the standard values for τ at equilibrium and N_{ref} as given in [1].

Table 2.2: Various device parameters used for a PN diode.

$\tau_{n0}(s)$	$N_n^{ref}(cm^{-3})$	$\tau_{p0}(s)$	$N_p^{ref}(cm^{-3})$
5×10^{-5}	5×10^{16}	5×10^{-5}	5×10^{16}

The equation used to model *low field mobility* [48] can be expressed as,

$$\mu_{\nu\perp,\parallel}^{low} = \mu_{\nu\perp,\parallel}^{min} + \frac{\mu_{\nu\perp,\parallel}^{max} - \mu_{\nu\perp,\parallel}^{min}}{1 + \left(\frac{N_D+N_A}{N_\nu^{ref}}\right)^{\alpha_\nu^\mu}} \tag{2.22}$$

$$\mu_{\nu\perp,\parallel}^{max} = \mu_{\nu\perp,\parallel,300}^{max} \cdot \left(\frac{T}{300K}\right)^{\gamma_\nu^\mu} \tag{2.23}$$

$$\mu_{\nu\perp,\parallel}^{min} = \mu_{\nu\perp,\parallel,300}^{min} \cdot \left(\frac{T}{300K}\right)^{\beta_\nu^\mu} \tag{2.24}$$

where $\nu = (n, p)$; T is temperature; (α, γ, β) are fitting parameters as given in [48].

The above mentioned analytical model is employed to develop a framework to analyze semiconductor devices by incorporating different discretization schemes. The discretization process transforms a nonlinear PDE into its linear algebraic formulation in the form of linear algebraic equations. Thus, the solution of these linear equations produces the final outcome of a device analysis. Different numerical techniques to solve these linear algebraic equations are implemented to build the framework and are discussed briefly in the next section.

2.5 Numerical Solution of Linear Algebraic Equations

The solution of linear algebraic equations, in the form of system matrix $Ax = b$, arising from the discretization of fundamental PDEs representing different transport models can be computed numerically by employing either direct methods such as Gauss Elimination method, LU Decomposition etc. or indirect methods such as Conjugate Gradient, Gauss-Seidel, Successive Over-Relaxation methods etc. [49], [50], [51]. These methods are incorporated in the proposed TCAD framework for the analysis of semiconductor devices.

TH-2607_126102004

2.6 Summary

A brief review on the vital role of TCAD in semiconductor device analysis and its working principle is presented in this chapter. Comparison of different discretization techniques which impacts accuracy and computation time of the solution of device analysis is also presented in this chapter. Different boundary conditions and formulation on basic device equations are discussed briefly in this chapter. In subsequent chapters, the proposed methodologies for TCAD analysis, designing of the basic framework, its extension to achieve better accuracy and reducing computation time are presented in detail.

Indian Institute of Technology Guwahati

TH-2607_126102004

3

Implementation of FEM and its Variants for Device Analysis

Contents

3.1 TCAD Framework using Finite-Element Discretization 32
3.2 TCAD Framework VEDA using DG-FEM 46
3.3 TCAD Framework VEDA using SUPG Stabilization Technique 51
3.4 Quantum Simulations with Schrödinger-Poisson Solver 54
3.5 Performance Analysis of Different Discretization Schemes 58
3.6 Summary 58

TH-2607_126102004

In this chapter, detailed description about indigenous TCAD framework VEDA (Very Efficient Device Analyzer) to accurately analyze semiconductor devices is presented which incorporates Finite Element Method (FEM) and its variants. Discontinuous-Galerkin FEM (DG-FEM) and Streamline upwind Petrov Galerkin FEM (SUPG-FEM) are the variants of FEM discussed in detail in this chapter to improve accuracy of the solution. In DG-FEM, basis functions employed for the discretization are discontinuous piecewise polynomials. SUPG stabilization technique exhibits an edge over the classical Scharfetter-Gummel method to solve a convection-diffusion equation. The effectiveness of the proposed methods are validated by analyzing different semiconductor devices, such as PN junction diode, MOS capacitor and MOSFET. The simulation results of these devices are compared with the commercially available TCAD simulator, Sentaurus for validating correctness of the proposed simulator.

3.1 TCAD Framework using Finite-Element Discretization

The fundamental difference between finite difference method (FDM) and FEM lies in the process of their discretization as shown in Fig. 3.1. In the discretization process of FEM whole domain or geometry of a device is divided into a large number of finite small local domains or elements using a linear basis (or shape) functions [10], [52]. The solution of the function is evaluated over these local domains to approximate the global solution. In this way, an infinite dimension problem is converted into a finite dimension problem, which can easily be represented in the form of linear algebraic equations or a system matrix. This system matrix can solved by employing a suitable direct or iterative matrix computation method.

A geometrically simple linear element can be chosen as a basis function in FEM, such as triangle, tetrahedrons, rectangular, cubes, hexahedrons, etc. One such basis function of the form tetrahedron is exhibited in Fig. 3.2, which can be employed to discretize a three-dimensional geometry. These basis functions must depend linearly on x, y, z coordinates, and are also defined in such a way that it has a defined value of "1" for node n (with coordinates as $\vec{p_n} = \{x_n, y_n, z_n\}^T$) and has "0" value for all

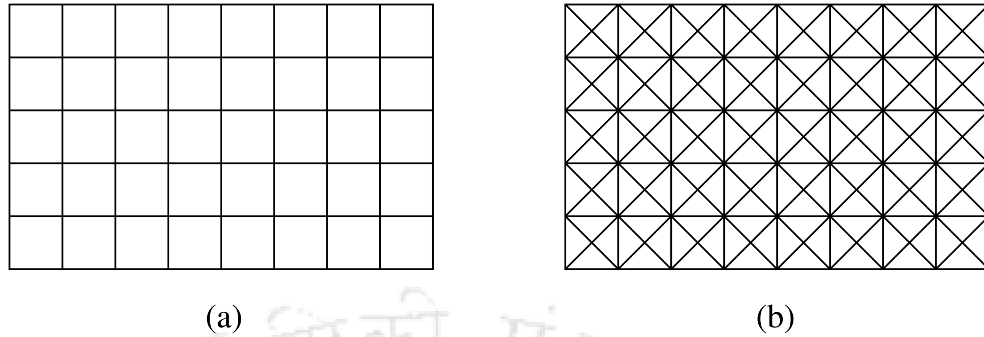


Figure 3.1: Comparison of different discretization schemes employed for (a) FDM and (b) FEM.

other mesh nodes. It can be represented as

$$N_i = \frac{1}{6V}(a_i + b_i x + c_i y + d_i z) \tag{3.1}$$

where V is the volume of tetrahedral element

$$N_i(x_j, y_j, z_j) = \begin{cases} 1 & \text{for i = j} \\ 0 & \text{for i} \neq \text{j} \end{cases} \tag{3.2}$$

These elements must span the entire geometry in such a way that there exist no empty spaces in-between without any overlap. Accuracy of approximate final solution depends on the size as well as on the number of finite elements in the discretized domain [10].

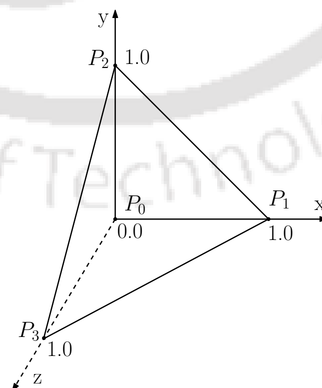


Figure 3.2: A tetrahedron shape function to be used as an element in FEM

FEM is usually implemented using a continuous Galerkin method of weighted residuals [10]. Galerkin method describes the finite-dimensional approximation of an infinite-dimensional space, in which a nonlinear continuous function is termed as *strong form* and is later converted to *weak form*. The weak

TH-2607_126102004

form can be interpreted as a function considered to be same for each basis element and the complete solution of the function is approximated by integrating it over the whole domain. The set of governing PDEs with boundary conditions is called the strong form of the problem [10]. Here, an example of second order PDE (1D Poisson's equation) is described to explain application of FEM in detail.

$$\frac{d^2u}{dx^2} = p_0 \quad - \text{Strong form} \tag{3.3}$$

$$\frac{d^2u}{dx^2} - p_0 = 0 \ - \text{Residual form} \tag{3.4}$$

$$\int_0^L \left(\frac{d^2u}{dx^2} - p_0\right) v dx = 0 \qquad - \text{Weak form} \tag{3.5}$$

The weak form is a variational statement of the problem, which is integrated against a test function.

$$\int_0^L \frac{d^2u}{dx^2}\, vdx = \int_0^L p_0\, vdx \tag{3.6}$$

Integrating above mentioned equation by parts yields

$$= -\int_0^L \frac{du}{dx}\frac{dv}{dx}dx + \left[v(L)\frac{du}{dx}\right]_{x=0}^{x=L} \tag{3.7}$$

$$= -\int_0^L \frac{du}{dx}\frac{dv}{dx}dx + v(L)\frac{du}{dx}|_{x=L} - v(0)\frac{du}{dx}|_{x=0} \tag{3.8}$$

Now problem is analyzed locally on each element. Considering finite basis function as $\{\varphi_i\}_{i=i}^{N}$, it can be mentioned that

$$u(x) = \sum_{j=1}^{N} c_j \varphi_j(x), \tag{3.9}$$

$$v(x) = \sum_{j=1}^{N} b_j \varphi_j(x), \tag{3.10}$$

Substituting equation 3.9 and equation 3.10 into equation 3.8, the following expression is obtained,

$$\int_0^L \sum_{j=1}^{N} c_j \frac{d\varphi_j}{dx}(x) \sum_{i=1}^{N} b_i \frac{d\varphi_i}{dx}(x)\, dx = \int_0^L p_0 \sum_{j=1}^{N} b_j \varphi_j(x)\, dx \tag{3.11}$$

Equation 3.11 can be rearranged to obtain the following expressions,

$$\sum_{i=1}^{N} b_i \sum_{j=1}^{N} c_j \int_0^L \frac{d\varphi GT_j}{dx}\frac{d\varphi_i}{dx}\, dx = \sum_{i=1}^{N} b_i \int_0^L p_0\varphi_i\, dx \tag{3.12}$$

$$\sum_{j=1}^{N} c_j \int_0^L \frac{d\varphi_j}{dx}\frac{d\varphi_i}{dx}\, dx = \int_0^L p_0\varphi_i\, dx \tag{3.13}$$

This can be reformulated in the form of a system of linear equations represented by $Kc = F$, where K is a symmetric matrix and c is an unknown vector. The elements of matrix K and vector F can be obtained by equations 3.14 and 3.15. The solution of $Kc = F$ can be found by employing a suitable matrix solver.

$$K_{ij} = \int_0^L \frac{d\varphi_j}{dx}\frac{d\varphi_i}{dx}\, dx \tag{3.14}$$

$$F_i = \int_0^L p_0\varphi_i\, dx \tag{3.15}$$

3.1.1 Step-by-Step FEM Formulation of a Second Order 1D PDE

In this section FEM formulation of a second order PDE is described with a simple geometry as shown in Fig. 3.3. A linear shape function is employed for the formulation to simplify the process. Total 4 nodes and 3 finite elements are chosen to showcase the example.

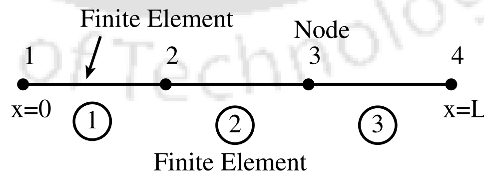


Figure 3.3: A simple 1D geometry to understand the implementation of FEM.

A 1D PDE for the example given is considered which is represented by equation 3.16 and its weak formulation can be defined by equation 3.17.

$$\frac{d}{dx}\left(k\frac{dT}{dx}\right) + S = 0 \tag{3.16}$$

TH-2607_126102004

$$\int_0^L \left[\frac{d}{dx}\left(k\frac{dT}{dx}\right) + S\right] w \, dx = 0 \tag{3.17}$$

Performing integration by parts on the above mentioned equation leads to the following mathematical expression.

$$\left[wk\frac{dT}{dx}\right]_0^L - \int_0^L \frac{dw}{dx}\, k\frac{dT}{dx}\, dx + \int_0^L Sw \, dx = 0 \tag{3.18}$$

The weak formulation presented in equation 3.17 can be expressed as summation of weak formulation over all the individual elements, which can be described as,

$$\sum_e \int_{x_i}^{x_j} \left[\frac{d}{dx}\left(k\frac{dT}{dx}\right) + S\right] w \, dx = 0 \tag{3.19}$$

Performing integration by parts of individual elements of equation 3.19 leads to the equation 3.20 given below.

$$\left[wk\frac{dT}{dx}\right]_i^j - \int_{x_i}^{x_j} \frac{dw}{dx}\, k\frac{dT}{dx}\, dx + \int_{x_i}^{x_j} Sw \, dx = 0 \tag{3.20}$$

Now, a linear trial function presented by equation 3.21 is chosen, which is valid over each individual element,

$$T = a_0 + a_1 x \quad \text{(Trial Function)} \tag{3.21}$$

$$\text{at,} \quad x = x_i,\ T = T_i \quad \text{and} \quad x = x_j,\ T = T_j$$

Therefore,

$$T_i = a_0 + a_1 x_i \tag{3.22}$$

$$T_j = a_0 + a_1 x_j \tag{3.23}$$

TH-2607_126102004

Solving equations 3.22 and 3.23 for a_0 and a_1,

$$a_1(x_j - x_i) = T_j - T_i$$
$$\therefore \quad a_1 = \frac{T_j - T_i}{x_j - x_i} \tag{3.24}$$

$$a_0 = T_i - a_1 x_i = T_i - \frac{T_j - T_i}{x_j - x_i} x_i$$
$$\therefore \quad a_0 = \frac{T_i x_j - T_j x_i}{x_j - x_i} \tag{3.25}$$

Substituting equation 3.24 and 3.25 in 3.21,

$$T = \frac{T_i x_j - T_j x_i}{x_j - x_i} + \frac{T_j - T_i}{x_j - x_i} x \tag{3.26}$$

$$T = \underbrace{\left(\frac{x_j - x}{x_j - x_i}\right)}_{N_i} T_i + \underbrace{\left(\frac{x - x_i}{x_j - x_i}\right)}_{N_j} T_j \tag{3.27}$$

As it is already discussed, the shape function or interpolation function has a peculiar feature of having value "1" at that particular node and "0" elsewhere, it can be represented in the following manner.

$$N_i = 1 \text{ at node } i, \quad \because \quad \text{when } x => x_i, \; N_i = \frac{x_j - x_i}{x_j - x_i} = 1 \tag{3.28}$$

$$N_i = 0 \text{ at node } j, \quad \because \quad \text{when } x => x_j, \; N_i = \frac{x_j - x_j}{x_j - x_i} = 0 \tag{3.29}$$

$$\therefore \quad T = N_i T_i + N_j T_j \tag{3.30}$$

$$T = \begin{bmatrix} N_i & N_j \end{bmatrix} \begin{bmatrix} T_i \\ T_j \end{bmatrix} = \begin{bmatrix} N \end{bmatrix} \begin{bmatrix} T \end{bmatrix} \tag{3.31}$$

A similar expression for the basis function, w can be formulated as follows,

$$w = \begin{bmatrix} N_i & N_j \end{bmatrix} \begin{bmatrix} w_i \\ w_j \end{bmatrix} = \begin{bmatrix} N \end{bmatrix} \begin{bmatrix} w \end{bmatrix} \tag{3.32}$$

Since w is a scalar, equation 3.32 can be represented as,

$$w^T = \begin{bmatrix} w \end{bmatrix}^T \begin{bmatrix} N \end{bmatrix}^T \tag{3.33}$$

Substituting equations 3.31 and 3.33 in the equation 3.20,

$$-\underbrace{\left[\begin{bmatrix} w \end{bmatrix}^T \begin{bmatrix} N \end{bmatrix}^T q^{''}\right]}_{Term\ 1} - \underbrace{\begin{bmatrix} w \end{bmatrix}^T \int \begin{bmatrix} \frac{dN}{dx} \end{bmatrix}^T K \begin{bmatrix} \frac{dN}{dx} \end{bmatrix} dx}_{Term\ 2} + \underbrace{\begin{bmatrix} w \end{bmatrix}^T \int S \begin{bmatrix} N \end{bmatrix}^T dx}_{Term\ 3} = 0; \tag{3.34}$$

Since $[w]^T$ is arbitrary, it is just a variation and thus can be omitted from equation 3.34. Now, each individual term can be computed one by one using the steps given below.

$$\textbf{Term 1} = \left[-\begin{bmatrix} N_i \\ N_j \end{bmatrix} q^{''} \right]_i^j = -\begin{bmatrix} 0 \\ 1 \end{bmatrix} q_j^{''} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} q_i^{''} = \begin{bmatrix} +q_i^{''} \\ -q_j^{''} \end{bmatrix} \tag{3.35}$$

$$\textbf{Term 2} = \int_i^j k \begin{bmatrix} \frac{dN_i}{dx} \\ \frac{dN_j}{dx} \end{bmatrix} \begin{bmatrix} \frac{dN_i}{dx} & \frac{dN_j}{dx} \end{bmatrix} dx = \int_i^j k \begin{bmatrix} \frac{-1}{l_e} \\ \frac{1}{l_e} \end{bmatrix} \begin{bmatrix} \frac{-1}{l_e} & \frac{1}{l_e} \end{bmatrix} dx \tag{3.36}$$

$$= \frac{k}{l_e^2} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} (x_j - x_i) = \frac{k}{l_e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{3.37}$$

$$\textbf{Term 3} = S \int_i^j \begin{bmatrix} \frac{x_j - x}{l_e} \\ \frac{x - x_i}{l_e} \end{bmatrix} dx = \begin{bmatrix} \frac{Sl_e}{2} \\ \frac{Sl_e}{2} \end{bmatrix} \tag{3.38}$$

Substituting all the terms back to equation 3.34 and rearranging them gives us the following expression, **Term 2 = Term 1 + Term 3**

$$\frac{K}{l_e} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{bmatrix} T_i \\ T_j \end{bmatrix} = \begin{bmatrix} q_i^{''} \\ -q_j^{''} \end{bmatrix} + Sl_e \begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix} \tag{3.39}$$

Later, all the terms for each individual element are assembled to form a global matrix given below.

$$\begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 1+1 & -1 & 0 \\ 0 & -1 & 1+1 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} q_1^{''} \\ -q_2^{''} + q_2^{''} \\ -q_3^{''} + q_3^{''} \\ -q_4^{''} \end{bmatrix} + Sl_e \begin{bmatrix} 1/2 \\ 1/2+1/2 \\ 1/2+1/2 \\ 1/2 \end{bmatrix} \tag{3.40}$$

Equation 3.40 exhibits the final system matrix representing linear algebraic equations arising due to the discretization process of finite element method.

3.1.2 Computational Analysis of FEM

The performance of any discretization method depends upon the amount of accuracy and its computation time. In this section computational analysis of FEM is discussed in brief. The computational complexity of FEM can be derived using the pseudocode presented in Algorithm 1. It can be seen that the final outcome of FEM depends on the number of finite elements considered for a given geometry, and the choice of basis function or its degree of freedom at each individual element. The higher the number of finite element or the degree of freedom of basis function is, the more accurate the solution would be. However, need for the more accurate solution imposes higher computation time penalty on the solution process and depends on the application.

Algorithm 1: FEM Coefficient Matrix Formulation

```
Input: E is total no. of elements in problem domain and n is number of degrees of freedom
       per element
Output: Coefficient matrix: K
for e → E, total no. of elements do
    for i → n do
        for j → n do
            K[i, j] = K[i, j] + k[i, j]; stiffness matrix
        end
    end
end
Solve: K[i, j]u_i = f_i
Matrix Size: E × E
```

A general form of the matrix formulated in the discretization process of FEM can be presented us-

TH-2607_126102004

ing equation (3.41). It can be seen that the matrix generated by this method is sparse but contains more number of non-zero elements as compared to FDM. Since the non-zeros elements signify dependency of elements over other neighboring nodes and depend on the choice of basis function, therefore, a less sparse matrix tends to give more accurate solution. Although FEM provides a more accurate solution the error introduced in the discretization process of FEM depends on two main factors; type of basis function or its degree of freedom and the discretization error while transforming a physical problem into a finite element model which can be related to modeling the boundary conditions. Therefore, the above mentioned limitations imposed by FEM provide a good scope of improvising solution accuracy.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 14 & -8 & 1 & -8 & 0 & 0 \\ 0 & -8 & 16 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 14 & -8 & 0 & -8 \\ 0 & -8 & 0 & -8 & 16 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -8 & 0 & 0 & 16 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix} \tag{3.41}$$

In this section, discretization process of a PDE using FEM is discussed in detail. In the next section implementation details of indigenous TCAD framework and solution obtained from the analysis of different semiconductor devices using our proposed framework VEDA (Very Efficient Device Analyzer) is presented.

3.1.3 Implementation Details and Simulation Results

An efficient framework, Very Efficient Device Analyzer (VEDA), to analyze semiconductor devices is developed using finite element discretization method, capable of handling 2D/3D device geometry. The implementation details of VEDA and its application for the analysis of different semiconductor devices are discussed in this section.

PDEs representing classical drift-diffusion model is incorporated in VEDA and are discretized using FEM. The whole geometry is discretized into small discrete elements to solve the PDEs locally.

TH-2607_126102004

VEDA has the flexibility to select elements of different sizes and shapes. By default, triangular elements are used to discretize a 2D geometry, whereas 3D geometry can be discretized using tetrahedral elements. This leads to the formation of a mesh-like structure throughout the geometry in such a way that no two elements overlap each other. The simulator is capable of generating both uniform and non-uniform meshes depending on the choice given. The employability of non-uniform mesh at specific places helps to achieve more accurate results. This results into minimizing computation time with respect to a uniform denser mesh. As we know that the solution of PDEs cannot be parallelized through standard procedures because of heavy dependency of neighboring nodes on each other. Therefore, VEDA is parallelized while generating system matrix or stiffness matrix using local matrices of each element as described in the previous sections. OpenMP is employed to parallelize this process and a speedup of 1.5x is achieved using Dell Precision T7610 having QuadCore processor and $16\ GB$ RAM.

During the analysis, all the parameters and quantities are normalized using their standard factors and are transformed into unit-less parameters and quantities. Table 3.1 presents various normalization factors and their corresponding values used in the analysis of semiconductor devices.

Table 3.1: List of normalization factors for the quantities of interest.

Parameter	Symbol	Normalization factor	Normalization factor Value
Position coordinate	x, y, z	Debey Length, L_D	$9.56685 \times 10^{-5} cm$
Electrostatic potential	ψ, ϕ, V_A	V_t	$0.025875V$
Electric field	E	V_t/L_D	$270.465V/cm$
Carrier densities	n, p	n_i	$1.5 \times 10^{10} cm^{-3}$
Donor densities	N_A, N_D	n_i	$1.5 \times 10^{10} cm^{-3}$

For exhibiting effectiveness of the proposed TCAD simulator VEDA, analysis results of a PN junction diode, MOS capacitor and MOSFET performed by VEDA are validated with the commercially available TCAD device analyzer, Sentaurus. The simulation results obtained for specific device parameters and geometry are given below for each device under test.

TH-2607_126102004

3.1.3.1 PN Junction Diode

For studying PN junction diode, silicon is selected as a base material. All the other related device parameters used for the analysis of a PN junction diode are listed in Table 3.2. The uniform and non-

Table 3.2: Device parameters and their values for a PN junction diode.

Device parameters	Value
Device Length: P-region, N-region	50 μm
Doping Profile: N_d, N_a	$10^{16}\ cm^{-3}$
Mobility of electrons: μ_n	1350 $cm^2/V-s$
Mobility of holes: μ_p	480 $cm^2/V-s$
Carrier lifetime: n, p	$10^{-6}s$
Temperature: T	300 K

uniform meshes created by VEDA for a PN junction diode are shown in Fig. 3.4(a) and Fig. 3.4(b), respectively. It should be noted that a dense mesh is created near the junction (depletion region) to get more accurate results. Fig. 3.5 exhibits potential profile of a diode at thermal equilibrium without

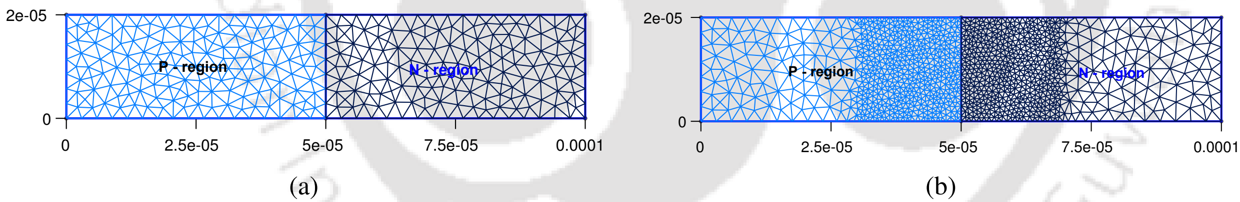


Figure 3.4: (a) Uniform mesh over PN junction diode (b) Non-uniform mesh over PN junction diode

applying any bias voltage. It can also be seen from Fig. 3.5 that the built-in potential of a diode is $\approx 0.67V$, which validates correctness of the proposed TCAD framework, VEDA. The electric field inside a diode at thermal equilibrium is shown in Fig. 3.6 and it can be seen that the electric field is present only at the junction and is zero elsewhere. The total current density for a forward bias voltage across a diode analyzed by VEDA is shown in Fig. 3.7. It can be seen that diode remains switched off (cut-off region) as long as the applied voltage is below built-in potential and it gets switched on once it crosses built-in potential barrier.

A 3D structured PN junction diode is also analyzed with the same device parameters. Non-uniform meshing using tetrahedral elements throughout the volume of the device as shown in Fig. 3.8

TH-2607_126102004

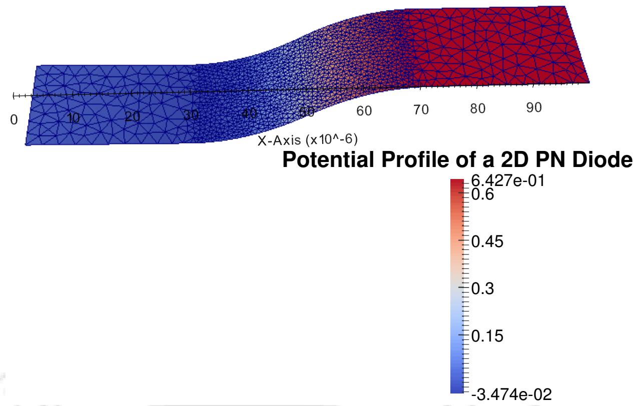


Figure 3.5: Potential profile of a 2D PN junction diode at equilibrium

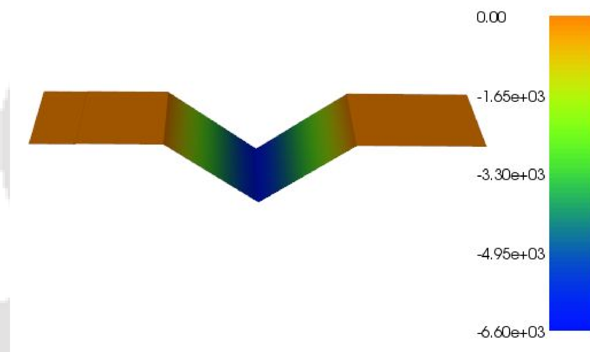


Figure 3.6: Electric filed inside a PN diode at equilibrium

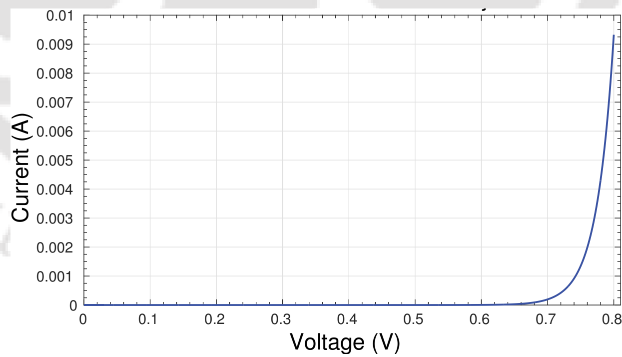


Figure 3.7: I-V characteristic of a forward bias PN junction diode

is created. The potential profile for this 3D diode is shown in Fig. 3.9.

3.1.3.2 MOS Capacitor

The device parameters used for the analysis of a 3D MOS capacitor by VEDA are listed in Table 3.3. Figure 3.10 exhibits potential profile inside a MOS capacitor when gate is supplied with positive

TH-2607_126102004

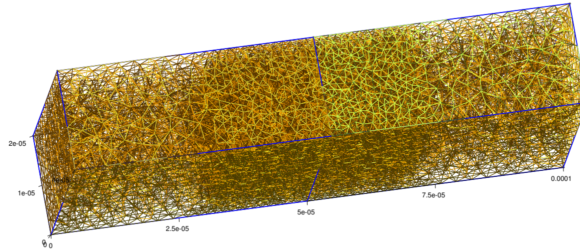


Figure 3.8: Non-uniform meshing for a 3D PN junction diode

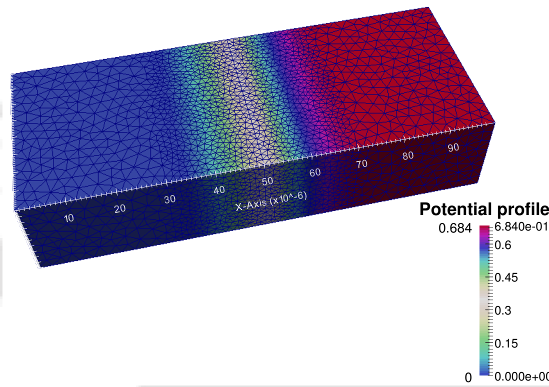


Figure 3.9: Potential profile of a 3D PN junction diode at equilibrium

voltage. The analysis of MOS capacitor also validates correctness of our proposed TCAD analyzer VEDA for 2D/3D geometries using FEM. The analysis of MOS capacitor is further extended to the analysis of MOSFET using our proposed TCAD framework VEDA. Analysis results of MOSFET are presented in the next section.

Table 3.3: Device parameters and their values for a 3D MOS capacitor.

Device parameters	Value
Device dimensions: Width, Depth	180 nm
Device dimensions: Length	250 nm
Oxide material	SiO_2
Temperature: T	300 K

3.1.3.3 MOSFET

The 3D model of an n-Channel MOSFET is developed with non-uniform meshing around the channel so that more accurate results can be obtained at specific regions of importance. Fig. 3.11 exhibits the mesh created for a 3D MOSFET, whereas Fig. 3.12 shows doping profile of an n-Channel

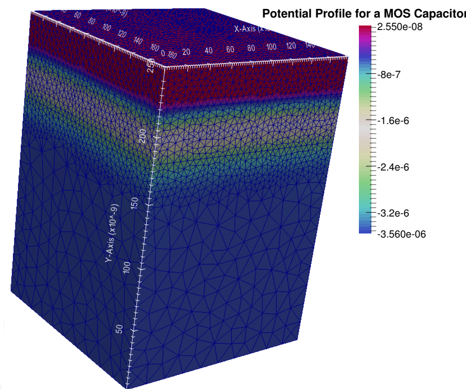


Figure 3.10: Potential profile inside a 3D MOS capacitor

MOSFET. The analysis to obtain output characteristics of a MOSFET at different gate potentials is performed by VEDA.

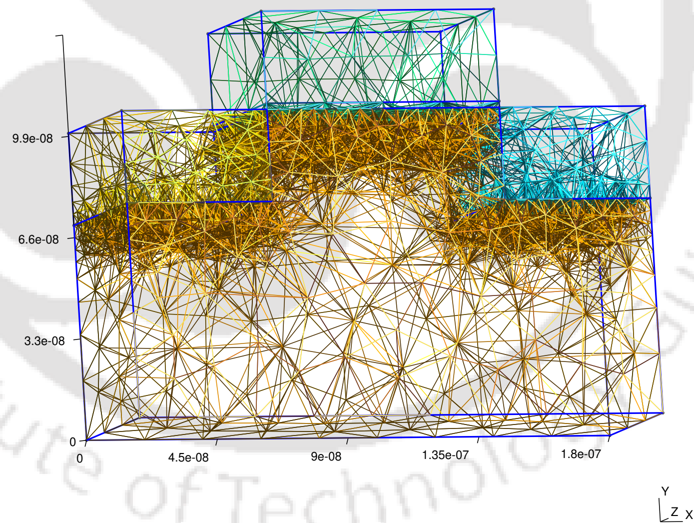


Figure 3.11: Non-uniform mesh formation inside a 3D MOSFET

Fig. 3.13 presents the comparison of output characteristic obtained using VEDA and commercially available TCAD software *Sentaurus*. Both the simulations are performed for the same set of parameters, transport model (Drift-Diffusion) and under similar assumptions for mobility, and recombination and generation. It is observed that VEDA and Sentaurus produce similar characteristics, which validates the accuracy and correctness of our proposed TCAD analyzer VEDA. In the next

TH-2607_126102004

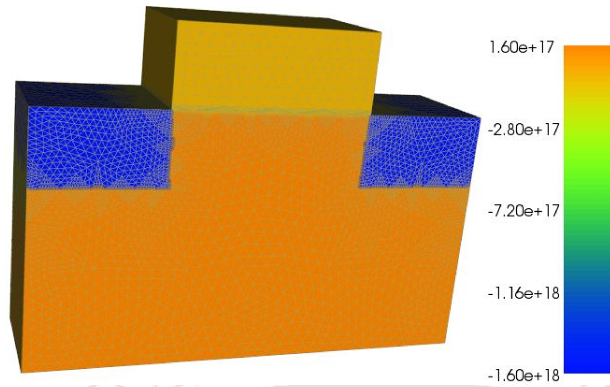


Figure 3.12: Doping profile of a 3D MOSFET

sections, detailed description of the variants of FEM are presented. These variants improve accuracy and computation time of the semiconductor device analysis significantly.

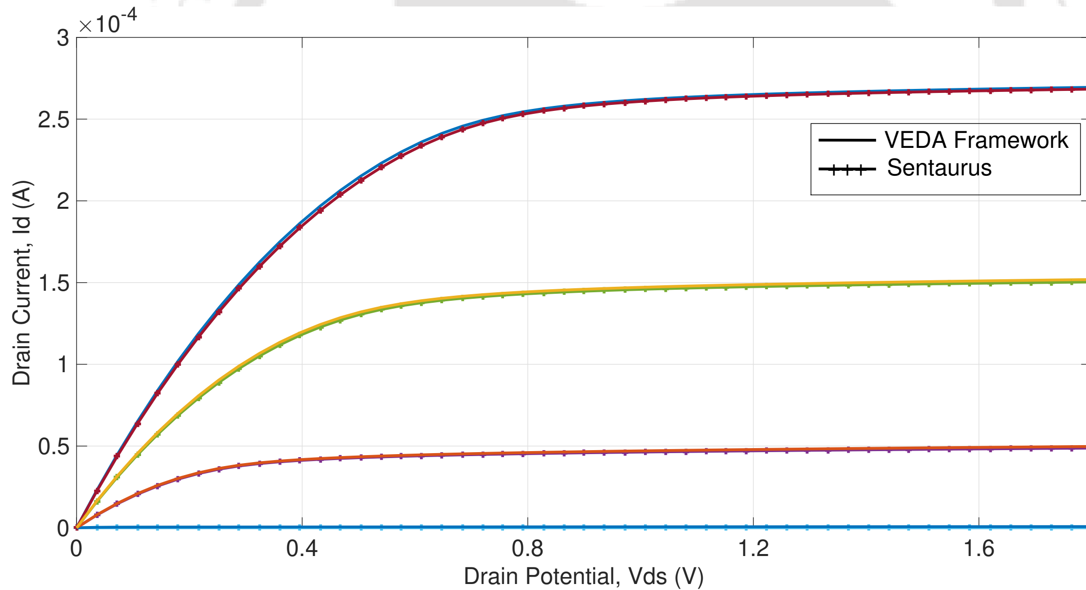


Figure 3.13: Comparison of output characteristics for a n-Channel MOSFET.

3.2 TCAD Framework VEDA using DG-FEM

Discontinuous Galerkin (DG) method is a special case of FEM, in which the basis functions employed for discretization are discontinuous piecewise polynomials. Discontinuous basis functions efficiently handle interactions between element boundaries to achieve accurate and stable solution for nonlinear hyperbolic systems. Features like high parallelizability, ease of handling complex geome-

TH-2607_126102004

tries, nonlinear stability and higher order accuracy make this discretization scheme better suitable for the analysis of semiconductor devices. The discontinuous basis function with a degree of freedom (DOF) 1 and 2 are as shown in Fig. 3.14 and Fig. 3.15, respectively. The difference in basis functions for FEM and DG-FEM is also exhibited in Fig. 3.16 and Fig. 3.17, respectively. It can be seen that the DG-FEM basis functions show discontinuity at nodes, whereas FEM basis functions are continuous in nature.

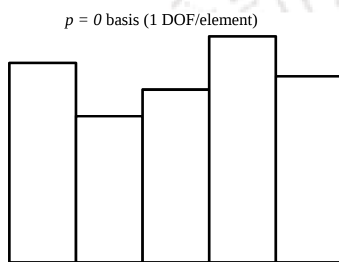


Figure 3.14: DG basis function with one degree of freedom per element.

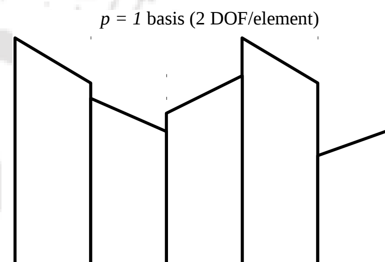


Figure 3.15: DG basis function with two degree of freedom per element.

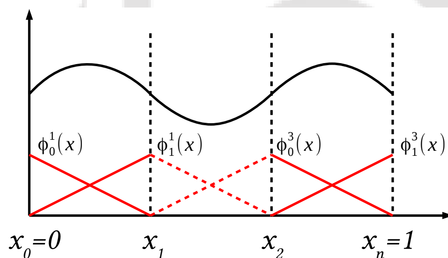


Figure 3.16: Basis function as used in Finite Element Method.

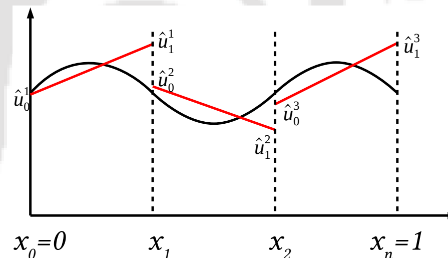


Figure 3.17: Basis function as used in DG-Finite Element Method.

Details of the implementation of DG method [53] are given below. In DG method, a generic conservation equation is chosen which is given below.

$$\frac{\partial \tilde{u}}{\partial t} + \nabla \cdot \vec{g} = 0 \tag{3.42}$$

Multiplying equation 3.42 with a basis function and integrating it over the whole domain (Ω) yields the following mathematical expression.

$$\int_{\Omega} v \frac{\partial u}{\partial t} dV + \int_{\Omega} v \nabla \cdot \vec{g} dV = 0, \ \ \forall_v \in \Phi \tag{3.43}$$

$$= \sum_e \int_e v \frac{\partial u}{\partial t} dV + \sum_e \left(- \int_e \nabla v \cdot \vec{g} dV + \oint_{\partial e} v \vec{g} \cdot \vec{n} dS \right) \tag{3.44}$$

For the efficient conservation of flux at the boundary, two different operators, *Jump* and *Average*, are defined with respect to the normal $\vec{n}^{+} = \vec{n} = -\vec{n}^{-}$ ($+ \ and \ -$ denotes outflow and inflow, respectively). Jump and Average operators are described below.

$$\begin{aligned} \text{Jump operator}: \ [v]_{x_k} &= v_{I_k}(x_k) - v_{I_{k+1}}(x_k) \\ \text{Average operator}: \ \{v\}_{x_k} &= \frac{1}{2}(v_{I_k}(x_k) + v_{I_{k+1}}(x_k)) \end{aligned} \tag{3.45}$$

Substituting equation 3.45 in the equation 3.44 results into the following expression.

$$= \sum_e \int_e v \frac{\partial u}{\partial t} dV - \sum_e \int_e \nabla v \cdot \vec{g} dV + \sum_f \int_f [v]\{\vec{g}\} + [\vec{g}]\{v\} dS \tag{3.46}$$

DG discretization is then defined as

$$\sum_e v \frac{\partial u}{\partial t} - \sum_e \int_e \nabla v \cdot \vec{g} dV + \sum_f \int_f \gamma(\tilde{u}^{+}, \tilde{u}^{-}, v^{+}, v^{-}, \vec{n}) dS = 0, \ \ \forall_v \in \Phi \tag{3.47}$$

The function parameter, γ, is required to have following properties: i) stability, ii) consistent i.e. $u^{+} = u^{-} = \tilde{u}$, and iii) conservative i.e. weight, $W = 1 \, \forall_x \in e, \ W = 0 \, \forall_x \notin e$.

The mathematical formulation of equation (3.42) using DG-FEM is given below.

$$\sum_e \int_e v \frac{\partial u}{\partial t} - \sum_e \int_e \nabla v \cdot \vec{g} dV + \sum_f \int_f \gamma(\tilde{u}^{+}, \tilde{u}^{-}, v^{+}, v^{-}, \vec{n}) dS = 0, \ \ \forall_v \in \Phi \tag{3.48}$$

The solution of equation 3.42 can be described as the summation of local solution of each individual

element shown below.

$$u = \sum_{e} u^e \tag{3.49}$$

The above mentioned formulation can be transformed to a set of linear algebraic equations in a similar manner discribed in Section 3.1.1. Implementation and simulation results of VEDA using DG-FEM is presented in the next section.

3.2.1 Implementation Details and Simulation Results

In this section, our proposed framework VEDA to analyze semiconductor devices based on DG-FEM is presented, which is capable of solving 2D device geometries incorporating five fundamental device equations resulting from Drift-Diffusion model. Discretization process of DG-FEM is similar to FEM as discussed in the previous sections. Just like FEM, DG-FEM also has the flexibility to select elements of different size and shape, capable of generating both uniform and non-uniform mesh structures. Although triangular elements have been chosen in this analysis to discretize 2D geometry, our proposed TCAD simulator VEDA supports linear, triangular and quadrilateral basis functions as well. The choice of triangular elements helped us to prove the effectiveness of the proposed methodology to analyze semiconductor devices correctly. The proposed TCAD simulator is parallelized during formulation of system matrix similar to FEM method as described in the previous sections. OpenMP directives are employed to achieve parallelization, and a speedup of $1.4x$ is obtained using Dell Precision T7610 having QuadCore processor and 16 GB RAM. Future work on the parallelizing of the solution of the system matrix is anticipated to improve the performance of our proposed TCAD analyze VEDA. The correctness of the proposed simulator VEDA is validated by comparing the solution of a PN junction diode obtained by VEDA with the solution of the same PN junction diode extracted using commercially available TCAD analyzer *Sentaurus* [2].

3.2.1.1 PN Junction Diode

Various parameters used to analyze a silicon-based PN junction diode are shown in Table 3.4. Fig. 3.18 and Fig. 3.19 show the formation of uniform and non-uniform mesh for a PN junction diode,

TH-2607_126102004

respectively. Potential profile inside a diode at thermal equilibrium is shown in Fig. 3.20. The built-in potential of a PN junction diode is found to be $\approx 0.67V$. The total current density for a forward bias voltage is as shown in Fig. 3.21. Our proposed TCAD simulator VEDA also supports complex geometries, such as FinFET and GAAFET etc., and the work can be extended for the analysis of these devices with a detailed study of their physical phenomenon. The work presented here validates the working principle of the discretization methodology DG-FEM along with its performance i.e. accuracy and computation time as compared to FEM.

Table 3.4: Various device parameters used for a PN diode.

Device Parameters	Value
Length of P-region	30 μm
Length of N-region	70 μm
Doping Profile: N_a, N_d	10^{16} cm^{-3}
Hole Mobility: μ_p	480 $cm^2/V-s$
Electron Mobility: μ_n	1350 $cm^2/V-s$
Temperature: T	300 K

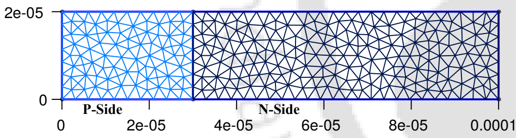


Figure 3.18: Uniform mesh over PN junction diode.

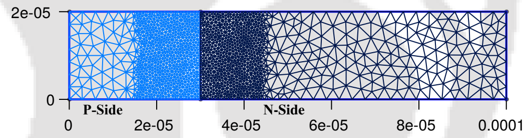


Figure 3.19: Non-uniform mesh over PN junction diode.

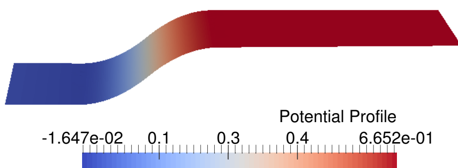


Figure 3.20: Potential profile of a 2D PN junction diode at equilibrium.

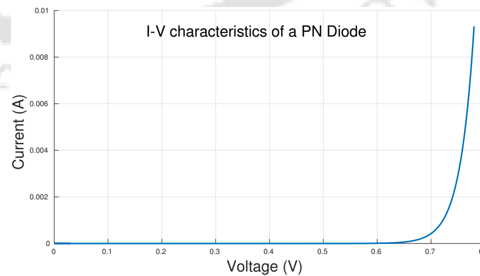


Figure 3.21: Forward biased current-voltage characteristics of a PN junction diode.

TH-2607_126102004

3.3 TCAD Framework VEDA using SUPG Stabilization Technique

Streamline upwind Petrov-Galerkin (SUPG) stabilization technique is another variant of FEM [54], [55], which is used to calculate flux and flow of charges in the semiconductor device efficiently. SUPG stabilization technique is proved to be better than the classical Scharfetter-Gummel method to solve a convection-diffusion problem. The semi-classical Drift-Diffusion model is employed to analyze semiconductor devices along with different models for carrier generation-recombination and mobility. SUPG technique enables coarse unstructured mesh to get accurate results which effectively reduces computation time due to fewer calculations as compared to FEM and DG-FEM. Analytical results for the different mesh sizes and degree of polynomials in the basis functions are presented for a PN junction diode to prove effectiveness of the proposed method in the analysis of semiconductor devices.

3.3.1 Formulation of Drift-Diffusion equation using SUPG method

The transport and continuity equations from Drift-Diffusion model is employed to understand the formulation of SUPG based discretization scheme as follows,

$$J_n = qn\mu_n E + qD_n \bigtriangledown n \tag{3.50}$$

$$\frac{\partial n}{\partial t} = \frac{1}{q} \bigtriangledown \cdot J_n + R_n \tag{3.51}$$

Considering a static analysis, equations 3.50 and 3.51 can be clubbed together to form an advection-diffusion equation which can be expressed as,

$$\frac{1}{q} \bigtriangledown \cdot (qn\mu_n E + qD_n \bigtriangledown n) = -R_n \tag{3.52}$$

$$\bigtriangledown \cdot (n\mu_n E + D_n \bigtriangledown n) = -R_n \tag{3.53}$$

Equation 3.53 can be reformulated for Function, F and residual, r and is mentioned below.

$$\text{Function, } F = \bigtriangledown \cdot (n\mu_n E) + \bigtriangledown \cdot (D_n \bigtriangledown n) = -R_n \tag{3.54}$$

$$\text{resiudual, } r = \bigtriangledown \cdot (n\mu_n E) + \bigtriangledown \cdot (D_n \bigtriangledown n) + R_n \tag{3.55}$$

TH-2607_126102004

The Function, F and residual, r expressed in the variational form to be used in the FEM formulation are given below.

$$r = \mu \cdot dot(E, \bigtriangledown n) - D \cdot \bigtriangledown(\bigtriangledown n) - R_n \tag{3.56}$$

$$F = D \cdot n \cdot dot(E, \bigtriangledown v) \cdot dx + \mu \cdot dot(\bigtriangledown v, \bigtriangledown n) \cdot dx - R_n \cdot v \cdot dx \tag{3.57}$$

$$\text{vnorm} = sqrt(dot(E, E)) \tag{3.58}$$

$$h = 2.0 \cdot \text{Circumradius}(triangle) \tag{3.59}$$

$$\text{delta} = h/(2.0 \cdot \text{vnorm}) \tag{3.60}$$

$$F = F + \text{delta} \cdot dot(E, \bigtriangledown v) \cdot r \cdot dx \tag{3.61}$$

where, $dot(a, b)$ denotes dot product of a and b; $vnorm$ is normalization; h denotes the mesh size, and in this particular analysis triangular mesh elements are used.

3.3.2 Simulation Results

As we know, SUPG technique helps to calculate flow of the charge in the device. For a given mesh, SUPG technique takes 11 iterations to converge to the final solution as compared to 19 iterations without employing SUPG. This also implies that the mesh spacing with SUPG can be made coarser to get better solution.

3.3.2.1 PN Junction Diode

The structure of PN junction employed during the analysis of DG-FEM is used for the simulation using VEDA enabled with SUPG technique. It is observed that the similar solution is produced but with less number of iteration (11 using SUPG as compared to 19 in DG-FEM). Fig. 3.22 shows the maximum error obtained during the simulation of Poisson's equation with different Lagrange finite elements ranging degree of freedom from 1-5. It is also seen that the amount of maximum error is reduced with the increase in the degree of polynomials for Lagrange elements. It is observed that the error decreases with smaller mesh sizes.

Fig. 3.23 presents the *accuracy-convergence* product (normalized error x computation time) of a

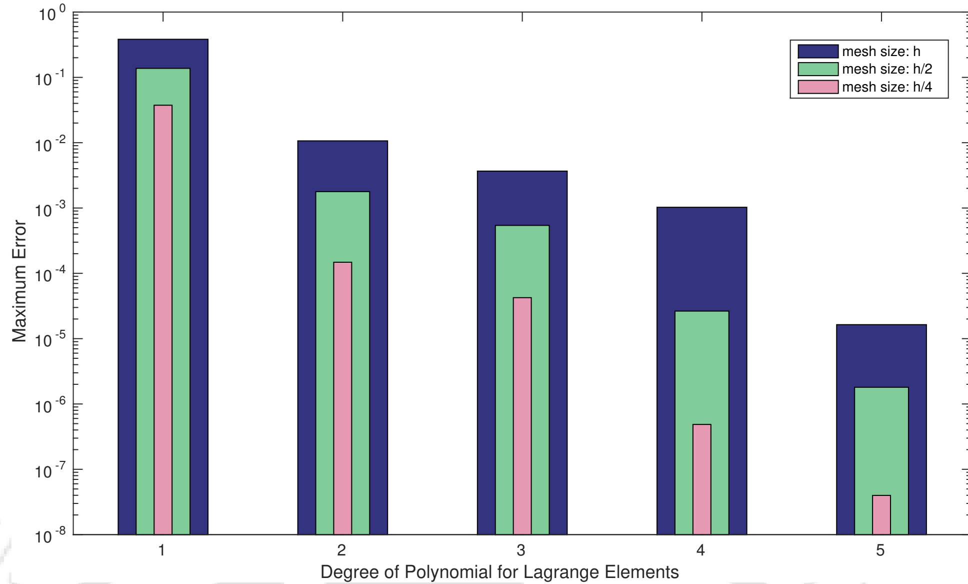


Figure 3.22: Error analysis of finite element method with different degree of Lagrange elements.

2D Poisson's equation. It can be observed that, at first, when mesh size decreases the normalized error decreases at a faster rate w.r.t. increase in computation time. With further decrease in mesh size the normalized error reaches saturation, so that the increase in computation time is more w.r.t. decrease in error.

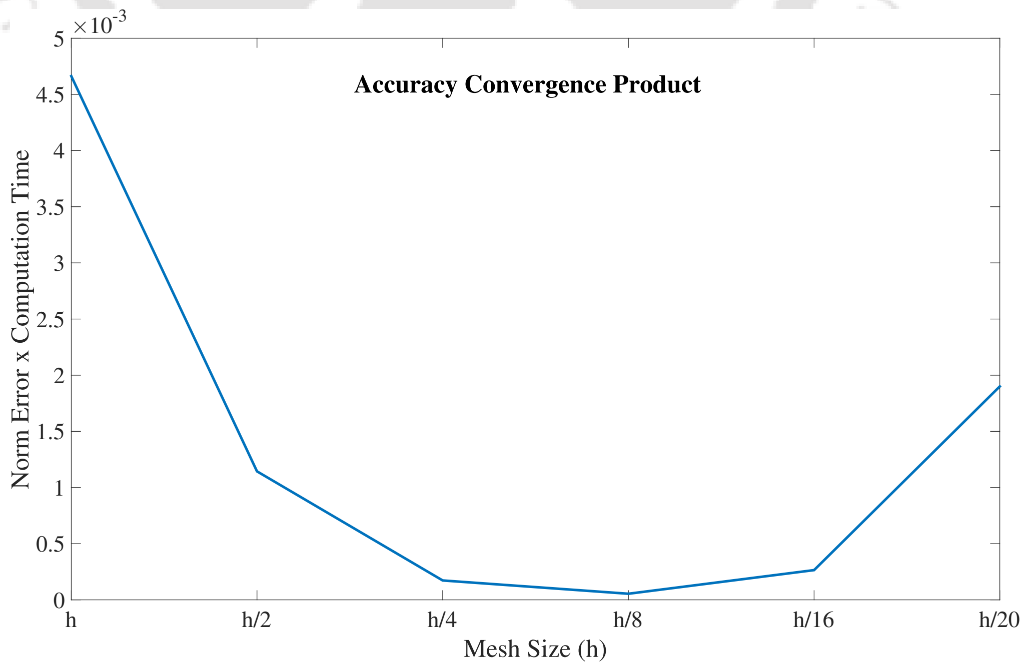


Figure 3.23: Accuracy COnvergence Product.

TH-2607_126102004

3.4 Quantum Simulations with Schrödinger-Poisson Solver

In the nano regime, analysis of semiconductor devices becomes very complex and compute intensive due to nonlinear short channel effects. Because of this, semi-classical Drift-Diffusion model fails to get desired level of accuracy in the nano regime. In order to efficiently model secondary or tertiary effects in the device at nano regime, it becomes imperative to analyze quantum transport model such as, the Schrödinger and Poisson's, in which equations are analyzed self-consistently to predict electrical behaviour of the nano scale devices. The procedure to obtain self-consistent solution of Schrödinger-Poisson's equation is presented in [56] using finite difference method. In the proposed work, similar model is analyzed using finite element method extending the capabilities of VEDA to simulate semiconductor devices in the nano regime. The model used for the analysis is presented below.

Schrödinger Equation:

$$-\frac{\hbar^2}{2}\frac{d}{dx}\left(\frac{1}{m^*(x)}\frac{d}{dx}\right)\psi(x) + V(x)\psi(x) = E\psi(x) \tag{3.62}$$

where V is the potential energy; E is the energy; $\psi(x)$ is wave function; m^* is the effective mass and h is Planck's constant.

Poisson's Equation:

$$\frac{d}{dx}\left(\epsilon_s(x)\frac{d}{dx}\right)\phi(x) = \frac{-q[n - p + N_D - N_A]}{\epsilon_0} \tag{3.63}$$

The relationship between the potential energy V and the electrostatic potential ϕ in a quantum well of random potential energy profile is given below.

$$V(x) = -q\phi(x) + \triangle E_c(x) \tag{3.64}$$

TH-2607_126102004

$\triangle E_c(x)$ is the pseudo-potential energy due to the band-offset at the interface. The relationship between wave function $\psi(x)$ and electron density $n(x)$ can be described as

$$n(x) = \sum_{m}^{k=1} \psi_k^*(x)\psi_k(x)n(k) \tag{3.65}$$

where m defines the number of bound states, and n_k is electron occupation for each state. The concentration of electron for each individual state can be determined as

$$n(k) = \frac{m^*}{\pi\hbar^2}\int_{E_k}^{\infty} \frac{1}{1+exp^{(E-E_F)/KT}} dE \tag{3.66}$$

where E_k is eigen energy for k^{th} state.

The above mentioned quantum transport model is employed to analyze a heterostructure and a MOS capacitor with dimensions in the nano regime. The simulation results corresponding to test heterostructure (GaAs/AlGaAs Heterostructure) and MOS capacitor are presented in the next section.

3.4.1 GaAs/AlGaAs Heterostructure

The device structure of GaAs/AlGaAs presented in [56] is shown in Fig. 3.24. Meshing of GaAs/AlGaAs device structure is shown in Fig. 3.25. Further, Figs. 3.26 and 3.27 showcase electron density profile and the potential profile inside the device, respectively. Figs. 3.28 and 3.29 present wavefunctions $k = 1$ for the first eigen value and $k = 2$ for the second eigen value, respectively.

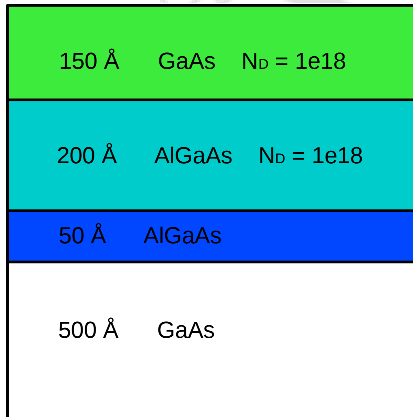


Figure 3.24: Structure of a GaAs/AlGaAs used in simulation.

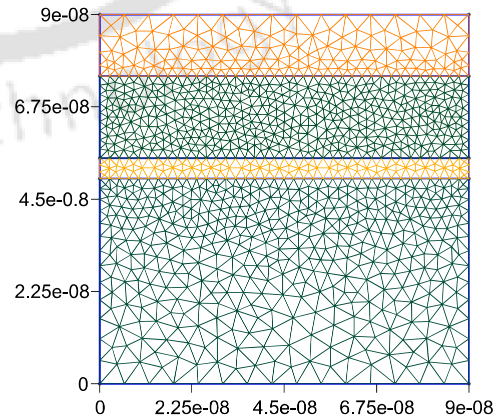


Figure 3.25: Meshing used in the simulation.

TH-2607_126102004

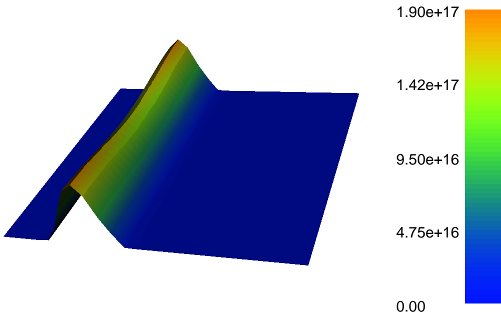


Figure 3.26: Electron density profile inside GaAs/AlGaAs heterostructure.

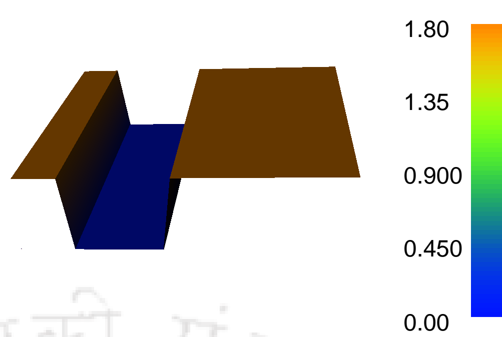


Figure 3.27: Potential profile.

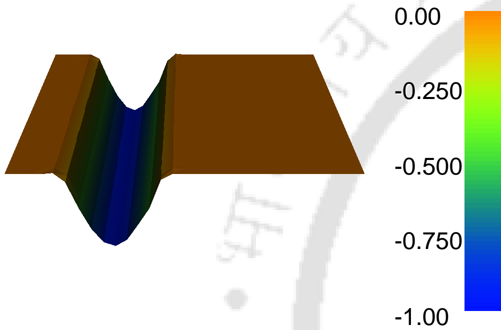


Figure 3.28: Wavefunction $k = 1$ for first eigen value.

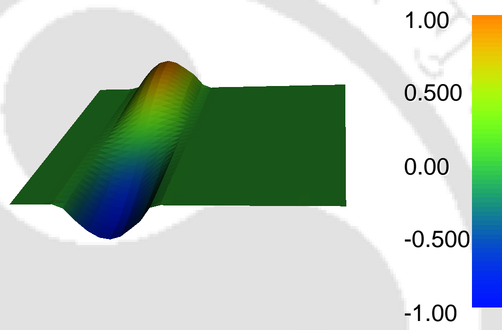


Figure 3.29: Wavefunction $k = 2$ for second eigen value.

3.4.2 MOS Capacitor

Different parameters used to analyze silicon based MOS capacitor are shown in Table 3.5. Fig. 3.30 presents geometry of MOS capacitor and non-uniform meshing employed for the analysis. Figs. 3.31 and 3.32 show CV characteristics of MOS capacitor for $3nm$ and $8nm$ oxide thickness, respectively. Figs. 3.33 and 3.34 present the inversion charge for $3nm$ and $8nm$ oxide thickness of MOS capacitor, respectively.

Table 3.5: Various device parameters used for a MOS Capacitor.

Device Parameters	Value
Length of Oxide	$3\ nm$
Length of Substrate	$50\ nm$
Doping Profile: N_a	$10^{17}\ cm^{-3}$
Hole Mobility: μ_p	$480\ cm^2/V-s$
Electron Mobility: μ_n	$1350\ cm^2/V-s$
Temperature: T	$300\ K$

TH-2607_126102004

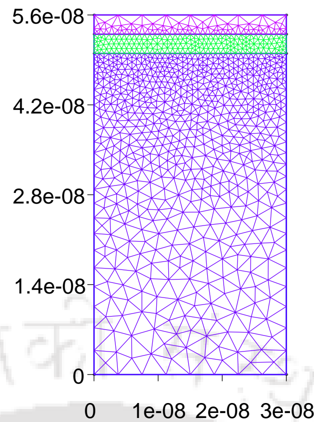


Figure 3.30: MOS-Capacitor used for simulation

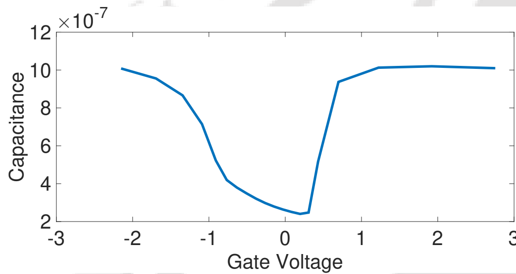


Figure 3.31: CV characteristics of a MOS with $3nm$ oxide thickness.

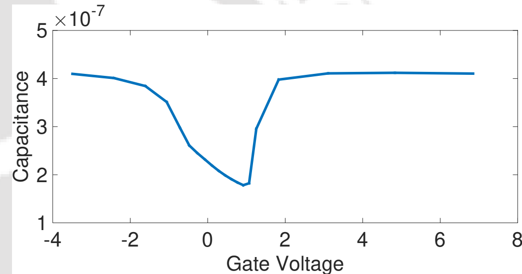


Figure 3.32: CV characteristics of a MOS with $8nm$ oxide thickness.

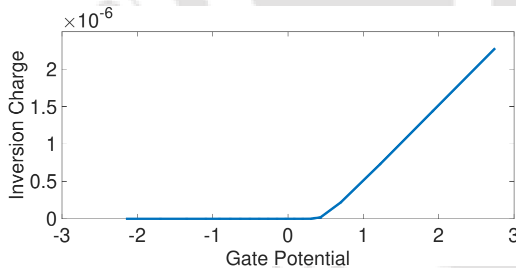


Figure 3.33: Inversion Charge for a MOS with $3nm$ oxide thickness.

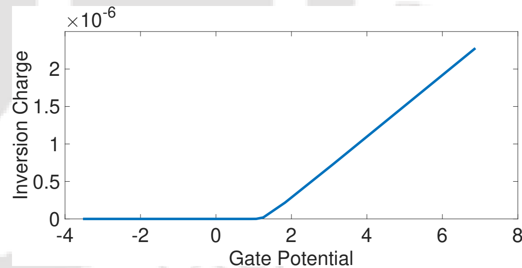


Figure 3.34: Inversion Charge for a MOS with $8nm$ oxide thickness.

The simulation results presented in this section validate the correctness of our proposed TCAD framework VEDA for the modeling of carrier transport in nano regime. The outcome of the VEDA is verified with open source analysis engines and is found to be correct. In the next section, performance analysis of all the discretization schemes introduced in this chapter is presented. The main objective of the extension of VEDA using DG-FEM and SUPG is to prepare an indigenous setup for the analysis of semiconductor devices aiming improvement in accuracy and computation time. Further extension

TH-2607_126102004

of VEDA is to improvise its capabilities in semiconductor device analysis at nano regime.

3.5 Performance Analysis of Different Discretization Schemes

An analysis of all the implemented discretization schemes is performed to compare accuracy of solution and computation time. The threshold criterion to achieve convergence is kept same for all the discretization methodologies. The only significant difference can be observed in the computation time or the number of iterations required to obtain final solution. Table 3.6 presents comparison of computation time for different discretization schemes having same geometry and threshold criterion.

Table 3.6: Comparison of computation time for different discretization schemes.

Mesh Size (# elements or nodes)	Discretization Schemes			
	FDM	FEM	DG-FEM	SUPG
100	Not Converged	Not Converged	Not Converged	0.211267
500	0.588974	0.45719	0.34817	0.337181
1000	0.75165	0.859708	0.560451	0.57438
2000	1.642170	1.364064	1.03399	0.84727
5000	6.078355	3.23607	2.5299	2.4638
10000	31.76503	10.4885	8.98808	8.67359

It can be observed from Table 3.6 that FEM with SUPG outperforms all the remaining discretization schemes with least computation time. It is also observed that for the less number of elements in the geometry, only SUPG would converge to the final solution, where as all remaining fails to achieve the same, which makes SUPG a suitable discretization methodology for the analysis of semiconductor devices.

3.6 Summary

In this chapter, implementation details of indigenously developed semiconductor device simulator VEDA is presented and its performance is compared with the currently available open-source and commercial simulators for correctness and reliability. Our proposed framework employs different discretization methodologies, such as finite element method, Discontinuous-Galerkin FEM, and FEM with SUPG stabilization technique to discretize partial differential equations. The proposed framework incorporates classical drift-diffusion model and quantum Schrödinger-Poisson model and

TH-2607_126102004

analyzes various fundamental device equations efficiently. The framework is capable of producing uniform and non-uniform mesh structures through out the geometry. The correctness, efficiency and reliability of our proposed device analyzer VEDA is validated by analyzing different semiconductor devices, such as PN junction diode for both 2D and 3D geometry, MOS capacitor, and MOSFET. Various electrical properties, such as potential profile, electric field and total current density are derived through the analysis and semiconductor devices in nano regime are also analyzed efficiently. VEDA is parallelized on a shared memory platform and a speedup of 1.5x is achieved.

Indian Institute of Technology Guwahati

TH-2607_126102004

TH-2607_126102004

4

An Accurate and Adaptive Framework based on EFG Method

Contents

4.1 Proposed Implementation of EFG Method for Device Analysis 62
4.2 Element-Free Galerkin Method 64
4.3 Application of EFG method for the Analysis of Semiconductor Devices 71
4.4 Parametric Analysis of EFG 81
4.5 Implementation Details and Results 85
4.6 Summary 88

TH-2607_126102004

This chapter presents the implementation of a highly accurate and adaptive framework based on Element-Free Galerkin Method to analyze semiconductor devices. In EFG method, a set of nodes are scattered throughout the domain to formulate algebraic equations representing carrier transport in semiconductor devices without employing mesh for domain discretization. The correlation among neighboring field nodes for mesh generation is not required apriori to formulate linear algebraic equations and thus makes this method more adaptive. The primary difference between FEM and EFG lies in the construction of their shape functions. In FEM, elements are used to construct shape functions which are identical for all the similar elements. The shape function can be chosen before beginning of analysis by predetermining its value. However, in EFG method, shape functions are constructed uniquely for each point of interest to achieve more accurate results. The solution provided by the proposed methodology is compared with other discretization techniques including the methodology adopted by commercially available TCAD simulator, *Sentaurus* [2]. Accuracy of the solution obtained by the proposed methodology is approximately $10\times$ better as compared to the solution provided by *Sentaurus*.

4.1 Proposed Implementation of EFG Method for Device Analysis

In the meshfree method, a number of nodes are distributed within the domain including its boundaries for discretization. A different sub-domain is formulated for each node to formulate linear algebraic equations from partial differential equations (PDEs) under consideration. It makes meshfree method favorable for highly accurate and adaptive analysis of PDEs and also generates favorable conditions for the parallelization of the solution. In this chapter, *Meshfree* or *Element-Free Galerkin (EFG)* method is extended for the analysis of coupled PDEs, and later it is employed to incorporate into our proposed framework, **VEDA (Very Efficient Device Analyzer)** for the analysis of semiconductor devices. For validating the effectiveness of the proposed framework, the semi-classical transport model (Drift-Diffusion) is incorporated for the analysis of semiconductor devices. Drift-Diffusion (DD) model consists of five fundamental equations describing two vital phenomena; i) transport mechanism governing the flow of charges carriers, ii) electric field based on static charges.

TH-2607_126102004

Both these phenomena are strongly coupled to each other, and therefore, it is imperative to analyze model equations self-consistently. The results obtained through **VEDA** validates the effectiveness of the applicability of meshfree method in the analysis of semiconductor devices.

For the solution of PDEs, many well-known methods, such as FDM, FEM, FVM, and MWS have been developed in the past. All of these methods depend on mesh generation for discretization, which makes them susceptible to various numerical errors. These issues are addressed by employing Element-free Galerkin method to formulate a system of linear algebraic equations. A comparative study of EFG and other discretization schemes is already presented in Table 2.1 to venture its dominance using four vital parameters termed as mesh generation, adaptability, range of problems and accuracy [43], [44]. Due to the absence of finite elements in the discretization, it is called the element free method. It is to be noted that EFG and meshfree are used interchangeably in this text. Since EFG method is said to be the more evolved version of FEM, it becomes imperative to discuss their distinguish features to validate the efficiency of EFG over FEM. Fig. 4.1 presents the different discretization schemes for FEM and EFG method. It can be observe that FEM employs dividing the whole geometry into number of finite small elements, where as EFG make use of scattered nodes to cover the whole domain. The main advantage of EFG comes from the deployment of these scattered nodes only, where correlation between neighboring nodes are defined by local support domain, which is discussed briefly in the subsequent section. This process flow of discretization for the analysis can be best understood using Fig. 4.2 comparing both FEM and EFG methods.

Apart from the differences, both these schemes works on the weak form of a PDE to formulate the discretization. Weak-form PDEs requires a weaker consistency of the approximate function. Discretization of weak form PDEs can produce a stable set of equations to achieve better solution accuracy. The central principle behind the creation of weak-form is the weighted residual method which is more general and powerful mathematical tool developed to discretize PDEs. Different methods are reported in the literature based on the strong and weak forms of partial differential equations, and have already been discussed in Chapter 2, Fig. 2.7.

TH-2607_126102004

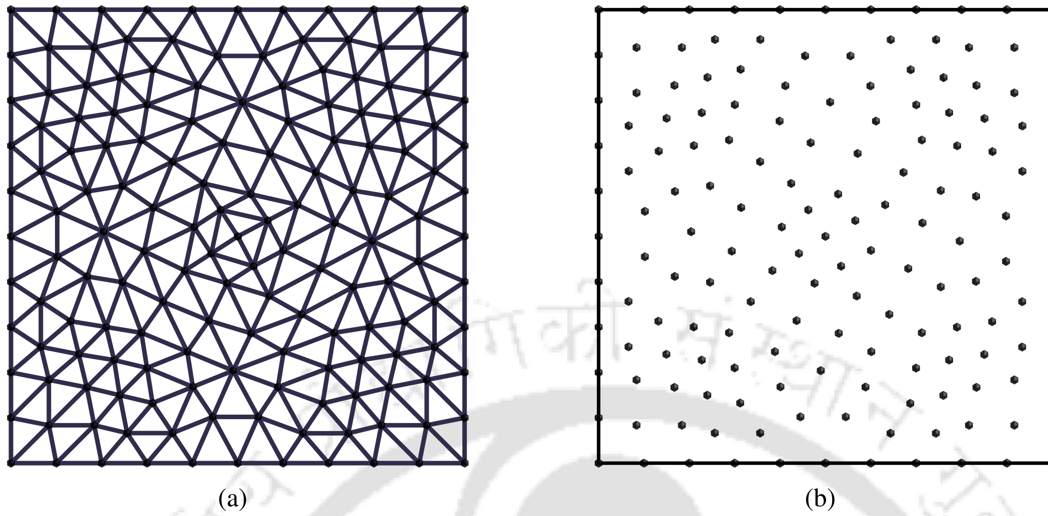


Figure 4.1: Different discretization schemes employed for (a) FEM and (b) Element Free Method

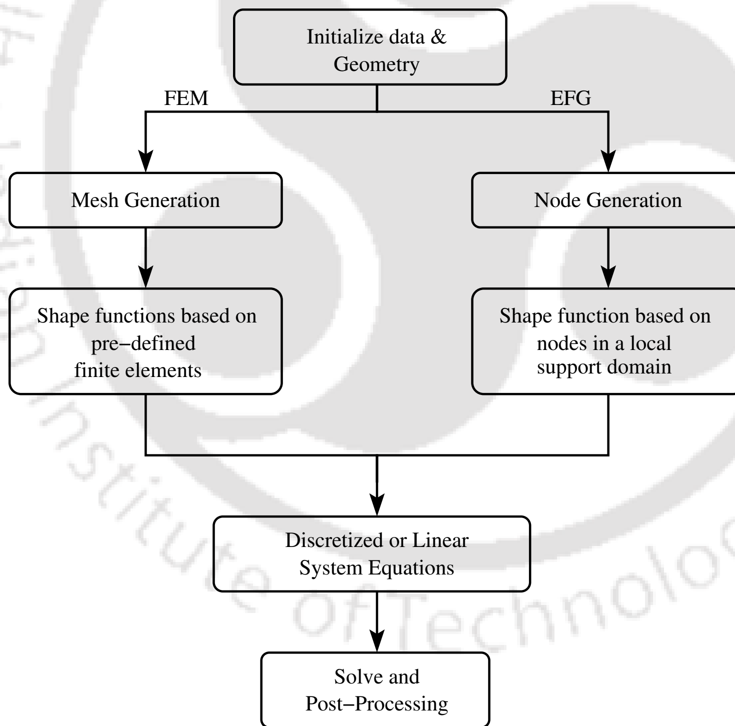


Figure 4.2: Flowchart of EFG and FEM implementation

4.2 Element-Free Galerkin Method

The Element-Free Galerkin (EFG) is a meshfree or meshless method in which a finite number of nodes are scattered within the problem domain and on its boundary to generate a system of discrete

TH-2607_126102004

equations corresponding to PDEs. These equations are used to evaluate unknown field function of PDEs. EFG employs Moving Least Square (MLS) method to approximate unknown field function. MLS method is composed of three main constituents, a weight function associated with each node, a basis function or the polynomial and a set of coefficients depending on the position. The weight function as mentioned above is non-zero over a small domain around a node called its support domain, and zero elsewhere. This support domain of a node also signifies its *domain of influence*, the domain over which it contributes to the approximation. The connectivity between two particular nodes is defined by the overlap of their respective domain of influence. Fig. 4.3 shows local support domain around the point of interest for a given problem domain.

The distinguishing feature of EFG is the construction of its shape functions. In FEM, elements are used to construct shape functions which are identical throughout the geometry. The choice of shape function can be made before the beginning of analysis by predetermining its value. However, in the EFG method, shape functions are constructed uniquely at each point of interest to achieve more accurate results.

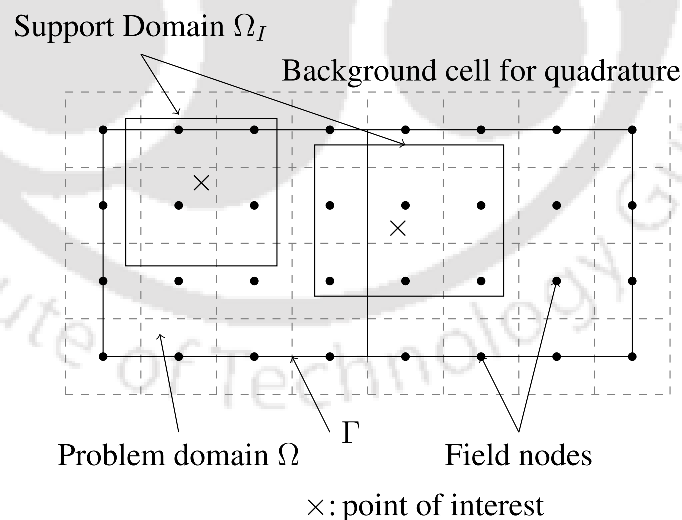


Figure 4.3: Local support domains used in the MFree method to construct shape functions.

The basic fundamentals required to formulate EFG method are described below.

(i) *Problem Domain* is a geometric area which is considered for the analysis.

(ii) *Field Nodes* are the set of nodes scattered within the problem domain and its boundaries. These

TH-2607_126102004

nodes depend only on the local support domain (explained below) for interpolation or approximation. The correlation among neighboring field nodes for mesh generation is not required apriori to formulate linear algebraic equations. The field nodes carry information of field variables or functions in an EFG formulation to generate a meshfree environment.

(iii) *Local Support Domain* at any particular point of interest signifies the total number of nodes contributing to approximate function at that point. The shape or dimension of local support domain can be different at different points depending on the desired accuracy.

(iv) *Weight Function* represents the weak or variational form of a problem. As we know, the quality of the solution depends on the selection of weight function. The approach is called Galerkin's method when the chosen weight function is similar to the shape function.

(v) The formulation of *Shape Function* is an integral part for the approximation of field variables. Shape functions in EFG are different for different nodes and are formulated independently for each node. Although this is a compute-intensive process, it is compensated by the increase in accuracy due to locally defined shape functions. A shape function in EFG can be constructed by using various methods, such as Moving Least Square (MLS), and Radial Point Interpolation method (RPIM). In this chapter, EFG with MLS and RPIM methods is employed to develop a framework to analyze semiconductor devices. It is observed that MLS exhibits better accuracy as compared to RPIM. Detailed results are presented in Section 4.5 supporting the same.

The computational complexity of EFG can be estimated by the formulation and size of its stiffness matrix, which is expressed in the form of pseudocode in Algorithm 2. An example matrix formulated in the discretization process of EFG is presented in equation (4.1). It can be seen that the matrix generated by this method is more dense as compared to the stiffness matrices generated by FDM and FEM. This matrix can be analyzed by using any state-of-the-art sparse matrix solver taking more computational time as compared to FDM and FEM. It can be interpreted from the previous discussion that the accuracy of the solution depends on the quality of the stiffness matrix, and EFG outperforms both FDM and FEM in this regard.

TH-2607_126102004

Algorithm 2: EFG Coefficient Matrix Formulation

Input: Total no. of nodes in problem domain and local domain: n and m
Output: Coefficient matrix: K

for $i \rightarrow n$, *total no. of nodes* **do**
  for $j \rightarrow m$, *no. of nodes in sub-domain* **do**
    for $k \rightarrow m_k$ **do**
      for $l \rightarrow m_l$ **do**
        $W[k,l] = W[k,l] + w[k,l]$; weight function
        $\Phi_{I,x} = \mathrm{p}_{,x}^T \mathrm{A}^{-1}\mathrm{B}_I + \mathrm{p}^T(\mathrm{A}_{,x}^{-1}\mathrm{B}_I + \mathrm{A}^{-1}\mathrm{B}_{I,x})$; shape function
      end
    end
    $K[k,l] = K[k,l] + k[k,l]$; stiffness matrix
  end
end
Matrix Size: $N \times N$, where $N = 2 \times n$

$$\begin{bmatrix} 7.7356 & -5.4998 & -2.1535 & -0.0823 & 0 & 0 & 0 \\ -5.4998 & 8.7523 & -1.1949 & -1.9753 & -0.0823 & 0 & 0 \\ -2.1535 & -1.1949 & 6.7023 & -1.2346 & -2.037 & -0.0823 & 0 \\ -0.0823 & -1.9753 & -1.2346 & 6.5844 & -1.2346 & -1.9753 & -0.0823 \\ 0 & -0.0823 & -2.037 & -1.2346 & 6.7023 & -1.1949 & -2.1535 \\ 0 & 0 & -0.0823 & -1.9753 & -1.1949 & 15.419 & -12.166 \\ 0 & 0 & 0 & -0.0823 & -2.1535 & -12.166 & 14.402 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix} \quad (4.1)$$

The need of the hour in the current scenario is to employ a method which not only can improve accuracy of the solution but also provides sufficient base for the parallelization of the solution of linear algebraic equations arising from the analysis of semiconductor devices. EFG naturally qualifies as a strong candidate to be employed in the solution of novel semiconductor devices. The incorporation of EFG in the TCAD simulator, VEDA, is an endeavour to provide scientific community an efficient platform to study novel device structures.

TH-2607_126102004

4.2.1 MLS Approximation

Moving least square approximation is a technique used to formulate shape functions for curve fitting. One of the most significant properties of MLS approximation is that its continuity is directly proportional to the continuity of weight function. This property enables the approximate function to inherit continuity of the weight function, even if the continuity of MLS basis function is different from the continuity of weight function. Therefore, a basis function with lower order polynomial, such as a linear function, can be utilized to achieve better convergence and more accurate results. The MLS approximation is explained below in detail for completeness.

Let $\psi(\overline{x})$ be the function of field variable over domain Ω and its approximation is described by

$$\psi(\overline{x}) = \sum_{i=0}^{m} p_i(\overline{x}) a_i(\overline{x}) = p^T(\overline{x}) a(\overline{x}) \tag{4.2}$$

where $p_i(\overline{x})$ is a given monomial in the polynomial basis function in coordinate space $\overline{x}^T = \begin{bmatrix} x & y \end{bmatrix}$ and m is the number of monomials in the basis function. It is to be noted that $\psi(\overline{x})$ is a vector and can be expressed as $\psi^T = \begin{bmatrix} \psi_1 & \psi_2 & \cdots & \psi_n \end{bmatrix}$. In equation (4.2), basis function vector $p(\overline{x})$ and the coefficient vector $a(\overline{x})$ are defined as, respectively,

$$p^T(\overline{x}) = \begin{bmatrix} p_0(\overline{x}) & p_1(\overline{x}) & \cdots & p_{m-1}(\overline{x}) & p_m(\overline{x}) \end{bmatrix} \tag{4.3}$$

$$a^T(\overline{x}) = \begin{bmatrix} a_0(\overline{x}) & a_1(\overline{x}) & \cdots & a_{m-1}(\overline{x}) & a_m(\overline{x}) \end{bmatrix} \tag{4.4}$$

In two dimensional (2D) space, linear basis functions is expressed as

$$p^T(\overline{x}) = \begin{bmatrix} 1 & x & y \end{bmatrix} \quad and \quad m = 3 \tag{4.5}$$

It is assumed that the local support domain of $\overline{x}$ contains n local nodes $(\overline{x}_I)$ around its neighborhood. Approximate values of field variables calculated as the contribution of $\overline{x}_I$ in the local support domain can be described as

$$\psi_L^h(\overline{x}, \overline{x}_I) = \sum_{j=0}^{m} p_j(\overline{x}_I) a_j(\overline{x}) = p^T(\overline{x}_I) a(\overline{x}) \tag{4.6}$$

Equation (4.2) can be analyzed to estimate $a(\overline{x})$ using weighted residual method [44] by minimiz-

TH-2607_126102004

ing the following weighted least square formulation,

$$J = \sum_{I=1}^{n} w(\overline{x} - \overline{x}_I)[p^T(\overline{x}_I)a(\overline{x}) - \psi(\overline{x}_I)]^2 \tag{4.7}$$

where $w(\overline{x} - \overline{x}_I)$ is the weight function. The significance of weight function is exhibited when nodes in the local support domain of $\overline{x}$ are more than monomials in the basis function $p^T(\overline{x})$. The minimization of J with respect to $a(\mathbf{x})$ yields a set of linear algebraic equations governed by

$$A(\overline{x})a(\overline{x}) = B(\overline{x})\psi(\overline{x}) \tag{4.8}$$

$$a(\overline{x}) = A^{-1}(\overline{x})B(\overline{x})\psi(\overline{x}) \tag{4.9}$$

where $A(\overline{x})$ and $B(\overline{x})$ are, respectively,

$$\begin{aligned} A(\overline{x}) &= \sum_{I=1}^{n} w(\overline{x} - \overline{x}_I)p(\overline{x}_I)p^T(\overline{x}_I) \\ &= w(\overline{x} - \overline{x}_1)\begin{bmatrix} 1 & x_1 & y_1 \\ x_1 & x_1^2 & x_1y_1 \\ y_1 & x_1y_1 & y_1^2 \end{bmatrix} + \\ &\quad w(\overline{x} - \overline{x}_2)\begin{bmatrix} 1 & x_2 & y_2 \\ x_2 & x_2^2 & x_2y_2 \\ y_2 & x_2y_2 & y_2^2 \end{bmatrix} + \\ &\quad \cdots + w(\overline{x} - \overline{x}_n)\begin{bmatrix} 1 & x_n & y_n \\ x_n & x_n^2 & x_ny_n \\ y_n & x_ny_n & y_n^2 \end{bmatrix} \end{aligned} \tag{4.10}$$

The similar expression can be obtained for $B(\overline{x})$. Using equations (4.9) and (4.2), $\psi(\overline{x})$ formulation

TH-2607_126102004

is expressed as

$$\psi(\overline{x}) = \sum_{I=1}^{n}\sum_{i=0}^{m} p_i(\overline{x})(A^{-1}(\overline{x})B(\overline{x}))_{iI}\psi_I$$
$$= \sum_{I=1}^{n}\Phi_I(\overline{x})\psi_I = \Phi(\overline{x})\psi \tag{4.11}$$

where $\Phi(\overline{x})$ is shape function vector and $\phi_I(\overline{x})$ is the shape function at the Ith node, which is defined as

$$\Phi_I(\overline{x}) = \sum_{i=0}^{m} p_i(\overline{x})(A^{-1}(\overline{x})B(\overline{x}))_{iI} = p^T A^{-1} B_I \tag{4.12}$$

The spatial derivative of the shape function is presented by

$$\Phi_{I,x} = \sum_{i=0}^{m}[p_{i,x}(A^{-1}B)_{iI} + p_i(A^{-1}_{,x}B + A^{-1}B_{,x})_{iI}]$$
$$= (p^T A^{-1} B_I)_{,x}$$
$$= p^T_{,x} A^{-1} B_I + p^T(A^{-1}_{,x} B_I + A^{-1} B_{I,x}) \tag{4.13}$$

The similar expression can be obtained for $\Phi_{I,y}$. Here $A^{-1}_{,i}$ and $B_{I,i}$ are evaluated using the following expressions given below.

$$A^{-1}_{,i} = -A^{-1} A_{,x} A^{-1} \tag{4.14}$$
$$B_{I,i}(\overline{x}) = \frac{dw}{di}(x - \overline{x})p(\overline{x}_I) \tag{4.15}$$

where the index succeeding comma is a spatial derivative.

4.2.2 The Weight Function

The most important part of EFG formulation is the selection of weight function because it directly affects the accuracy of $\Psi^h(x)$. Weight function is nonzero only within a small region of $\overline{\mathbf{x}}_I$, which is also known as its *domain of influence*. In this chapter, we have employed cubic spline weight function as a function of normalized radius, r described in equation 4.16.

TH-2607_126102004

$$w(r) = \begin{cases} \frac{2}{3} - 4r^2 + 4r^3 & \text{for} \quad r \leq 0.5 \\ \frac{4}{3} - 4r + 4r^2 - \frac{4}{3}r^3 & \text{for} \quad 0.5 < r \leq 1 \\ 0 & \text{for} \quad r > 1 \end{cases} \tag{4.16}$$

where $r = \frac{d_I}{d_{mI}}$. In this equation, d_{mI} is the size of domain of influence of Ith node and $d_I = \|x - x_I\|$. Further, d_{mI} is also defined as

$$d_{mI} = d_{max} c_I \tag{4.17}$$

where d_{max} is the scaling parameter ranging from $2.0 - 4.0$. In this chapter, d_{max} is considered to be 3.5 for the analysis of semiconductor devices. c_I is the maximum distance from I^{th} node to the neighboring nodes in its domain of influence. For estimating c_I, it is mandatory to have at least two nodes in the domain of influence of Ith node. The spatial derivative of weight function at I^{th} node is determined using an equation given below.

$$\frac{dw_I}{dx} = \frac{dw_I}{dr}\frac{dr}{dx} \tag{4.18}$$

Using equation (4.16), equation (4.18) is further simplified as

$$\frac{dw_I}{dx} = \begin{cases} (-8r + 12r^2) sign(x - x_I) & \text{for} \quad r \leq 0.5 \\ (-4 + 8r - 4r^2) sign(x - x_I) & \text{for} \quad 0.5 < r \leq 1 \\ 0 & \text{for} \quad r > 1 \end{cases} \tag{4.19}$$

The use of MLS approximation and weight function in the formulation of EFG method for semiconductor devices is presented in the next section.

4.3 Application of EFG method for the Analysis of Semiconductor Devices

This chapter presents application of EFG method to develop a framework for the accurate analysis of semiconductor devices. In this section, implementation of EFG method to analyze *Drift-Diffusion*

TH-2607_126102004

model represented in the form of partial differential equations is described in detail. The five fundamental equations of the Drift-Diffusion model can be categorized as

(i) *Poisson equation*

$$\nabla \cdot (-\epsilon \nabla \psi) = \rho, \tag{4.20}$$

where $\rho = q[p - n + N_d - N_a]$,

(ii) *Transport equations*

$$J_n = qn\mu_n E + qD_n \nabla n \tag{4.21}$$

$$J_p = qp\mu_p E - qD_p \nabla p, \tag{4.22}$$

(iii) *Continuity equations*

$$\frac{\partial n}{\partial t} = \frac{1}{q}\nabla \cdot J_n + U_n, \frac{\partial p}{\partial t} = -\frac{1}{q}\nabla \cdot J_p + U_p, \tag{4.23}$$

where μ and D are mobility and diffusion coefficient, respectively; U is net generation and recombination rate; ψ and E are electrostatic potential and electric field, respectively; n, p are electrons and holes concentration, respectively; J, and q are current density and elementary charge, respectively; N_d and N_a are donar and acceptor concentration, respectively.

A general two-dimensional second order differential equation with problem domain Ω and boundary Γ is represented as

$$\nabla^2 V = -f \tag{4.24}$$

where, ∇^2 is the laplacian operator given by

$$\nabla^2 = \frac{d^2}{dx^2} + \frac{d^2}{dy^2} \tag{4.25}$$

TH-2607_126102004

Equation (4.24) can be expressed in the matrix form as

$$\begin{bmatrix} \frac{d^2}{dx^2}+\frac{d^2}{dy^2} & 0 \\ 0 & \frac{d^2}{dy^2}+\frac{d^2}{dx^2} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} + \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = 0 \tag{4.26}$$

$$\begin{bmatrix} \frac{d^2}{dx^2} & 0 \\ 0 & \frac{d^2}{dy^2} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} + \begin{bmatrix} \frac{d^2}{dy^2} & 0 \\ 0 & \frac{d^2}{dx^2} \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} + \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = 0 \tag{4.27}$$

$$L_1^T L_1 V + L_2^T L_2 V + F = 0 \tag{4.28}$$

$$LV + F = 0 \quad in \quad \Omega \tag{4.29}$$

$$(\vec{E}_1 + \vec{E}_2)\cdot\hat{n} = \overline{\tau} \quad on \quad \Gamma_n \tag{4.30}$$

$$V = \overline{v} \quad on \quad \Gamma_e \tag{4.31}$$

where Γ_n and Γ_e denotes the natural (Neumann) and essential (Dirichlet) boundaries of problem domain Ω. $\hat{n}$ is the unit normal vector perpendicular at a point on the natural boundary (Γ_n). L_1 and L_2 are the matrix differential operators which can be represented as

$$L_1 = \begin{bmatrix} \frac{d}{dx} & 0 \\ 0 & \frac{d}{dy} \end{bmatrix} \quad and \quad L_2 = \begin{bmatrix} \frac{d}{dy} & 0 \\ 0 & \frac{d}{dx} \end{bmatrix} \tag{4.32}$$

The electric field distribution vector $\vec{E}$, scalar field vector V and input F are denoted as,

$$\vec{E}_1 = -L_1 V_1 \quad and \quad \vec{E}_2 = -L_2 V_2 \tag{4.33}$$

$$F^T = \begin{bmatrix} f_1 & f_2 \end{bmatrix} \tag{4.34}$$

$$V^T = \begin{bmatrix} V_1 & V_2 \end{bmatrix} \tag{4.35}$$

TH-2607_126102004

4.3.1 Discretization of Poisson Equation

In this section, 2D Poisson's equation given in equation 4.20 is discretization using EFG method. The equation 4.20 can be reformulated as

$$\nabla \cdot (\epsilon \nabla \psi) + \rho = 0, \;\; or \;\; \nabla \cdot (\nabla \psi) + \theta = 0 \tag{4.36}$$

where $\theta = \rho/\epsilon$.

In order to obtain discretized system of equations using EFG method, differential equations are first converted to their integral forms by employing weighted residual method. The integral formulation of a differential equation improves stability and accuracy of the solution by removing possible errors caused by function approximations. The integral form of the equation 4.36 can be represented as

$$\int_{\Omega} \delta\psi^T \nabla \cdot (\nabla \psi) \mathrm{d}\Omega + \int_{\Omega} \delta\psi^T \theta \mathrm{d}\Omega = 0 \tag{4.37}$$

where ψ and $\delta\psi$ are the shape and test function, respectively. The Galerkin weak-form for the equation 4.37 can be described as

$$\begin{aligned}\int_{\Omega} (L_1 \delta\psi)^T \, (L_1 \psi) \mathrm{d}\Omega + \int_{\Omega} (L_2 \delta\psi)^T \, (L_2 \psi) \mathrm{d}\Omega \\ - \int_{\Omega} (\delta\psi)^T \theta \mathrm{d}\Omega - \int_{\Gamma_n} (\delta\psi)^T \overline{\tau} \mathrm{d}\Gamma \\ - \int_{\Gamma_e} (\delta\lambda)^T (\psi - \overline{\psi}) \mathrm{d}\Gamma - \int_{\Gamma_e} (\delta\psi)^T \lambda \mathrm{d}\Gamma = 0\end{aligned} \tag{4.38}$$

where λ is Lagrange multiplier used to enforce essential boundary conditions in the EFG method. In order to analyze equation 4.38, the problem domain Ω is discretized into field nodes for the approximation of electrostatic potential in Poisson's equation. For the purpose of simplicity, equation 4.38 can be sub-divided into following terms.

$$I_1 = \int_{\Omega} (L_1 \delta\psi)^T \, (L_1 \psi) \mathrm{d}\Omega + \int_{\Omega} (L_2 \delta\psi)^T \, (L_2 \psi) \mathrm{d}\Omega$$

$$I_2 = \int_{\Omega} (\delta\psi)^T \theta \mathrm{d}\Omega$$

$$I_3 = \int_{\Gamma_n} (\delta\psi)^T \overline{\tau} \mathrm{d}\Gamma$$

TH-2607_126102004

$$I_4 = \int_{\Gamma_e} (\delta\lambda)^T(\psi - \overline{\psi})\mathrm{d}\Gamma$$

$$I_5 = \int_{\Gamma_e} (\delta\psi)^T\lambda\mathrm{d}\Gamma$$

The MLS shape function is employed for the approximation of the potential at the field nodes within its local support domain. The approximate potential for a local support domain containing n points is

$$\delta\psi^h_{2\text{x}1} = \begin{bmatrix} \delta\psi_x \\ \delta\psi_y \end{bmatrix} = \begin{bmatrix} \Phi_1 & 0 & \cdots & \Phi_n & 0 \\ 0 & \Phi_1 & \cdots & 0 & \Phi_n \end{bmatrix} \begin{bmatrix} \delta\psi_{1x} \\ \delta\psi_{1y} \\ \vdots \\ \delta\psi_{nx} \\ \delta\psi_{ny} \end{bmatrix}$$
$$= \Phi_{(2\text{x}2\text{n})}\delta\psi_{(2\text{nx}1)} = \sum_{I=1}^{n} \begin{bmatrix} \Phi_I & 0 \\ 0 & \Phi_I \end{bmatrix} \begin{bmatrix} \delta\psi_{Ix} \\ \delta\psi_{Iy} \end{bmatrix} = \sum_{I=1}^{n} \Phi_I \delta\psi_I \tag{4.39}$$

Using equations 4.32 and 4.39, $L_1\delta\psi^h$ can be formulated as

$$L_1\delta\psi^h = L_{1(2\text{x}2)}\Phi_{(2\text{x}2\text{n})}\delta\psi_{(2\text{nx}1)}$$
$$= \begin{bmatrix} \frac{d}{dx} & 0 \\ 0 & \frac{d}{dy} \end{bmatrix} \begin{bmatrix} \Phi_1 & 0 & \cdots & \Phi_n & 0 \\ 0 & \Phi_1 & \cdots & 0 & \Phi_n \end{bmatrix} \begin{bmatrix} \delta\psi_{1x} \\ \delta\psi_{1y} \\ \vdots \\ \delta\psi_{nx} \\ \delta\psi_{ny} \end{bmatrix} \tag{4.40}$$
$$= \begin{bmatrix} \frac{d\Phi_1}{dx} & 0 & \cdots & \frac{d\Phi_n}{dx} & 0 \\ 0 & \frac{d\Phi_1}{dy} & \cdots & 0 & \frac{d\Phi_n}{dy} \end{bmatrix} \begin{bmatrix} \delta\psi_{1x} \\ \delta\psi_{1y} \\ \vdots \\ \delta\psi_{nx} \\ \delta\psi_{ny} \end{bmatrix}$$

TH-2607_126102004

$$L_1 \delta\psi^h = D_{1(2\text{x}2\text{n})} \delta\psi_{(2\text{nx}1)} = \sum_{I=1}^{n} D_{1I} \delta\psi_I \tag{4.41}$$

Similarly, $L_2 \delta\psi^h$ can be expressed as

$$L_2 \delta\psi^h = D_{2(2\text{x}2\text{n})} \delta\psi_{(2\text{nx}1)} = \sum_{I=1}^{n} D_{2I} \delta\psi_I \tag{4.42}$$

Using equations 4.38, 4.41 and 4.42, I_1 can be formulated as

$$\begin{aligned} I_1 &= \int_\Omega \left[\sum_{I=1}^{n} D_{1I} \delta\psi_I \right]^T \left[\sum_{J=1}^{n} D_{1J} \psi_I \right] \mathrm{d}\Omega + \\ &\int_\Omega \left[\sum_{I=1}^{n} D_{2I} \delta\psi_I \right]^T \left[\sum_{J=1}^{n} D_{2J} \psi_I \right] \mathrm{d}\Omega \\ &= \int_\Omega \left[\sum_{I=1}^{n} \sum_{J=1}^{n} \delta\psi_I^T \left(D_{1I}^T D_{1J} + D_{2I}^T D_{2J} \right) \right] \psi_J \mathrm{d}\Omega \end{aligned} \tag{4.43}$$

$$\begin{aligned} I_1 &= \sum_{I=1}^{n} \sum_{J=1}^{n} \delta\psi_I^T \left[\int_\Omega \left(D_{1I}^T D_{1J} + D_{2I}^T D_{2J} \right) \mathrm{d}\Omega \right] \psi_J \\ &= \sum_{I=1}^{n} \sum_{J=1}^{n} \delta\psi_I^T K_{IJ} \psi_J \end{aligned} \tag{4.44}$$

where K_{IJ} is the nodal stiffness matrix and is defined as

$$K_{IJ} = \int_\Omega \left(D_{1I}^T D_{1J} + D_{2I}^T D_{2J} \right) \mathrm{d}\Omega \tag{4.45}$$

Redefining limits of summation in equation 4.44 for the entire problem domain transforms it to the following form.

$$I_1 = \sum_{I=1}^{N} \sum_{J=1}^{N} \delta\psi_I^T K_{IJ} \psi_J = \delta\Psi^T K \Psi \tag{4.46}$$

Further, using equations 4.39 and 4.38, the following formulation is derived,

$$\begin{aligned} I_2 &= \int_\Omega \left[\sum_{I=1}^{n} \Phi_I \delta\psi_I \right]^T \theta \mathrm{d}\Omega \\ I_2 &= \sum_{I=1}^{n} \delta\psi_I^T \left[\int_\Omega \Phi_I^T \theta \right] \mathrm{d}\Omega = \sum_{I=1}^{n} \delta\psi_I^T F_I^\theta \mathrm{d}\Omega \end{aligned} \tag{4.47}$$

where $F_I^{(\theta)}$ is a vector of the ratio of space charge density to the Permittivity of silicon. Further,

TH-2607_126102004

redefining limits of summation for the entire problem domain in equation 4.47 transforms it in the form given below.

$$I_2 = \sum_{I=1}^{N} \delta\psi_I^T F_I^\theta \mathrm{d}\Omega = \delta\Psi^T F^\theta \tag{4.48}$$

Similar formulation of I_3 in equation 4.38 can be represented as mentioned below.

$$I_3 = \sum_{I=1}^{N} \delta\psi_I^T F_I^t \mathrm{d}\Omega = \delta\Psi^T F^t \tag{4.49}$$

The last two terms, I_4 and I_5 in equation 4.38 are evaluated by the method of Lagrange multiplier to incorporate essential boundary condition. Discretized formulation of the Lagrange multipliers, λ for a given node at essential boundary is obtained by interpolating its nodal values and shape functions, which can be represented as follows.

$$\delta\lambda^h_{2\text{x}1} = \begin{bmatrix} \delta\lambda_x \\ \delta\lambda_y \end{bmatrix} = \begin{bmatrix} N_1 & 0 & \cdots & N_{n_\lambda} & 0 \\ 0 & N_1 & \cdots & 0 & N_{n_\lambda} \end{bmatrix} \begin{bmatrix} \delta\lambda_{1x} \\ \delta\lambda_{1y} \\ \vdots \\ \delta\lambda_{n_\lambda x} \\ \delta\lambda_{n_\lambda y} \end{bmatrix} \tag{4.50}$$

$$= N(s)_{(2\text{x}2\text{n}_\lambda)} \delta\lambda_{(2\text{n}_\lambda \text{x}1)}$$

where n_λ is the total number of nodes in the local support domain used for interpolation; N_I is shape function for I^{th} node; s is arc-length and λ is vector of nodal Lagrange multipliers on the essential boundary. Equation 4.50 can also be represented in the nodal matrix form as given below.

$$\delta\lambda^h_{2\text{x}1} = \sum_{I=1}^{n_\lambda} \begin{bmatrix} N_I & 0 \\ 0 & N_I \end{bmatrix} \begin{bmatrix} \delta\lambda_{Ix} \\ \delta\lambda_{Iy} \end{bmatrix} = \sum_{I=1}^{n} N_I \delta\lambda_I \tag{4.51}$$

The Lagrange interpolant shape function $N_I(s)$ of order n can be expressed as,

TH-2607_126102004

$$N_k^n(s) = \prod_{\substack{j=0 \\ j \neq k}}^{n} \left[\frac{(s - s_j)}{(s_k - s_j)} \right] \tag{4.52}$$

Equation 4.52 can be evaluated at points $s = s_0$ and $s = s_1$ by using first order linear Lagrange interpolants, and is expressed as

$$N_0(s) = \frac{(s - s_1)}{s_0 - s_1} \tag{4.53}$$

$$N_1(s) = \frac{(s - s_0)}{s_1 - s_0} \tag{4.54}$$

The essential boundaries are analyzed by employing line segments and Lagrange multiplier. The functional defining the essential boundary condition ($\psi = \overline{\psi}$ on Γ_e) can be expressed in an integral form as

$$\begin{aligned} \int_{\Gamma_e} \delta\lambda^T(\psi - \overline{\psi})\mathrm{d}\Gamma = & \int_{\Gamma_e} \left[\sum_{I=1}^{n_\lambda} N_I \delta\lambda_I \right]^T \left[\sum_{J=1}^{n} \Phi_J \delta\psi_J \right] \mathrm{d}\Gamma \\ & - \int_{\Gamma_e} \left[\sum_{I=1}^{n_\lambda} N_I \delta\lambda_I \right]^T \overline{\psi} \mathrm{d}\Gamma \end{aligned} \tag{4.55}$$

$$\begin{aligned} I_4 = & \sum_{I=1}^{n_\lambda} \sum_{J=1}^{n} \delta\lambda_I^T \left[\int_{\Gamma_e} N_I^T \Phi_J \mathrm{d}\Gamma \right] \delta\psi_J \\ & - \sum_{I=1}^{n_\lambda} \delta\lambda_I^T \left[\int_{\Gamma_e} N_I^T \overline{\psi} \mathrm{d}\Gamma \right] \end{aligned} \tag{4.56}$$

$$\begin{aligned} I_4 = & - \sum_{I=1}^{n_\lambda} \sum_{J=1}^{N} \delta\lambda_I^T \left[G_{IJ}^T \right] \delta\psi_J \\ & + \sum_{I=1}^{n_\lambda} \delta\lambda_I^T \left[q_I \right] = \delta\Lambda^T \Big(- G^T \Psi + Q \Big) \end{aligned} \tag{4.57}$$

where, $G_{IJ}^T = -\int_{\Gamma_e} N_I^T \Phi_J \mathrm{d}\Gamma$ and $q_I = -\int_{\Gamma_e} N_I^T \overline{\psi} \mathrm{d}\Gamma$

TH-2607_126102004

$$\begin{aligned}\int_{\Gamma_e}(\delta\psi)^T\lambda \mathrm{d}\Gamma &= \int_{\Gamma_e}\left[\sum_{I=1}^{n}\Phi_I\delta\psi_I\right]^T\left[\sum_{J=1}^{n_\lambda}N_J\lambda_J\right]\mathrm{d}\Gamma \\ &= \sum_{I=1}^{n_\lambda}\sum_{J=1}^{n}\delta\psi_I^T\left[\int_{\Gamma_e}\Phi_I^T N_J \mathrm{d}\Gamma\right]\lambda_J \\ &= -\sum_{I=1}^{n_\lambda}\sum_{J=1}^{N}\delta\psi_I^T\left[G_{IJ}\right]\lambda_J = -\delta\Psi^T G\lambda\end{aligned} \tag{4.58}$$

Now, substituting equations 4.46, 4.48, 4.49, 4.57 and 4.58 in equation 4.38, the following mathematical formulation can be obtained.

$$\delta\Psi^T K\Psi - \delta\Psi^T F^\theta - \delta\Psi^T F^t - \delta\Lambda^T\left(-G^T\Psi + Q\right) + \delta\Psi^T G\lambda = 0 \tag{4.59}$$

Simplified form of equation 4.59 can be represented as

$$\delta\Psi^T\left[K\Psi + G\lambda - \left(F^\theta + F^t\right)\right] + \delta\Lambda^T\left[G^T\Psi - Q\right] = 0 \tag{4.60}$$

or,

$$\delta\Psi^T\left[K\Psi + G\lambda - F\right] + \delta\Lambda^T\left[G^T\Psi - Q\right] = 0 \tag{4.61}$$

Since, in the above equation both $\delta\Psi$ and $\delta\Lambda$ are arbitrary, equation 4.61 can be satisfied if and only if

$$K\Psi + G\lambda - F = 0 \tag{4.62}$$

$$G^T\Psi - Q = 0 \tag{4.63}$$

Therefore, equations 4.62 and 4.63 can be represented as a set of linear algebraic equations which can be analyzed by employing numerical methods and can be represented as

$$\begin{bmatrix} K & G \\ G^T & 0 \end{bmatrix}\begin{bmatrix}\Psi \\ \Lambda\end{bmatrix} = \begin{bmatrix} F \\ Q\end{bmatrix} \tag{4.64}$$

TH-2607_126102004

4.3.2 Discretization of Continuity Equation

The second set of equations in Drift-Diffusion model consist of transport and continuity equations. EFG formulation of these equations at steady state condition with *zero* net generation and recombination is expressed in this section.

$$\frac{\partial n}{\partial t} = \frac{\partial p}{\partial t} = 0 U_n = U_p = 0 \tag{4.65}$$

Thus, the continuity equation turns out to be

$$\nabla \cdot J_n = \nabla \cdot J_p = 0 \tag{4.66}$$

Substitute expressions of J_n and J_p in equation 4.66 and deriving resultant weak-form of these equations leads to the following mathematical formulation.

$$\begin{aligned} &\int_\Omega (L_1 \delta n)^T (L_1 n) \mathrm{d}\Omega + \int_\Omega (L_2 \delta n)^T (L_2 n) \mathrm{d}\Omega \\ &\quad - \frac{1}{V_t} \int_\Omega (\delta n)^T E \cdot (L_1 n) \mathrm{d}\Omega - \int_{\Gamma_n} (\delta n)^T \overline{n}_\tau \mathrm{d}\Gamma \\ &\quad - \int_{\Gamma_e} (\delta \lambda)^T (n - \overline{n}) \mathrm{d}\Gamma - \int_{\Gamma_e} (\delta n)^T \lambda \mathrm{d}\Gamma = 0 \end{aligned} \tag{4.67}$$

$$\begin{aligned} &\int_\Omega (L_1 \delta p)^T (L_1 p) \mathrm{d}\Omega + \int_\Omega (L_2 \delta p)^T (L_2 p) \mathrm{d}\Omega \\ &\quad + \frac{1}{V_t} \int_\Omega (\delta p)^T E \cdot (L_1 p) \mathrm{d}\Omega - \int_{\Gamma_n} (\delta p)^T \overline{p}_\tau \mathrm{d}\Gamma \\ &\quad - \int_{\Gamma_e} (\delta \lambda)^T (p - \overline{p}) \mathrm{d}\Gamma - \int_{\Gamma_e} (\delta p)^T \lambda \mathrm{d}\Gamma = 0 \end{aligned} \tag{4.68}$$

Here, equations 4.67 and 4.68 represent Galerkin weak-form for electron and hole continuity equations, respectively.

Performing the mathematical operations on equations 4.67 and 4.68 similar to Poisson's equation, matrix formulation of transport and continuity equations can be expressed as following system of discrete linear equations below.

TH-2607_126102004

$$\begin{bmatrix} K_n & G_n \\ G_n^T & 0 \end{bmatrix} \begin{bmatrix} n \\ \Lambda_n \end{bmatrix} = \begin{bmatrix} 0 \\ Q_n \end{bmatrix} \tag{4.69}$$

$$\begin{bmatrix} K_p & G_p \\ G_p^T & 0 \end{bmatrix} \begin{bmatrix} p \\ \Lambda_p \end{bmatrix} = \begin{bmatrix} 0 \\ Q_p \end{bmatrix} \tag{4.70}$$

4.4 Parametric Analysis of EFG

A thorough study has been performed to analyze the effect of individual parameters on the accuracy and computation time of the implementation. The various parameters which affects the performance of an EFG based analysis are as follows, d_{max} (size of support domain), choice of shape function, method of discretization, m (number of monomials in the basis function), k (number of points in gauss quadrature integration), n (number of nodes in the domain), and number of points in background cell. The corresponding results have been presented in Tables 4.1 - 4.12. The impact of number of monomials and number of gauss quadrature points on the analysis is not presented due to the fact that increment in their values directly increase the accuracy and computation time, and vice versa. So, these two parameters are kept constant to study the effect of d_{max} (size of support domain), choice of shape function and method of discretization, which does not follow the same convection of directly or indirectly proportionality. Table 4.1 - 4.3 presents the convergence analysis of EFG-MLS method for different number of nodes and gaussian quadrature points, where as, Table 4.4 - 4.6 presents the same anaysis for MWS-MLS. It can be observed from the analysis that the size of support domain (d_{max}) at 3.5 performs better in terms of accuracy as compared to its higher or lower values. Based on Tables 4.1 - 4.6, it can be stated that EFG provides more accurate results as compared to MWS method. Table 4.7 - 4.12 presents the same convergence analysis for RPIM method of discretization. Tables 4.1 - 4.3 and 4.7 - 4.9 can also be referred to state the better performance of MLS as compared to RPIM basis function.

TH-2607_126102004

Table 4.1: Method: EFG-MLS, Parameters: $k = 4$, $mm = 3$, $d_{max} = 2.5$

# of Nodes	# of Gaussian Quadrature Points (GP)					
	GP = 150		GP = 350		GP = 500	
	Error	Time	Error	Time	Error	Time
100	0.003315	2.112991	0.003193	9.472781	0.003045	10.929918
200	0.002960	2.587594	0.002891	9.627192	0.002431	11.132193
300	0.002554	2.851139	0.002611	10.010873	0.001953	11.958240
400	0.001973	2.968687	0.002719	10.227603	0.002047	12.349252
500	0.002284	3.191369	0.003014	10.601859	0.002054	12.839370

Table 4.2: Method: EFG-MLS, Parameters: $k = 4$, $mm = 3$, $d_{max} = 3.5$

# of Nodes	# of Gaussian Quadrature Points (GP)					
	GP = 150		GP = 350		GP = 500	
	Error	Time	Error	Time	Error	Time
100	0.012087	5.050570	0.003941	12.909250	0.003049	13.268244
200	0.009292	5.561108	0.002878	14.316825	0.001933	15.072342
300	0.001975	5.859481	0.002028	15.186016	0.002030	16.312086
400	0.003496	6.090245	0.002601	15.486509	0.001997	16.640590
500	0.002448	6.507966	0.002987	16.232149	0.002065	17.913202

Table 4.3: Method: EFG-MLS, Parameters: $k = 4$, $mm = 3$, $d_{max} = 4.5$

# of Nodes	# of Gaussian Quadrature Points (GP)					
	GP = 150		GP = 350		GP = 500	
	Error	Time	Error	Time	Error	Time
100	0.003126	5.650937	0.004851	16.516326	0.003129	19.737632
200	0.002111	6.666597	0.002927	20.017314	0.002109	24.893156
300	0.011610	7.442970	0.002619	21.110197	0.002076	25.217519
400	0.006559	8.573712	0.002788	21.857830	0.002002	26.317429
500	0.023828	10.384999	0.003135	22.217394	0.039008	29.356805

Table 4.4: Method: MWS-MLS, Parameters: $k = 4$, $mm = 3$, $d_{max} = 2.5$

# of Nodes	# of Gaussian Quadrature Points (GP)					
	GP = 150		GP = 350		GP = 500	
	Error	Time	Error	Time	Error	Time
100	0.069608	0.520227	0.057985	1.034265	0.053826	2.114900
200	0.058826	0.896105	0.051666	1.931793	0.049826	3.216505
300	0.049236	1.325959	0.046274	2.098889	0.043336	3.922184
400	0.043379	1.442037	0.041339	2.569075	0.040379	4.431154
500	0.056949	1.753803	0.051221	3.135136	0.048249	5.523704

TH-2607_126102004

Table 4.5: Method: MWS-MLS, Parameters: $k = 4$, $mm = 3$, $d_{max} = 3.5$

# of Nodes	# of Gaussian Quadrature Points (GP)					
	GP = 150		GP = 350		GP = 500	
	Error	Time	Error	Time	Error	Time
100	0.057083	0.788956	0.011879	1.217313	0.010783	2.521067
200	0.044083	1.201603	0.011201	2.634898	0.010083	3.947082
300	0.037690	1.518937	0.024285	3.196873	0.021690	4.380561
400	0.026224	1.978863	0.015730	3.571289	0.013224	5.930073
500	0.028934	2.210765	0.011405	3.792211	0.008934	6.240989

Table 4.6: Method: MWS-MLS, Parameters: $k = 4$, $mm = 3$, $d_{max} = 4.5$

# of Nodes	# of Gaussian Quadrature Points (GP)					
	GP = 150		GP = 350		GP = 500	
	Error	Time	Error	Time	Error	Time
100	0.174513	0.937042	0.128329	1.549347	0.179513	3.135082
200	0.124513	1.486861	0.057635	2.967145	0.124513	4.793767
300	0.015276	1.746144	0.017436	3.550568	0.015276	5.448619
400	0.048857	2.165526	0.051822	3.964386	0.048857	6.193979
500	0.070645	2.720631	0.035236	4.391344	0.070645	7.416214

Table 4.7: Method: EFG-RPIM, Parameters: $k = 4$, $mm = 3$, $d_{max} = 2.5$

# of Nodes	# of Gaussian Quadrature Points (GP)					
	GP = 150		GP = 350		GP = 500	
	Error	Time	Error	Time	Error	Time
100	0.005719	4.687095	0.005240	10.818942	0.005093	15.051606
200	0.007210	5.141058	0.006287	12.093599	0.006449	16.915538
300	0.003217	5.484822	0.002261	12.355045	0.002329	17.477307
400	0.002116	5.861267	0.002119	12.866888	0.002143	17.763805
500	0.002295	5.902220	0.002097	13.089681	0.002092	18.171231

Table 4.8: Method: EFG-RPIM, Parameters: $k = 4$, $mm = 3$, $d_{max} = 3.5$

# of Nodes	# of Gaussian Quadrature Points (GP)					
	GP = 150		GP = 350		GP = 500	
	Error	Time	Error	Time	Error	Time
100	0.002418	9.502894	0.002246	22.445958	0.002219	31.199670
200	0.004378	12.077525	0.002039	27.576430	0.002017	39.243154
300	0.002462	24.004916	0.001997	56.110717	0.002054	43.441702
400	0.002136	27.778516	0.001869	59.257981	0.002085	51.060258
500	0.001795	29.541870	0.001798	63.388767	0.001693	69.346022

TH-2607_126102004

Table 4.9: Method: EFG-RPIM, Parameters: $k = 4$, $mm = 3$, $d_{max} = 4.5$

# of Nodes	# of Gaussian Quadrature Points (GP)					
	GP = 150		GP = 350		GP = 500	
	Error	Time	Error	Time	Error	Time
100	0.004132	15.816444	0.003532	40.230993	0.003124	56.872389
200	0.003943	39.510776	0.002701	57.294485	0.002597	64.840296
300	0.002574	49.723253	0.002065	63.246879	0.001959	79.013087
400	0.001611	54.999023	0.001951	71.397914	0.001809	86.521029
500	0.006557	57.185944	0.004392	82.934872	0.003927	94.148401

Table 4.10: Method: MWS-RPIM, Parameters: $k = 4$, $mm = 3$, $d_{max} = 2.5$

# of Nodes	# of Gaussian Quadrature Points (GP)					
	GP = 150		GP = 350		GP = 500	
	Error	Time	Error	Time	Error	Time
100	0.007061	1.288586	0.006972	1.501198	0.006193	2.288415
200	0.005364	1.671492	0.004995	1.973581	0.005128	2.961820
300	0.004906	1.980497	0.004162	2.430058	0.003917	3.230492
400	0.004309	2.383645	0.003849	2.866897	0.003748	3.877296
500	0.003759	2.712658	0.003601	3.122944	0.003572	4.627032

Table 4.11: Method: MWS-RPIM, Parameters: $k = 4$, $mm = 3$, $d_{max} = 3.5$

# of Nodes	# of Gaussian Quadrature Points (GP)					
	GP = 150		GP = 350		GP = 500	
	Error	Time	Error	Time	Error	Time
100	0.005356	1.601519	0.005184	1.796017	0.004972	2.620125
200	0.004965	2.578867	0.004397	2.988492	0.004465	3.613808
300	0.003841	5.415952	0.003126	5.880978	0.003812	7.386277
400	0.003332	6.292789	0.002891	6.833291	0.003102	9.427615
500	0.002971	10.143471	0.003469	12.862485	0.002743	16.843253

Table 4.12: Method: MWS-RPIM, Parameters: $k = 4$, $mm = 3$, $d_{max} = 4.5$

# of Nodes	# of Gaussian Quadrature Points (GP)					
	GP = 150		GP = 350		GP = 500	
	Error	Time	Error	Time	Error	Time
100	0.004488	2.037783	0.004137	3.068055	0.006215	5.293128
200	0.003720	5.460392	0.003547	9.257439	0.005124	12.169241
300	0.003413	10.178644	0.003122	14.921830	0.003546	20.082352
400	0.002988	16.902780	0.002730	19.605187	0.003109	24.620138
500	0.002836	17.888186	0.002983	23.021532	0.002791	29.682779

TH-2607_126102004

4.5 Implementation Details and Results

In this section, comparison of VEDA is presented with other methods to validate its effectiveness. As we know, the multi-terminal devices such as MOSFET, FINFET etc. can be modeled by using basic devices and PN junction diodes, we choose PN junction diode as a testcase to validate effectiveness of the EFG method. The simulation results of a PN junction diode are presented in this section for the parameters shown in Table 4.13. Simulation results from the proposed framework are compared with different discretization methods, commercially available TCAD software Sentaurus and analytical results. The EFG method used in the proposed framework provides the most accurate solution as seen in Fig. 4.4. Fig. 4.5 shows the comparison of electron-hole concentration between the proposed framework VEDA and Sentaurus.

Table 4.13: Device parameters of a PN junction diode used in the simulation.

Device Parameters	Value
Length of P-region	3 μm
Length of N-region	7 μm
Doping Profile: N_a	10^{17} cm^{-3}
Doping Profile: N_d	10^{16} cm^{-3}
Temperature: T	300 K

4.5.1 L_2-Norm Error Analysis

The most widely used method of error analysis, L_2-Norm or *Euclidean*-Norm, is used in this chapter to compare results obtained from different discretization scheme. The basic definition, l_2-norm can be defined as follows,

$$\|x\|_2 = \sqrt{\sum_i x_i^2} \tag{4.71}$$

$$\|x_1 - x_2\|_2 = \sqrt{\sum_i (x_{1_i} - x_{2_i})^2} \tag{4.72}$$

For studying proper correlation between two different schemes, the resultant from l_2-norm error analysis is normalized with the length of the vector, and is represented by *Mean-Squared Error* (MSE) as

TH-2607_126102004

follows,

$$MSE(x_1, x_2) = \frac{1}{n}\left\|x_1 - x_2\right\|_2^2 = \frac{1}{n}\sum\nolimits_i (x_{1_i} - x_{2_i})^2 \tag{4.73}$$

Element Free Galerkin method is used to perform simulations for different values of parameters associated with a problem and is compared with the results obtained from analytical and numerical solution (FDM, FEM and TCAD tool Sentaurus) to validate effectiveness of the proposed method. The obtained result in Figure 4.4 shows that EFG outperforms all the other methods in terms of accuracy. Figure 4.5 presents the comparison of carrier concentration computed using EFG and *Sentaurus*. It further validates effectiveness of the proposed implementation in terms of accuracy.

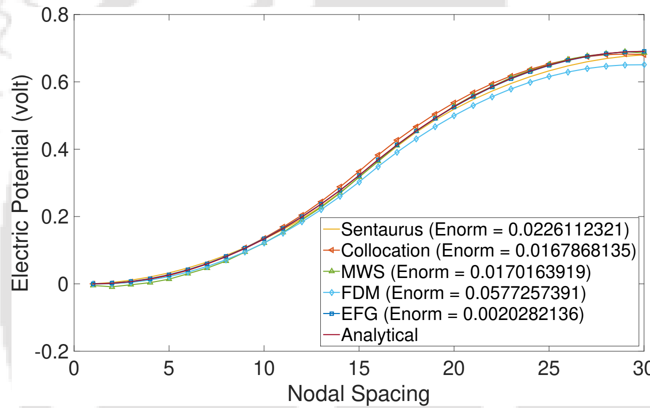


Figure 4.4: Comparison of different solvers with analytical solution of Poisson's equation.

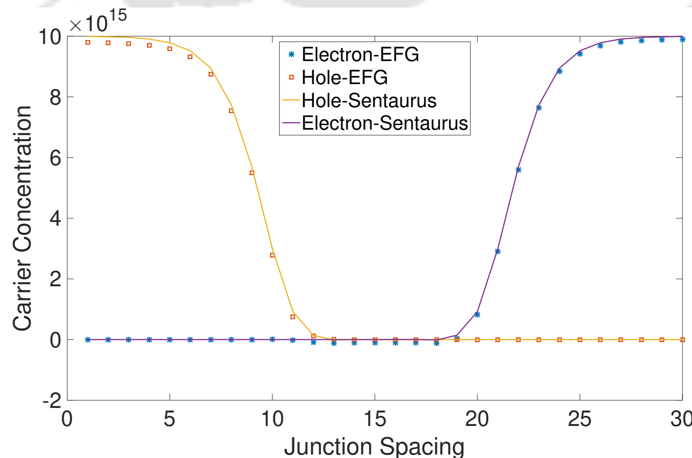


Figure 4.5: Comparison of carrier concentration computed using EFG and Sentaurus.

Two different interpolation methods, MLS and RPIM are employed for interpolation. EFG along with MLS provides the least normalized error. The effect of size of the support domain is investigated

TH-2607_126102004

also as shown in Figure 4.6, and it can be observed that EFG with MLS performs better at $d_{max} = 3$. Figure 4.7 presents the comparison of normalized error for different Gaussian points of integration, and it can be observed that EFG-MLS produces less error as compared to EFG-RPIM.

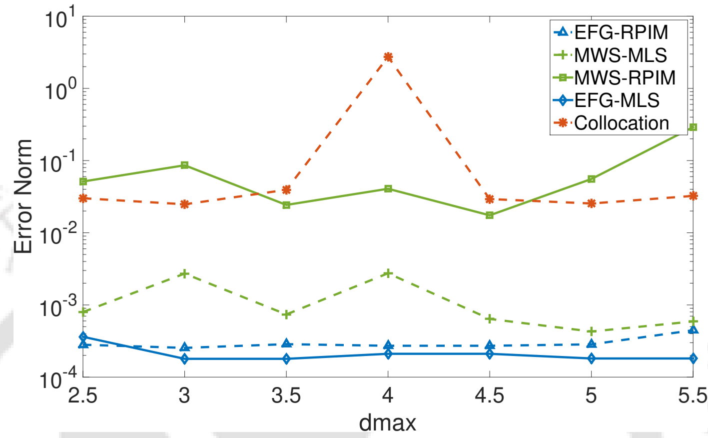


Figure 4.6: Comparison of normalized error for different sizes of support domain.

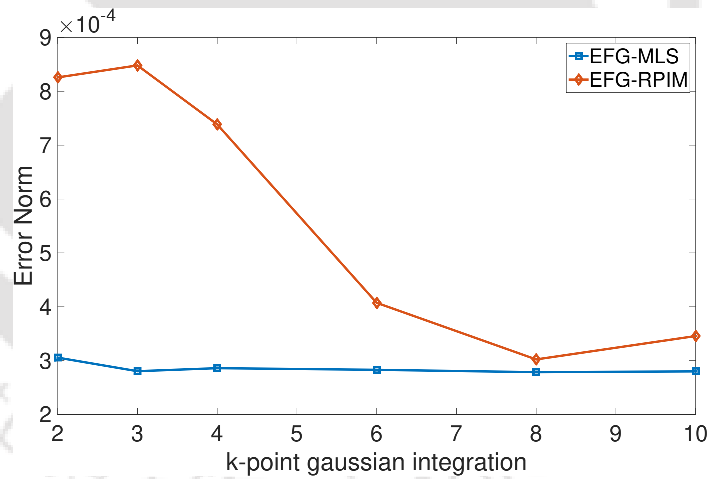


Figure 4.7: Comparison of normalized error for different Gaussian points of integration.

Simulations are performed to analyze the performance of the developed framework for different number of nodes. Figures 4.8 and 4.9 presents this study, which states that EFG produces the least normalized error for MLS and RPIM as compared to any other discretization method.

4.5.2 Computational Complexity

The high accuracy of EFG method results into more computational time when compared to other discretization schemes. This increase in computation time is mainly due the formation of localized

TH-2607_126102004

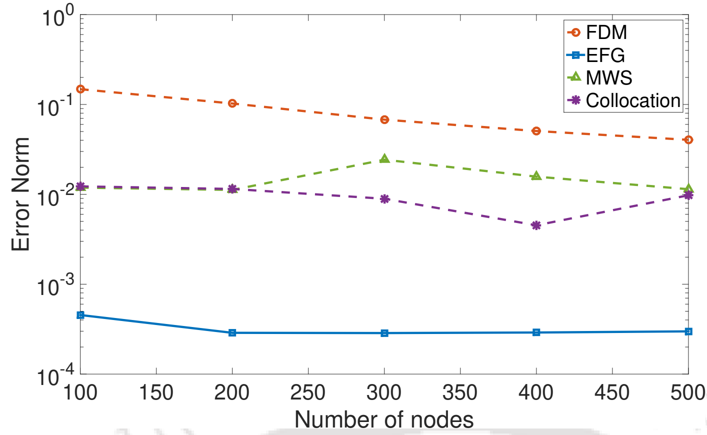


Figure 4.8: Comparison of normalized error for different number of nodes with MLS method.

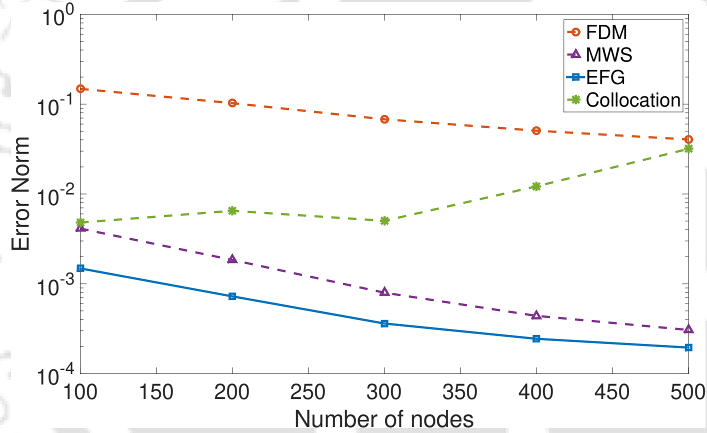


Figure 4.9: Comparison of normalized error for different number of nodes with RPIM method.

shape functions within the support domain of the node of interest. The computational complexity of a particular discretization scheme is determined by the formulation and size of its stiffness matrix, which is represented in the form of pseudocode in Algorithm 2 for EFG. Fig. 4.10 shows the comparison of computation time for different discretization schemes. Since EFG provides scope of embarrassing parallelization when clubbed with accelerated gradient methods, its computation time can be reduced by incorporating different parallelization techniques over manycore architecture.

4.6 Summary

In this chapter, we propose Element-Free Galerkin method along with MLS shape function to design a framework for analyzing semiconductor devices. Since the EFG method depends on a set

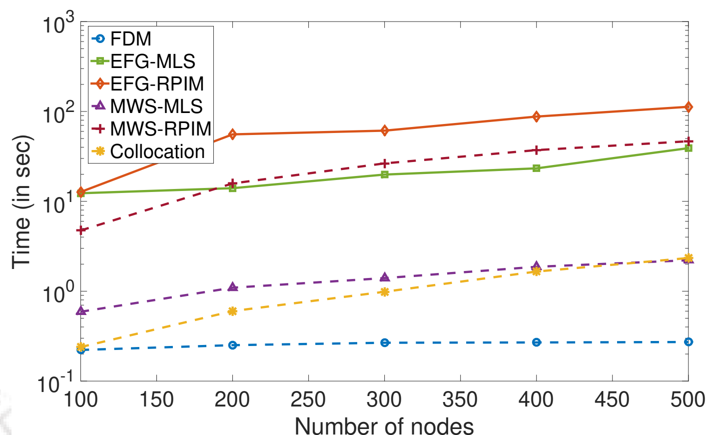


Figure 4.10: Comparison of computation time for different number of nodes.

of independent nodes for discretization, it is highly suitable for adaptive analysis. Simulation results are compared with different discretization schemes and commercially available TCAD software, *Sentaurus* in order to validate effectiveness of the proposed framework VEDA which has extended with EFG method. It is observed that EFG with MLS provides the least normalized error for PN junction diode as compared to other methods and produces $10\times$ accurate solution as compared to Sentaurus. Different parameters affecting the accuracy of EFG methods are analyzed, and corresponding results are presented to find their best possible values. EFG method provides a suitable way to trade-off accuracy with the computation time by forming localized shape functions within support domains of all the nodes. The computation time of the EFG method can be further reduced by incorporating different parallelization schemes in future.

TH-2607_126102004

TH-2607_126102004

5

Random Walk and Machine Learning Algorithms based Analysis of Semiconductor Devices

Indian Institute of Technology Guwahati

Contents

5.1 Proposed TCAD framework using Random Walk Algorithm 93
5.2 Implementation Details and Simulation Results 99
5.3 Proposed TCAD Framework using Machine Learning 103
5.4 Circuit Analysis Using Machine Learning 111
5.5 Summary 118

TH-2607_126102004

In this chapter, we propose *Random Walk* and *Machine Learning* based methodologies to speed up computation time of the device analysis and circuit simulation of novel devices. Our proposed simulator VEDA is extended to incorporate these methods. In the case of random walk method, equivalent electrical circuits of various fundamental device equations are analyzed in a coupled manner and the solution of TCAD analysis is formulated using the solution of electrical circuits. The random walks is a method based on the probability distribution calculated using node-weights and is embarrassingly parallelizable. The proposed method speeds up VEDA to produce 20% faster solution as compared to its sequential counterpart with the maximum error of 4%.

In the machine learning based approach, artificial neural networks are employed to generate a model to predict potential profile inside a semiconductor device for a particular set of parameters. The application of machine learning algorithms helps in getting semiconductor device equations or transport model to be analyzed in fewer steps, therefore, speeding up the entire device simulation. The employment of ANN provides the minimum speedup of $1.86\times$ (13 iterations to 7 iterations) and the maximum speedup of $3.5\times$ (21 iterations to 6 iterations) for 12 different parametric variations. This also enables us to perform discrete analyses of semiconductor devices in a given condition and reduces efforts of rigorous TCAD analysis to generate semiconductor device models for the electrical circuit analysis with acceptable inaccuracy. The proposed work is extended to conduct circuit analysis incorporating machine learning based semiconductor device models. Simulation results obtained using proposed analysis are validated with commercially available software, *Spectre* provided by Cadence Design Systems.

The machine learning based circuit analysis engine has a distinct advantage of less memory consumption as compared to look up table (LUT) based circuit analysis technique. This also paves way to invent a methodology for developing neural network based semiconductor device models and associated circuit simulation techniques. This initiative also envisages efforts in semiconductor device model development getting minimized during early stage of device development. It also helps in studying impact of variations of novel devices parameters on the performance of large circuits composed of novel devices. This would direct scientific community to concentrate efforts in the development of those novel devices which have potential to address issues of future electrical circuits

TH-2607_126102004

as per our need.

5.1 Proposed TCAD framework using Random Walk Algorithm

Random Walk (RW) method is a method based on probability and the weight distribution of each node in a given graph [16]. The simplest form of random walk can be represented by giving an example of a person who starts walking from the origin in a 2D plane and takes left or right turn randomly. It is similar to tossing a weighted coin at each step and wait for a probable outcome. Based on head or tail, decision of left or right turn is taken, respectively. Suppose that a drunkard (person) is allowed to roam in a city where the roads are laid down in a square format as shown in Figure 5.1. He can move in any direction at each crossing with a probability of $1/4$. While passing a corner or crossing, he has to pay penalty as his traveling expenses. When the drunkard reaches at one of his destinations, he is rewarded for reaching home and the walk comes to an end. As there can be more than one destinations, the drunkard can end his walk anywhere in the city having different rewards for different walks. It can be concluded from Figure 5.1 that the crossings may be considered as nodes and the path connecting to any two crossings may be interpreted as an edge.

The random walk of drunkard inspired scientific community [57, 58] to analyze electrical circuits especially power grid network of VLSI circuits [23, 58] to determine node potentials of power grid network composed of resistors, current sources and voltage sources for finding hotspots. In the application of random walk method to power grid network analysis, node potential is analogous to the reward earned by the drunkard. The accuracy of the solution provided by this method depends on the total number of walks initiated at each node. More the number of walks on each node, better the accuracy of the solution (node potential) is, which can be computed in parallel independently and is averaged at the end of the walks to obtain final node potential (with some error). The estimation of node potential through random walk method can be considered as an electrical network analysis through nodal analysis method. This process is embarrassingly parallel in nature since the solution of each node can be computed in parallel independently and does not depend on the solution of other nodes.

Random walk method has already been proposed to work efficiently for a generic Poisson's equa-

TH-2607_126102004

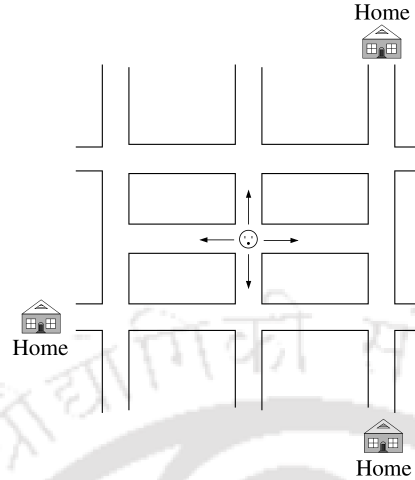


Figure 5.1: Random walk by a drunkard in a city.

tion in [17], where a partial differential equations is first converted to its equivalent electrical circuit and is solved by employing random walk method for a fast approximate solution. In the proposed work, we extend application of random walk method for all the fundamental equations of Drift-Diffusion model being analyzed in a coupled manner while analyzing semiconductor devices. The solution of these equations generates a less computationally extensive approximate solution which can be utilized as an initial guess for the analysis of semiconductor devices.

As we know that the main advantage of random walk method is its parallelizability. All the nodes of a problem domain can be analyzed independently to compute its solution quickly. Since, this method produces an approximate solution which always has certain acceptable error, it can only be used as an initial guess of the analysis instead of the final solution. A better initial guess provided using random walk method helps VEDA to converges faster to the final solution, thereby, reducing total computation time. A typical process flow of device simulator using random walk method to compute initial guess is shown in Fig. 5.2. It can also be stated that the random walk method is incorporated into our indigenous TCAD device analyzer VEDA to obtain an initial guess and to perform further analysis of semiconductor devices using various discretization schemes proposed in chapters 3 and 4.

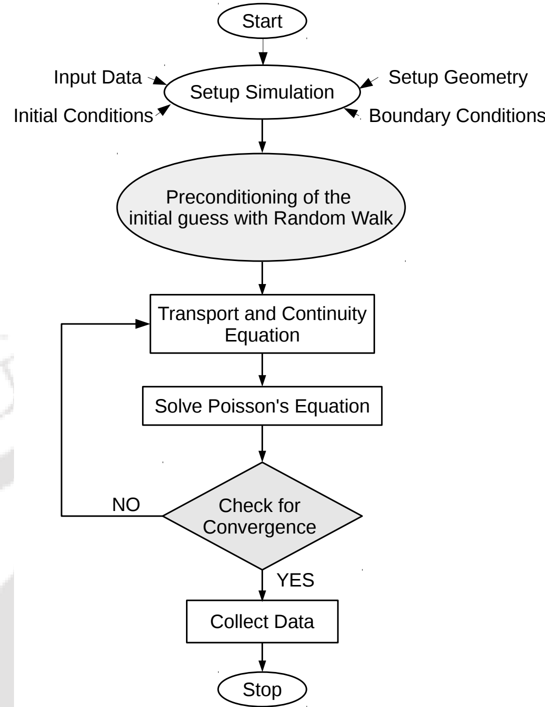


Figure 5.2: Process flow of proposed TCAD framework VEDA with random walk method

5.1.1 Discretization of PDEs and their Equivalent Electrical Circuit

Equivalent electrical circuits of various partial differential equations exhibiting Drift-Diffusion transport model are presented in this section. We begin with describing two dimensional Poissons equation's formulation into an equivalent electrical circuit below.

$$\nabla^2 u_{(x,y)} = C \quad or \quad \frac{d^2u}{dx^2} + \frac{d^2u}{dy^2} = C_{(x,y)} \tag{5.1}$$

Using finite difference method, above mentioned second order equation 5.1 can be expressed in its discretized form below.

$$\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{(\triangle x)^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{(\triangle y)^2} = C_{i,j} \tag{5.2}$$

Equation 5.2 can be rearranged in the following manner to be interpreted as nodal analysis expression.

$$\frac{u_{i-1,j} - u_{i,j}}{(\triangle x)^2} + \frac{-u_{i,j} + u_{i+1,j}}{(\triangle x)^2} + \frac{u_{i,j-1} - u_{i,j}}{(\triangle y)^2} + \frac{-u_{i,j} + u_{i,j+1}}{(\triangle y)^2} = C_{i,j} \tag{5.3}$$

The above mentioned nodal expression can be represented in the form of its equivalent electrical circuit for a single node and is shown in Fig. 5.3. The same equivalent circuit is valid for all the

TH-2607_126102004

nodes except boundary nodes. For nodes at the boundaries, their equivalent electrical circuit can be formulated using two different boundary conditions shown below.

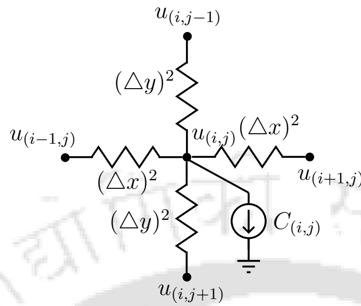


Figure 5.3: Equivalent electrical circuit of 2D Poisson's equation for a single node

Dirichlet Boundary Condition

As it is already defined that Dirichlet boundary condition implies solution to be known at boundary. The expression for Dirichlet boundary for a region bounded by $x \in [0, 1]$ can be presented as The equivalent electrical circuit for Dirichlet boundary can be shown in Fig. 5.4.

$$\frac{d^2(u)}{dx^2} = f \quad x \in [0, 1] \tag{5.4}$$

$$u_{x=0} = C_1 \ \text{[Constant]} \tag{5.5}$$

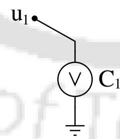


Figure 5.4: Equivalent electrical circuit for a Dirichlet boundary condition

Neumann Boundary Condition

The Neumann boundary condition implies that the value of derivative is known at the boundary and is shown below for the completeness. The equivalent electrical circuit for Neumann boundary can be presented in Fig. 5.5.

TH-2607_126102004

$$\frac{d^2(u)}{dx^2} = f \quad x \in [0,1] \tag{5.6}$$

$$\frac{d(u)}{dx}|_{x=0} = C_2 \;\; [\text{Constant}] \tag{5.7}$$

Figure 5.5: Equivalent electrical circuit for a Neumann boundary condition

The boundary condition for potential at ohmic contacts can be considered as Dirichlet boundary condition making it equal to the applied voltage. At remaining boundaries Neumann condition is applied by computing gradient of the potential equal to zero or constant depending on the interface. Boundary conditions for carrier concentrations can be well established by assuming zero space charge and thermal equilibrium at ohmic contacts.

$$\begin{aligned} np - n_i^2 &= 0 \\ n - p - N &= 0 \end{aligned} \tag{5.8}$$

Dirichlet boundary conditions for electrons and holes at ohmic contact can be deduced by rearranging equation 5.8 as

$$n = \frac{\sqrt{N^2 + 4n_i^2} + N}{2}, \tag{5.9}$$

$$p = \frac{\sqrt{N^2 + 4n_i^2} - N}{2}. \tag{5.10}$$

In this way, a second order partial differential equation can be easily converted to its equivalent electrical circuit for the whole geometry. Figure 5.6 shows an equivalent electrical circuit for the Poisson's equation for a defined geometry. Now, this electrical circuit can be analyzed using random walk method to produce approximate initial guess of the solution of semiconductor device analysis.

In the similar manner an equivalent electrical circuit for the current equation representing Drift-Diffusion model can also be deduced. As it is already described in chapter 2, by substituting current

TH-2607_126102004

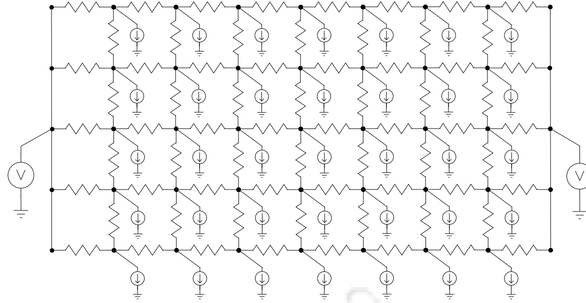


Figure 5.6: Equivalent electrical circuit of the Poisson's equation over a rectangular geometry

equation into continuity equation, we can express the combined equation as

$$\frac{\partial n}{\partial t} = \frac{1}{q}\nabla \cdot (qn\mu_n E + qD_n \nabla n) + U_n \tag{5.11}$$

After discretization, equivalent electrical circuit of the above mentioned combined current equation is formulated and is shown in Fig. 5.7 for a single node. Similarly, Fig. 5.8 presents an equivalent electrical network for the current equation over the complete geometry.

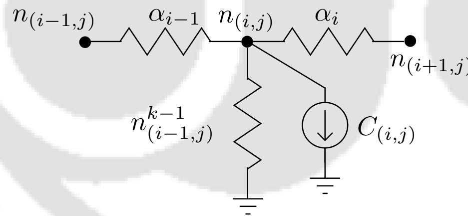


Figure 5.7: Equivalent electrical circuit of a current equation for a single node

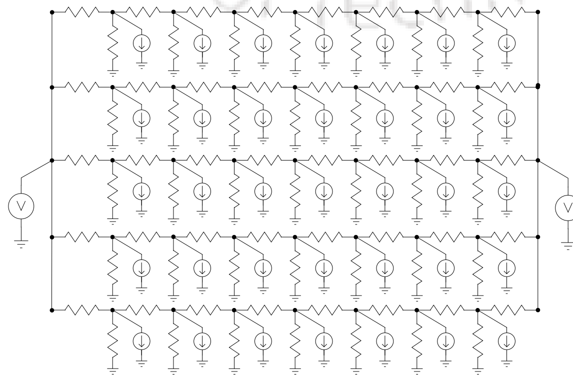


Figure 5.8: Equivalent electrical circuit of the Current equation over a rectangular geometry

It can be observed that random walk method can be easily applied to obtain approximate solution of the cross coupled equations representing Drift-Diffusion model. This approximate solution can be utilized as an initial guess for the analysis using the proposed framework VEDA which employs various discretization schemes, such as FEM, DG-FEM, SUPG, and EFG. It should be noted that the equivalent electrical network formulation is developed using FDM and there would be an endeavour to extend formulation of equivalent electrical networks for other discretization schemes as well. It is anticipated that the equivalent circuit formulation of FEM, FVM, EFG etc. methods would help random walk based initial guess estimation engine to produce a better initial guess.

5.2 Implementation Details and Simulation Results

In particular, Drift-Diffusion model comprises of transport equations describing the flow of carriers due to two different phenomena namely, *Drift* and *Diffusion*. As it is already stated, drift phenomenon describes the flow of carriers due to rate of change of the potential or effect of electric field, where as diffusion depicts the flow of carriers due to concentration gradient. In the proposed work, equivalent electrical circuits of both drift and diffusion terms are formulated to obtain an approximate solution using random walk method. The pseudocode of the proposed methodology is described in Algorithm 3.

The effectiveness of random walk method is validated by analyzing large scale benchmark electrical circuits composed of resistors, current sources and voltage sources. Table 5.1 presents speedup analysis of random walk method by comparing its execution time on Intel processor based machine and on Intel Xeon Phi co-processor. $t_{RW}^{serial}(s)$ is the time taken by random walk algorithm on the single processor performing all the walks sequentially. $t_{RW}^{XeonP}(s)$ is the time taken by OpenMP based random walk method on Intel Xeon Phi co-processor, which has 61 cores having each core four threads. One core of the co-processor is employed for the job scheduling purpose. Therefore, utilizing all 240 threads is the best choice to parallelize random walk method if input data size is large enough to achieve optimum performance. It can be observed from Table 5.1, as the size of input electrical network increases, speedup obtained using random walk method also increases from $9\times$ to $16\times$ with an average error of 4%. This makes random walk method a suitable choice for the approximate analysis

TH-2607_126102004

Algorithm 3: Pseudocode of Random Walk Algorithm

```
for i → 1, total no. of nodes do
    for j → 1, no. of walks do
        total reward → 0;
        while node type ≠ home do
            pick a random number;
            for k → 1, no of neighbours do
                compare random number with probability;
                if random number < probability then
                    break;
                end
            end
            find node id of k^th node;
            update neighbour and total reward;
        end
        update node potential;
    end
    node potential ← node potential walks;
    node type ← home;
    reward ← node potential;
end
```

of electrical networks.

Table 5.1: Speedup Analysis for IBM benchmarks on Intel Xeon Phi

Benchmarks	t_{RW}^{serial} (s)	$t_{RW}^{XeonPhi}$ (s)	**Speedup**
ibmpg2	0.796	0.086	9.2×
ibmpg4	5.144	0.621	8.3×
ibmpg5	12.73	0.809	15.7×
ibmpg6	21.675	1.317	16.4×

As it is already discussed, the main advantage of using random walk method is its parallelizability to calculate initial guess of the solution. Values at different nodes of an electrical circuit can be computed independently to obtain faster results by harnessing the power of various parallelization techniques, such as OpenMP, MPI-OpenMP, CUDA Blocks, and CUDA Threads. Fig. 5.9 shows the computation time of various parallelization schemes employed in the analysis of semiconductor devices using random walk method. It can be observed that CUDA with varying number of threads

TH-2607_126102004

produces the best results among all parallelization schemes. Fig. 5.10 presents the effectiveness of random walk method as compared to that of KLU sparse matrix solver. It is observed that the random walk method is at least $5\times$ faster as compared to KLU for any domain size. The analysis of PN junction diode using VEDA based on random walk is presented in the next section.

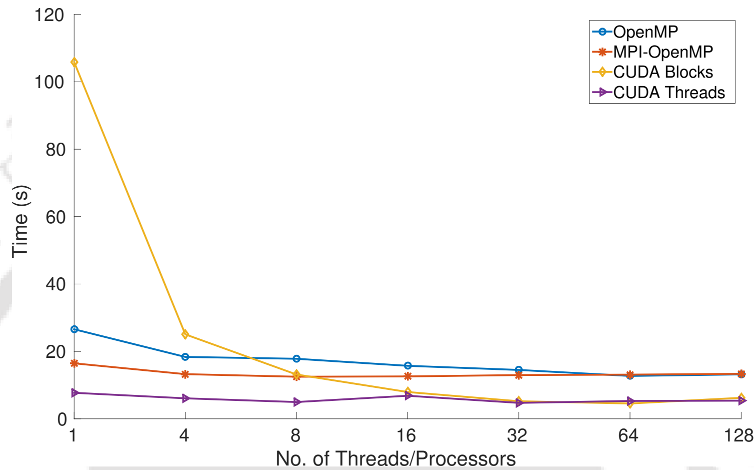


Figure 5.9: Computation time of various parallelization schemes employed in Random Walk

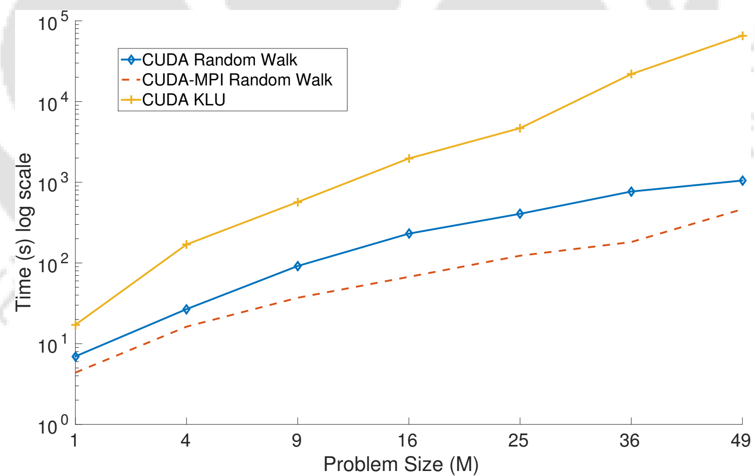


Figure 5.10: Comparison of computation time for CUDA and CUDA with MPI as problem size increases

5.2.1 PN Junction Diode

Different parameters used to analyze a silicon based PN junction diode are presented in Table 5.2. Fig. 5.11 shows percentage error of the initial guess with different number of walks per node taken in random walk method. It can be observed that the error in approximate solution as compared to the

actual solution reduces with the increase in number of walks. The error in the approximate solution reduces to $< 2\%$ when number of walks are increased to 100. Thus, a multi-core processor is an ideal platform for the analysis of semiconductor devices using random walk method. Initial guess for the analysis of PN junction diode is determined using random walk method and the corresponding potential profile inside a diode at thermal equilibrium is shown in Fig. 5.12. Average error produced in the estimation of an initial guess is approximately 4% as compared to the actual solution. The initial guess obtained using random walk method aids in getting the final solution of Drift-Diffusion equations converged in 6 iterations as compared to 11 iterations, when random walk method based initial guess estimation engine is not employed. Built-in potential of a PN junction diode is found to be approximately $0.67V$ matching the desired specifications.

Table 5.2: Various device parameters used for a PN diode

Device Parameters	Value
Length of P-region	30 μm
Length of N-region	70 μm
Doping Profile: N_a, N_d	10^{16} cm^{-3}
Hole Mobility: μ_p	480 $cm^2/V-s$
Electron Mobility: μ_n	1350 $cm^2/V-s$
Temperature: T	300 K

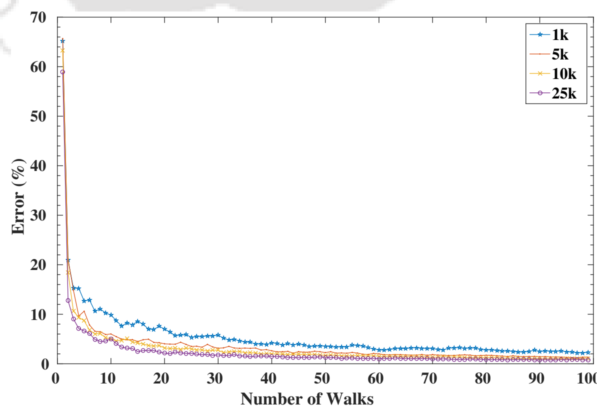


Figure 5.11: Plot showing percentage of error with number of walks at different number of nodes

TH-2607_126102004

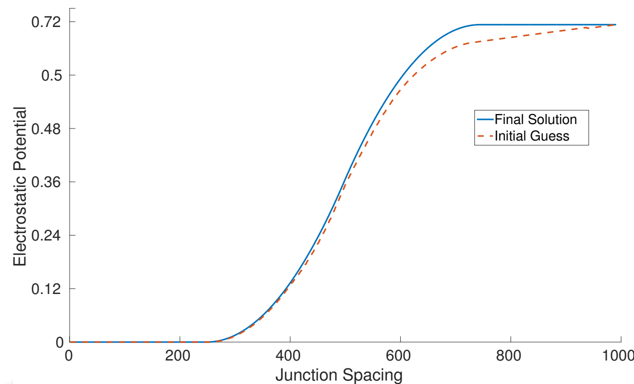


Figure 5.12: Comparison of initial guess (using Random Walk) of Poisson's equation with its final solution

5.3 Proposed TCAD Framework using Machine Learning

In this section, different *machine learning algorithms* are discussed which aim to produce a good prediction model for the estimation of initial guess to be utilized in the analysis of semiconductor devices. Logistic Regression, Decision Tree, Random Forest, Support Vector Machine (SVM), Naive Bayes and Deep Learning with Artificial Neural Network are some of the leading machine learning algorithms which are extensively studied to develop a better model for the analysis of semiconductor devices. Similar to the random walk method, VEDA is further extended to incorporate machine learning techniques to obtain an initial guess for the TCAD analysis. Process flow exhibiting extension of VEDA with machine learning algorithm is presented in Fig. 5.13.

The fundamental principle of machine learning algorithms is to learn and create models based on a predetermined dataset by employing different error minimization techniques. Thus, a larger and vast dataset is required to produce a more accurate model. Similar dataset for the analysis of semiconductor devices is generated using different parameters, such as dimension, doping, biasing, and carrier concentration. Different machine learning algorithms to create a good prediction model are described in the following sections.

5.3.1 Logistic Regression

Logistic regression is one the most powerful statistical method which is generally used to model the binomial outcome of the analysis with one or more input variables. In this method, a logistic

TH-2607_126102004

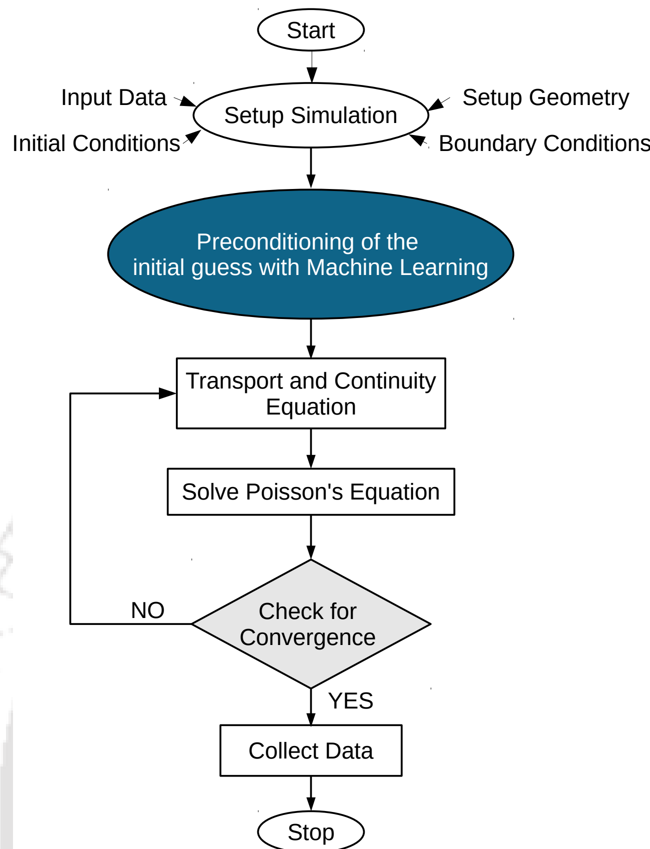


Figure 5.13: Process flow of the proposed framework VEDA using machine learning algorithm

function is defined to model the correlation between one or more input (independent) variables, and a output (dependent) variable based on probabilities. Therefore, this method is more suitable for the applications where dependent variables are categorical in nature. Most common usage of logistic regression method can be found in the applications, such as spam filter for emails or messages, and malignant test for a tumor etc. Thus, it can be said that this method is used to predict the outcome with discrete values, which can be either yes or no, true or false, or $0, 1$. Since this method predicts based on the probability, it generates output ranging from 0 to 1, which can be further divided based on a certain threshold. This method is also known as logit regression because a logit function is used to fit or predict the data based on their probability of occurrence.

Strengths and Weaknesses: The most common problem in machine learning methods is over-fitting of the data. Logistic regression method performs well to avoid such scenarios of over fitting and produces nice probabilistic correlation between independence and dependent variables. Although

TH-2607_126102004

this method does not performs well for a multiple outcome problem or when a nonlinear output is desired. Therefore, this method does not performs well to model the initial guess of the system for the analysis of semiconductor devices.

5.3.2 Decision Tree

It is one of the most frequently used classification algorithm to be employed in the applications with continuous or nonlinear outcomes. This method is based on the idea of supervised learning and categorizes the data based on certain levels of thresholds. It can be better understood with Fig. 5.14 which presents the classification technique for a decision tree algorithm. As it can observed that in this method the data is split into maximum possible homogeneous sets based on the threshold to produce a good predictive model.

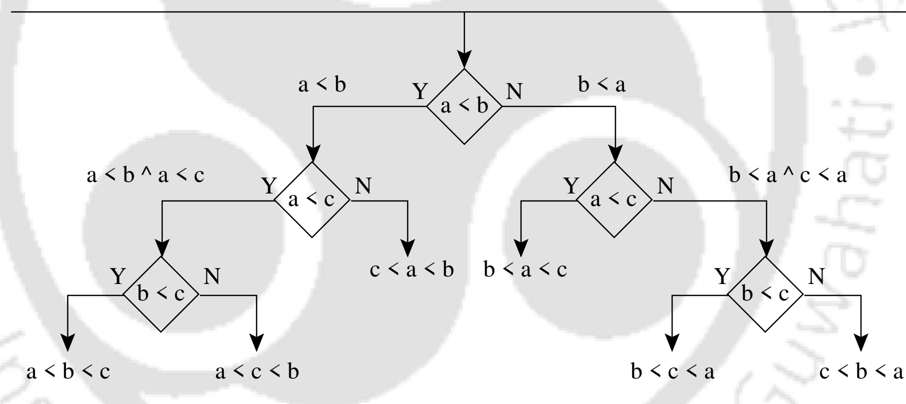


Figure 5.14: Basic principle of decision tree algorithm.

Strengths and Weaknesses: The most advantageous feature of decision tree algorithm is to model the nonlinearities of a dataset, and thus can be used for both continuous or categorical dependent or output variables. The only disadvantage of this method is over-fitting, because if it is left unchecked then this method can keep subdividing the data for numerous homogeneous sets which can produce bad predictive models. For a nonlinear output, this algorithm produces fair results only if the number of independent variables are ≤ 2, thus making it not suitable for semiconductor device analysis having > 2 varying parameters.

TH-2607_126102004

5.3.3 Random Forest

Random forest algorithm is an advanced or ensembled version of decision tree. It is a collection of individual decision trees, hence the name *Forest*. In this algorithm each individual tree cast its vote for the probable outcome. For a defined set of input data, an outcome with largest number of votes or decision trees is selected by the forest.

Strengths and Weaknesses: The main advantage of this method lies in its scope for parallelization because each individual decision tree can be obtained in parallel. The issue of over-fitting is also reduces due to the collective decision of all decision tress. The main disadvantage of this method is its complexity to construct the forest which makes it a harder and less intuitive approach. Similar to the decision trees, fair results are obtained only if the number of independent variables are ≤ 2, thus, making it not suitable for semiconductor device analysis having > 2 varying parameters.

5.3.4 Support Vector Machine (SVM)

Support vector machine (SVM) is another robust and good classification algorithm which works on the principle of dividing the pool of dataset into finite number of hyperplanes. For a set of data points with two different characteristics in a n dimensional space, SVM creates $n - 1$ dimensional hyperplane to distinguish them into two different groups. This examples can be best visualized using Fig. 5.15.

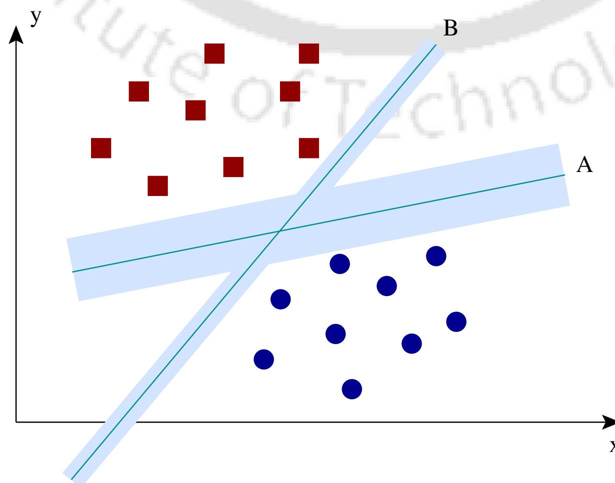


Figure 5.15: A 2-dimensional data set in support vector machine

TH-2607_126102004

Strengths and Weaknesses: SVM algorithm is quite robust against over-fitting and performs well for a high dimensional space. It can easily model nonlinear outcomes, and has many kernels or parameters to select from. Its high memory utilization and cumbersome process to tune the algorithm makes it less popular among different machine learning algorithms. SVM does not tend to adhere to the concept of scalability which makes it unsuitable for the analysis of semiconductor device.

5.3.5 Naive Bayes

This classification algorithm is based on the principle of *Bayes' theorem*. It assumes that the different predictors or characteristics are independent in nature, such that the effect of one particular input variable on the outcome is not related to the effect of the second input variable.

Naive Bayesian algorithm is easier to construct, thus, it can be efficiently used with huge datasets. Besides its simple nature, it is also known to outperform other more sophisticated classification techniques, such as decision trees and random forest. The technique to obtain a probable outcome is formulated by computing posterior probability as expressed in the following equation.

Likelihood   Class Prior Probability

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Posterior Probability   Predictor Prior Probability

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \cdots \times P(x_n|c) \times P(c) \tag{5.12}$$

where $P(c)$ is prior probability of the outcome (class); $P(x)$ is prior probability of input (predictor); $P(x|c)$ is the likelihood that a given input belongs to the given outcome and $P(c|x)$ is the posterior probability of an outcome correspond to a given input.

Most commonly know applications where Naive Bayes algorithm outperforms better classification techniques are listed below.

- Facial recognition applications
- Spam filtering of messages and emails
- Classifying news article among categories, such as politics, technology, finance and sports

TH-2607_126102004

- Classify a text based on sentiments

Strengths and Weaknesses: Even though the assumption of conditional probability or the independent nature of input variables rarely holds true for real world applications, this models actually produces good results in practice. It is very easy to implement and can easily adapt with a large scale dataset. The only disadvantage of this method is its simplicity, where it is often outperformed by other well tuned machine learning algorithms.

5.3.6 Deep Learning with Artificial Neural Network

Deep learning is another domain of machine learning which employs a network of neurons to create a learning model. The weight of each neuron in the network is fixed in such a way that it minimizes the cost function. Fig 5.16 presents basic principle of a neural network in which a neuron gets data input from the input layer and pass it to the output layer or to the another neuron after processing. The predicted outcome is then compared with its known value to estimation the error which is again back-propagated to adjust the weights for minimum error making it an iterative process. A large training dataset is required to train the neural network for good predictions on the test samples. When the number of hidden layers in a network are increased to more than two, the system becomes deep neural network as presented in Fig. 5.17. A optimum number of neurons are required in the network to create a good prediction modal for a particular set of input parameters.

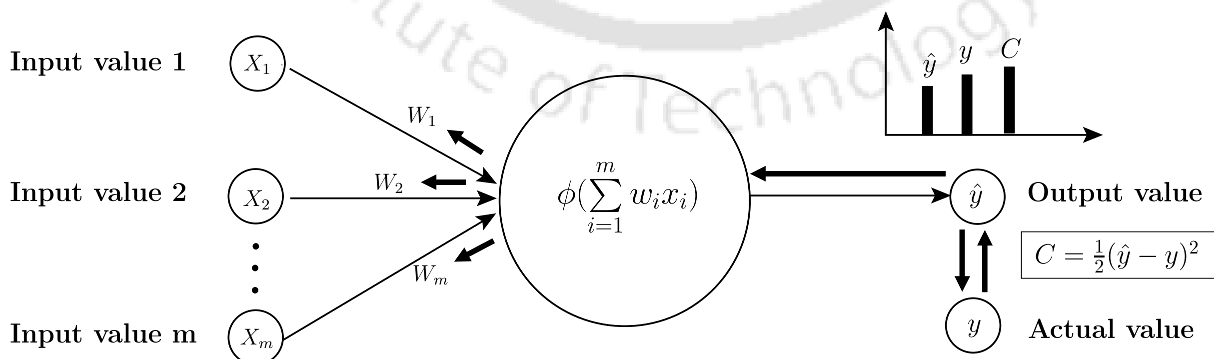


Figure 5.16: Basic principle of an artificial neural network

In artificial neural networks, the function used to define outcome of a particular node or neuron based on an input or a set of inputs is called activation function. Two different types of activation

TH-2607_126102004

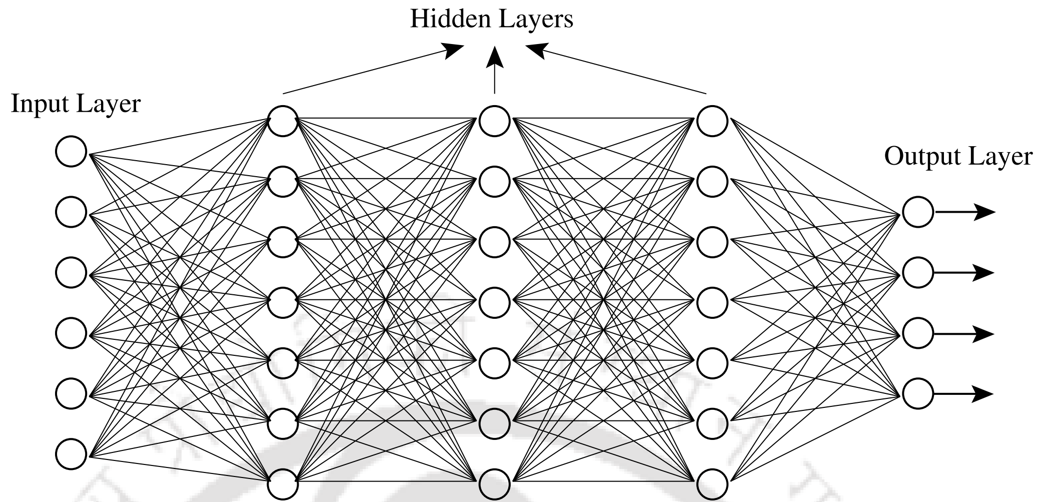


Figure 5.17: A deep neural network with multiple hidden layers

functions utilized to train the neural network for semiconductor device analysis are *Sigmoid* and *Rectifier* functions, which are shown in Fig. 5.18.

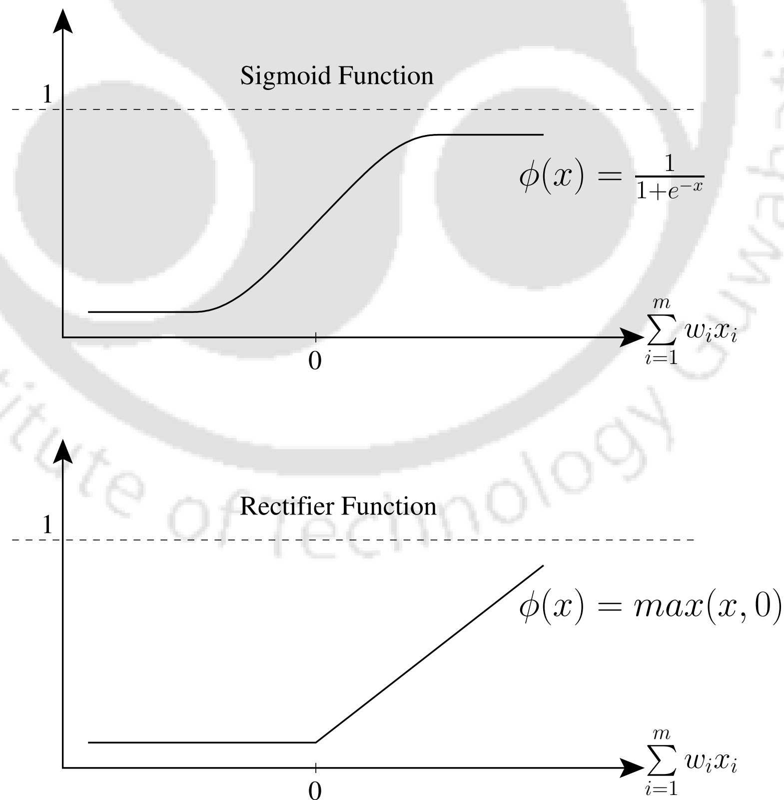


Figure 5.18: Two different activation functions used in model formation: Sigmoid and Rectifier

A huge dataset is generated to train the model based on the following parameters.

(i) Dimension (Input)

TH-2607_126102004

(ii) Material (Input)

(iii) Doping Concentration (Input)

(iv) Carrier Concentration (Input)

(v) Potential Profile (Output)

Our trained model consists of a total of 513 neurons on 5 hidden layers with different activation functions. The training dataset can be generated employing VEDA or commercially available TCAD simulator, *Sentaurus*. The model is trained to produce error $\leq 5\%$, and is used later to predict potential profile for different parameters other than the training dataset. The predicted potential profile is observed to be very close to the actual potential profile, thus, it can be used as an initial guess to achieve faster convergence. Figs. 5.19 and 5.20 present comparison of the predicted potential profile and actual potential profile near the depletion region for different doping concentrations, respectively.

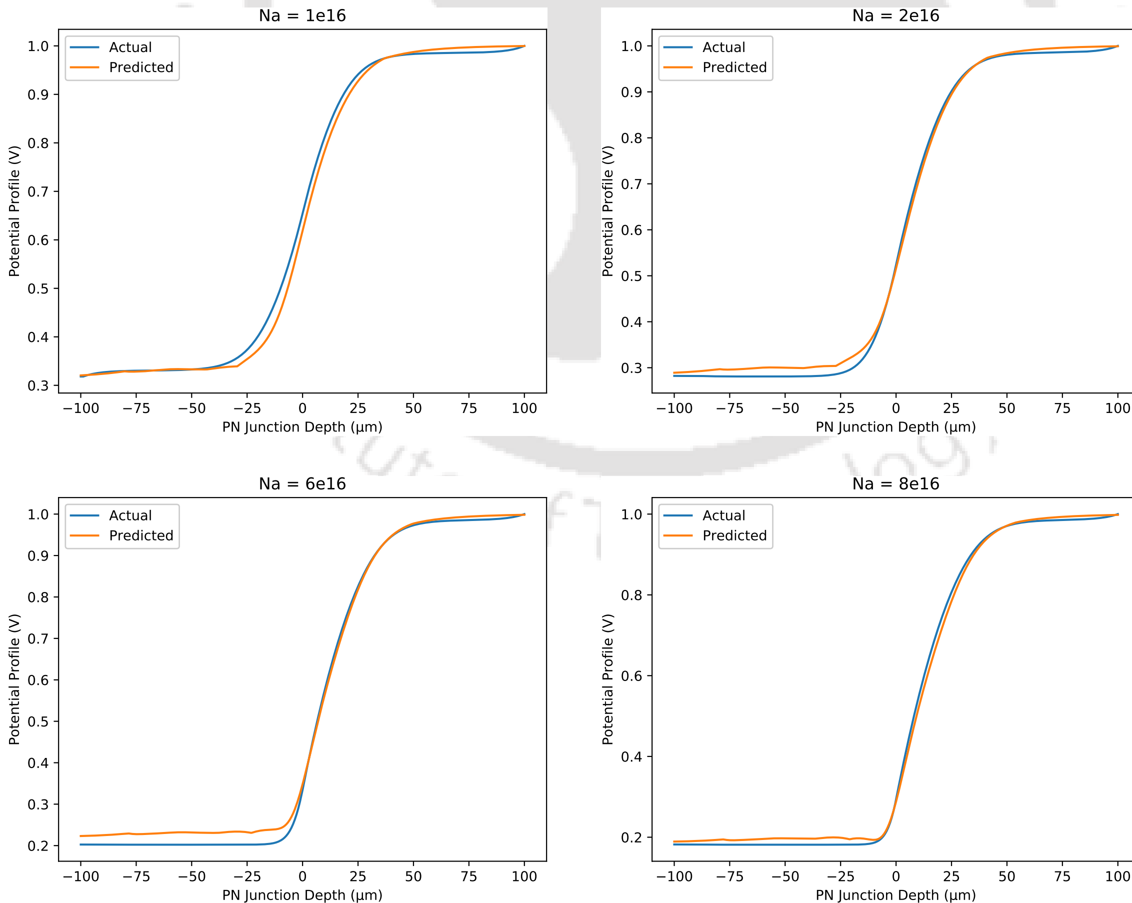


Figure 5.19: Comparison of potential profile at the depletion region for different doping concentrations

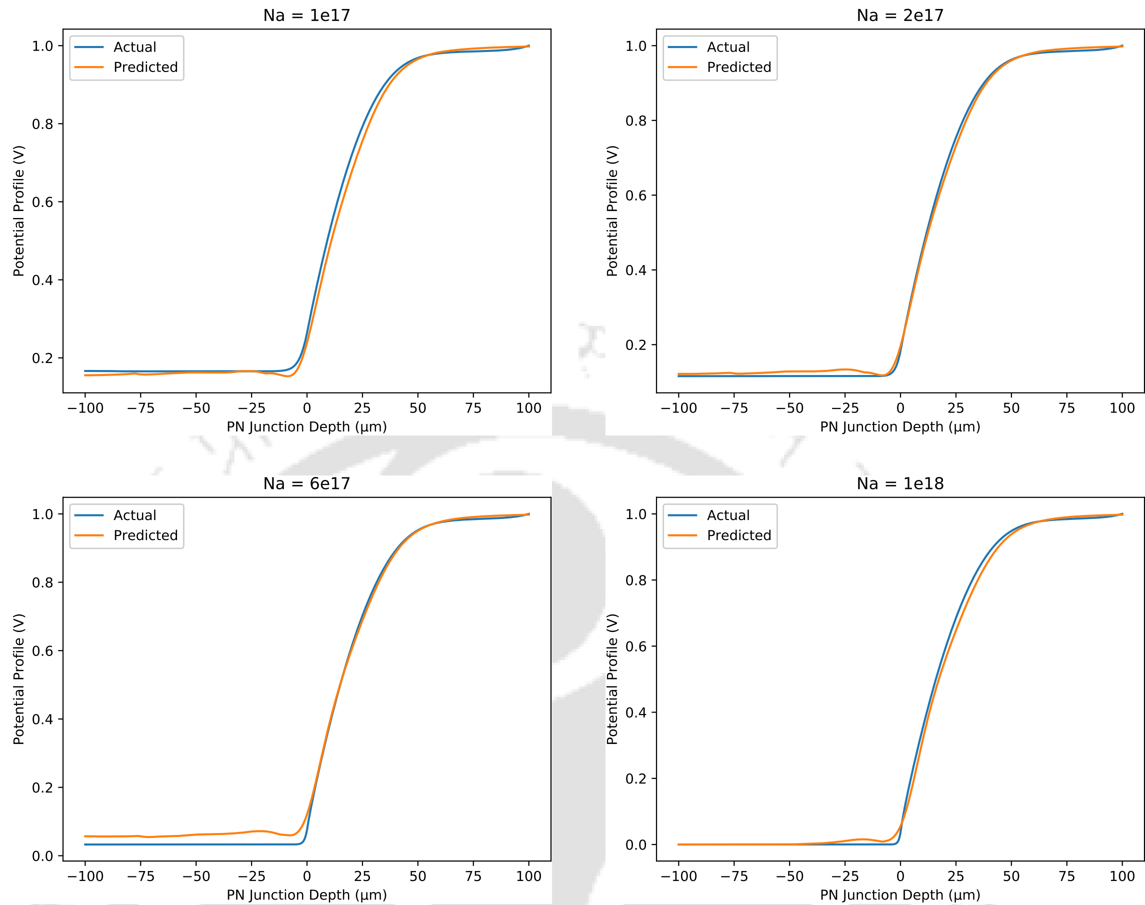


Figure 5.20: Comparison of potential profile at the depletion region for different doping concentrations

It can be observed that the predicted potential profile obtained from ANN model is very close to the actual outcome, therefore, it can be used as an initial guess for the analysis. This improved initial guess helps in speeding up the computation process by achieving faster convergence. It is observed that a minimum speedup of $1.86\times$ (13 iterations to 7 iterations) and maximum speedup of $3.5\times$ (21 iterations to 6 iterations) for 12 different doping concentrations are obtained using our proposed methodology. This is to be noted that the proposed methodology is successfully integrated with our proposed TCAD analyzer, VEDA transforming it to VEDA-ML.

5.4 Circuit Analysis Using Machine Learning

A novel and intuitive use of machine learning for the analysis of semiconductor devices further intrigues us to extend it for the analysis of electrical circuits. A similar deep neural network (DNN)

TH-2607_126102004

technique is employed to create a model for the output characteristics ($I_{DS} - V_{DS}$) of a MOSFET. DNN can be trained using data generated during TCAD analysis or characterization data collected after fabrication of the device. We prefer to use characterization data of a semiconductor device (in our case, n-channel MOSFET) to train DNN to be used for circuit analysis. The DNN based n-channel MOSFET model is employed to be used in the analysis of electrical circuits. The n-channel MOSFET is fabricated with 180nm Bulk CMOS technology available with SemiConductor Laboratory (SCL), Chandigarh India. The post-fabrication characterization data is provided by SCL Chandigarh to conduct the proposed study. Using the given dataset, a DNN model is trained to produce error $\leq 5\%$, and the corresponding simulation results for different parametric variations are presented in Fig. 5.21.

Commercially available circuit analyzer, *Spectre* is also employed to generate a huge dataset for four different parameters namely, drain-to-source voltage (V_{DS}), gate-to-source voltage (V_{GS}), gate length (L), and gate width (W) based on the following equation given below. It can also be stated that I_{DS} is a function of V_{GS}, V_{DS}, W and L, hence these four parameters are considered to populate the dataset using Process Design Kits (PDKs) provided by SCL Chandigarh. It also verifies correctness of the DNN based n-channel MOSFET model developed using characterization data.

$$I_{DS} = \frac{1}{2}\mu_n C_{ox}\frac{W}{L}\left[(V_{GS} - V_T)V_{DS} - V_{DS}^2\right] \tag{5.13}$$

$$I_{DS} = f(W, L, V_{GS}, V_{DS}) \tag{5.14}$$

The predicted results employing DNN model are compared with their actual solution using *Spectre* and SCL's PDKs for different values of V_{GS}. It can be observed that the predicted results are very close to their actual values and has computational error $< 2\%$. The main advantage of this proposed model lies in its ability to model all the dependent variables or parameters in a single equation based on DNN. This particular DNN model for the output characteristics of a MOSFET can be employed to harness its advantages for circuit simulation. The output characteristics of the MOSFET using proposed DNN model is shown in Fig. 5.22.

For the analysis of circuits having our proposed DNN based semiconductor device model, a suit-

TH-2607_126102004

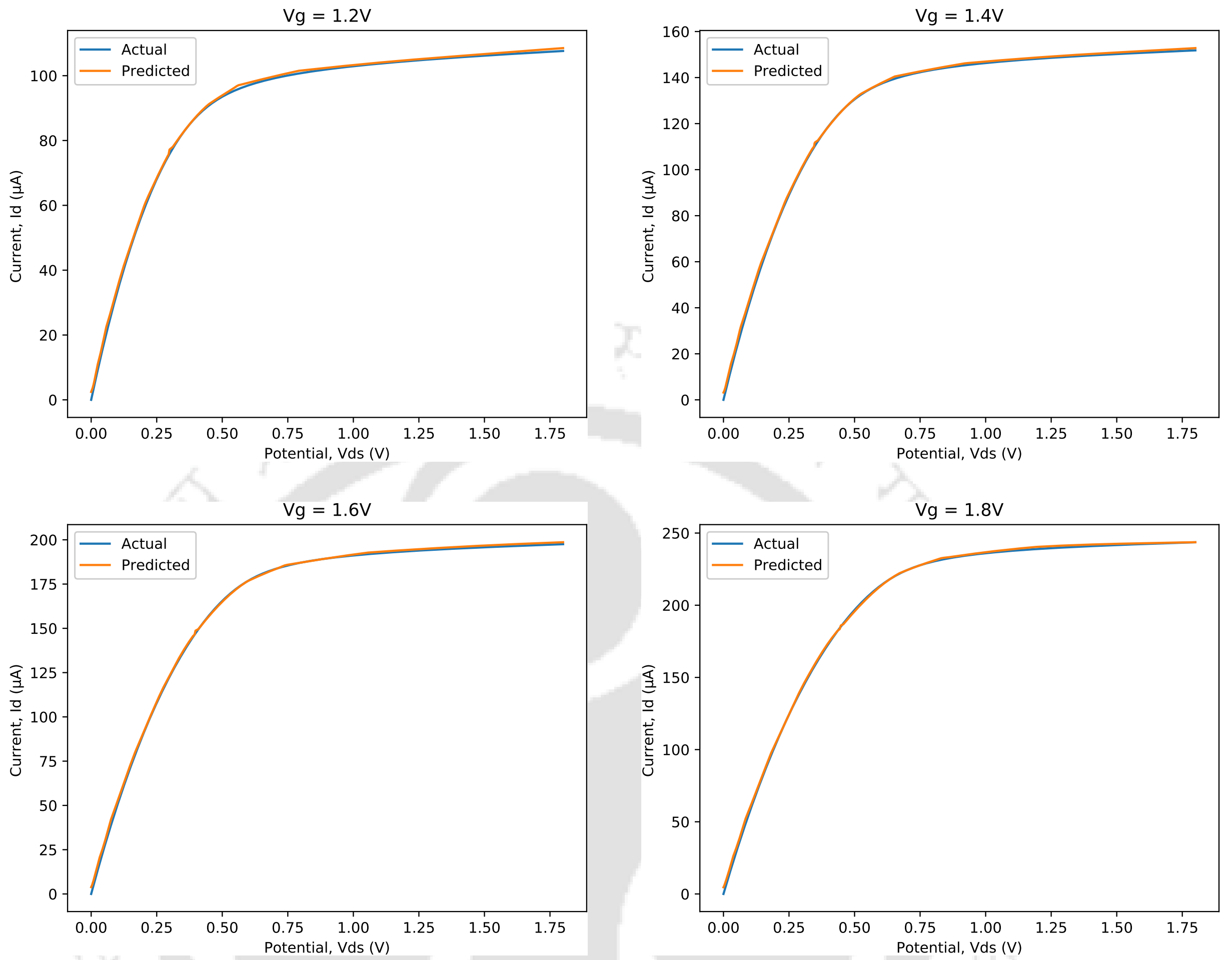


Figure 5.21: Comparison of Id-Vds of a MOSFET for different Vg.

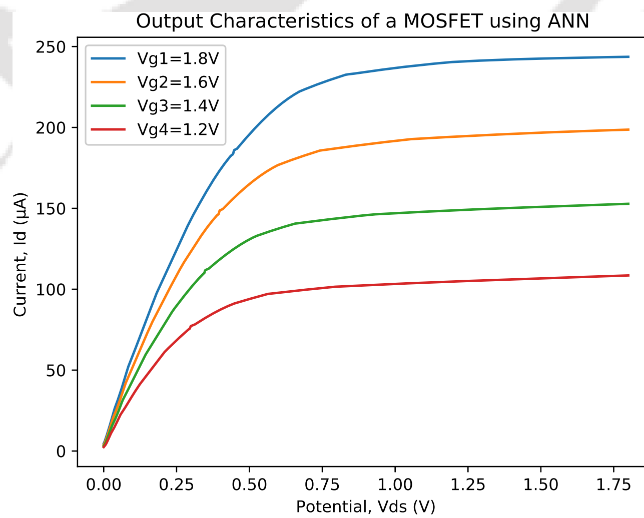


Figure 5.22: Output Characteristics of a MOSFET using ANN.

TH-2607_126102004

able circuit analysis method such as nodal analysis, modified nodal analysis etc. may be employed. To illustrate the process, a simple circuit having MOSFET as its element is shown in Fig. 5.23.

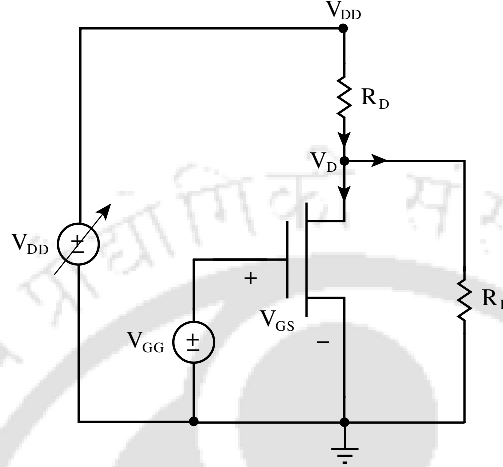


Figure 5.23: A simple MOSFET circuit.

In the proposed methodology, point relaxation based technique [59] is employed to analyze circuits composed of DNN based semiconductor devices. In this method, KCL equations are formulated on all the respective nodes and the following equation is produced.

$$\frac{V_{DD} - V_D}{R_D} = I_{DS} + \frac{V_D - 0}{R_L} \tag{5.15}$$

It should be noted that I_{DS} is considered as a function of four dependent parameters, and the proposed DNN based semiconductor device model is utilized to compute its value. The above mentioned equation can be analyzed by employing any iterative solver. We utilize *Gauss Seidel* based solution scheme to address this issue.

$$\frac{V_{DD} - V_D^{m+1}}{R_D} = I_{DS}(W, L, V_{GS}, V_{DS}^m) + \frac{V_D^{m+1} - 0}{R_L} \tag{5.16}$$

$$V_D^{m+1}\left(\frac{1}{R_D} + \frac{1}{R_L}\right) = \frac{V_{DD}}{R_D} - I_{DS}(W, L, V_{GS}, V_{DS}^m) \tag{5.17}$$

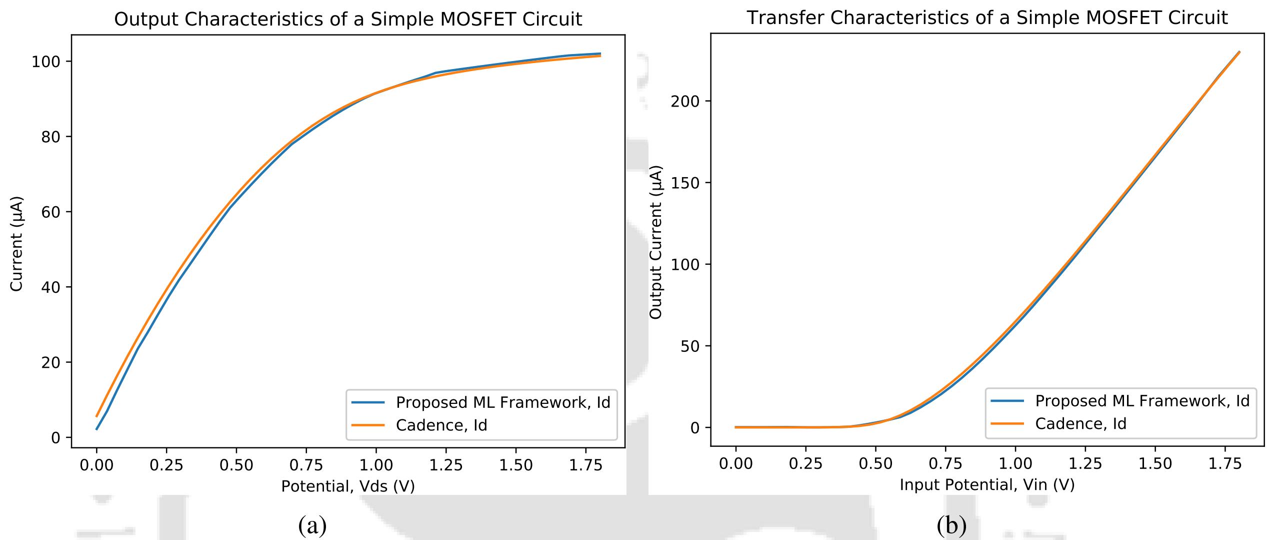
$$V_D^{m+1} = \frac{\frac{V_{DD}}{R_D} - I_{DS}(W, L, V_{GS}, V_{DS}^m)}{\frac{1}{R_D} + \frac{1}{R_L}} \tag{5.18}$$

where m represents the number of iterations.

Equation 5.18 is analyzed using *Gauss Seidel* method to produce the final solution. Fig. 5.24

TH-2607_126102004

presents the comparison of output and transfer characteristics of a simple MOSFET circuit shown in Fig. 5.23. It can be observed that the solution provided by using our proposed methodology is comparable in terms of accuracy with the solution produced by commercially available circuit analyzer *Spectre*, therefore, validating correctness of the proposed methodology.



(a) (b)

Figure 5.24: Comparison of different characteristics of a simple MOSFET circuit (a) Output Characteristics (b) Transfer Characteristics

The DNN based n-channel MOSFET (NMOS) model is utilized to analyze different electrical circuits to validate its robustness. Fig 5.25 presents a cascaded MOSFET circuit to be tested using our proposed methodology. Its transfer characteristics are obtained using our proposed circuit analyzer and its comparison with the commercially available circuit analyzer *Spectre* is presented in Fig. 5.26. The accuracy of the solutions provided by both the frameworks are acceptable validating correctness of the proposed methodology.

For analyzing a CMOS circuit, it becomes imperative to create a similar model for p-channel MOSFET (PMOS) as well. Therefore, the same procedure is followed for p-channel MOSFET and a similar DNN based semiconductor device model is created by utilizing characterization data provided by SCL Chandigarh.

The proposed DNN based device models for NMOS and PMOS are employed to analyze a CMOS inverter to test its accuracy and correctness. Fig. 5.27 presents transfer characteristics of a CMOS inverter and compares it with the results obtained using *Spectre*. It can be observed that the results

TH-2607_126102004

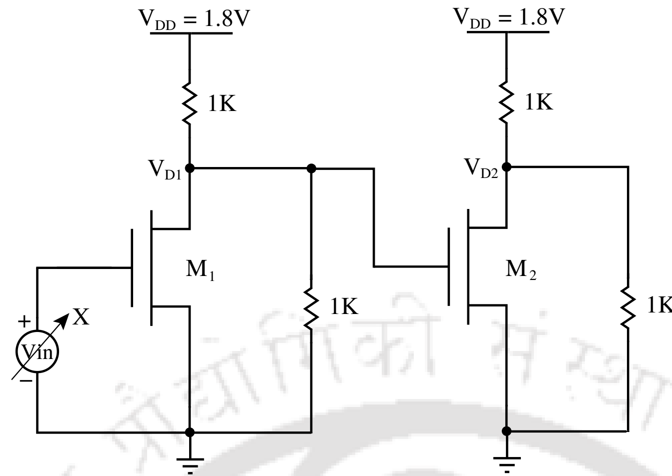


Figure 5.25: A simple cascaded MOSFET circuit

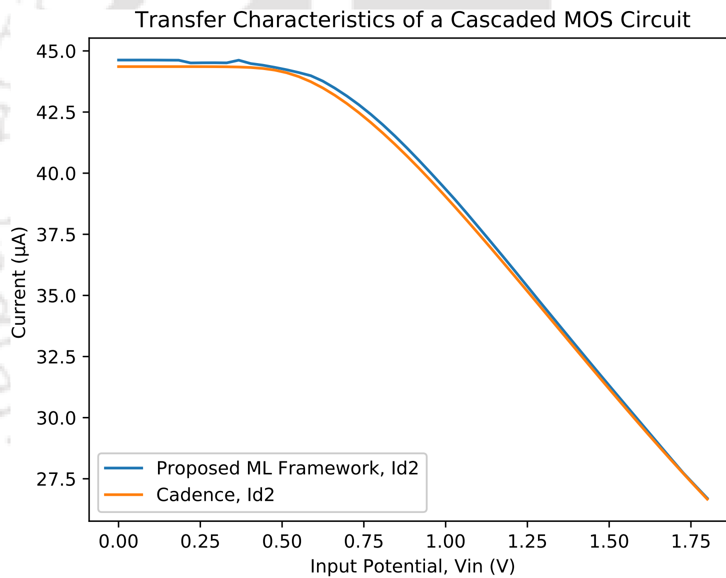


Figure 5.26: Comparison of transfer characteristics of a simple Cascaded MOS circuit

obtained using our proposed circuit analyzer having machine learning capabilities are within $< 2\%$ of the solution produced by commercially available circuit simulator *Spectre*.

The proposed methodology is employed to analyze a *Current Mirror* circuit to further validate its accuracy and effectiveness. This circuit is chosen to explore feasibility of applying our proposed methodology in other domains of electrical circuit design. Fig. 5.28 shows a simple current mirror circuit and is analyzed by using *Spectre* and our circuit simulator having machine learning capabilities. The output characteristics obtained by both the circuit analyzers are presented in Fig. 5.29. It can be

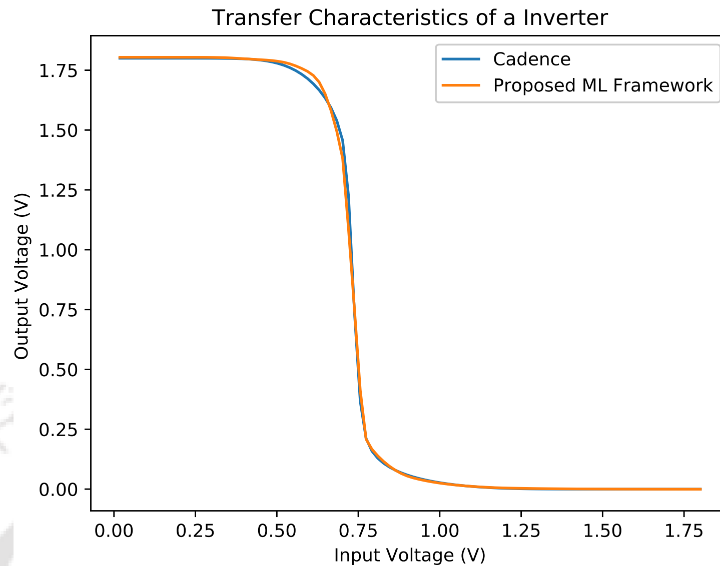


Figure 5.27: Comparison of transfer characteristics of a CMOS Inverter.

observed that the solution provided by both the frameworks are comparable which further validates accuracy of the proposed model.

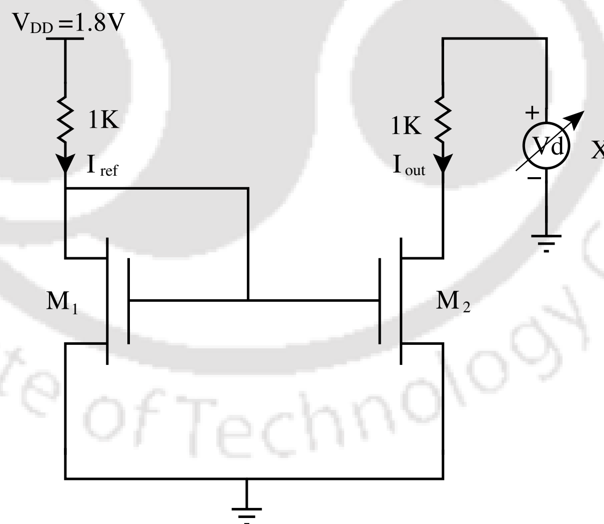


Figure 5.28: A current mirror circuit.

Apart from the accuracy of the proposed DNN based semiconductor device model for circuit analysis, its main advantage is minimal memory utilization and less computation time as compared to the lookup table based circuit analysis method. This point can be validated by providing the fact that our proposed model evaluates a mathematical expression to compute I_{DS}, whereas lookup table

TH-2607_126102004

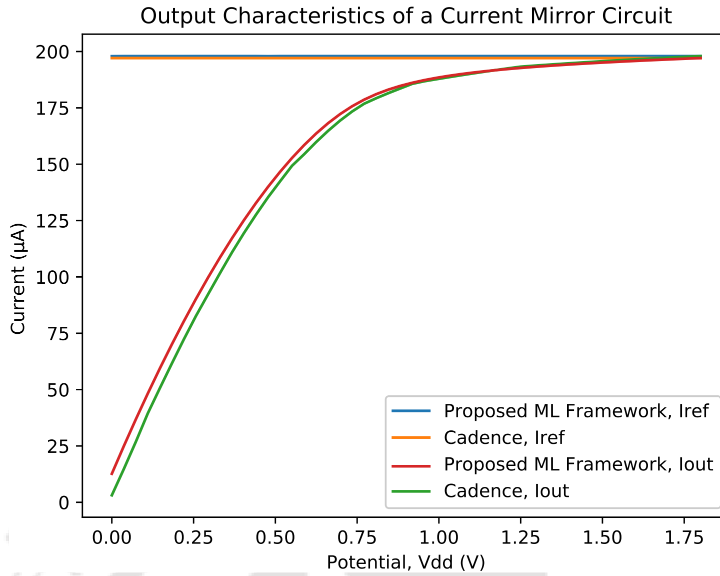


Figure 5.29: Comparison of output characteristics of a current mirror circuit.

method needs to store predetermined data points explicitly in order to find I_{DS} at intermediate data points which utilizes huge memory and time to compute I_{DS} at intermediate points.

The proposed methodology of creating a DNN based semiconductor device model for circuit analysis is also helpful in analyzing circuits based on novel devices where fundamentals of device modeling is yet to be explored and a straight forward analysis using device compact models is not feasible. Therefore, the work presented in this thesis is envisaged to play a key role in the paradigm of novel device and the machine learning based analysis of electrical circuits composed of novel devices.

5.5 Summary

In this chapter, we present an indigenous TCAD analyzer, VEDA with parallelization capabilities, which incorporates state of the art solution methodologies as compared to the currently available commercial simulators. The proposed framework uses random walk method and machine learning based methodologies to improve computation time of the analysis. Random walk improves computation time of the analysis by providing approximate initial guess of the solution and enables VEDA to achieve faster convergence to the final solution in less number of iterations. In the machine learning model, artificial neural networks are introduced to the solution process to predict potential profile

of a semiconductor device. This predicted potential profile is inducted to the solution process as an initial guess while analyzing self-consistent fundamental device equations. Due to this, the minimum speedup of $1.86\times$ (13 iterations to 7 iterations) and maximum speedup of $3.5\times$ (21 iterations to 6 iterations) for 12 different doping concentrations are achieved. The machine learning based device analysis methodology is incorporated in VEDA for the general purpose use. It is is further applied to the electrical circuit analysis, in which machine learning based models of NMOS and PMOS are utilized during analysis of electrical circuits. The proposed methodology is validated by designing various circuits and their solution is compared with the commercially available circuit simulator, *Spectre*. It is to be noted that the proposed machine learning algorithms based circuit analysis provides an edge in terms of memory utilization and computation time over lookup-table based approach of circuit analysis.

TH-2607_126102004

TH-2607_126102004

6

Conclusion and Future Aspects

TH-2607_126102004

The work presented in this thesis is dedicated to the design and development of a fast and accurate framework for the analysis of semiconductor devices. The proposed framework is extended to analyze electrical circuits composed of artificial neural network based semiconductor devices. The motivation to design an ideal framework for the analysis of semiconductor devices or TCAD environment enables us to develop indigenous TCAD engine, *VEDA (Very Efficient Device Analyzer)*. The crucial role of TCAD in the development of the electronics industry in the 21st century is the driving force because of enormous cost involved in the development of electronic circuits at lower technology nodes. At its core, a TCAD framework bundles a set of complex mathematical equations in an attempt to replicate the electrical behavior of a semiconductor device through simulations. The two main factors affecting solution of a TCAD framework can be described as, 1) The choice of transport model employed for the analysis, and 2) Different techniques to find an accurate numerical solution of a given transport model efficiently in terms of memory usage and computation time. The scope of this thesis focuses to explore different methods which can provide the best possible numerical solution to a given transport model in an efficient manner. In the pursuit of this, fundamental transport models, *Drift-Diffusion* and *Schrödinger-Poisson Solver* are employed for the development of our proposed framework.

In this thesis, a detailed description of TCAD methods employed in the analysis of semiconductor devices is presented. These methods can be broadly classified to include two main aspects of a TCAD framework, *Accuracy* and *Computation Time* required for the analysis. The work presented in this thesis aims to address both these issues as efficiently as possible. Accuracy aspect of a TCAD is resolved by studying different discretization methods, meshing techniques, and numerical methods targeting minimum error to be produced during analysis of semiconductor devices. Several discretization methods, such as finite difference method (FDM), finite element method (FEM) and finite volume method (FVM) are explored for the analysis of semiconductor devices. It is observed that FEM is more suitable for the analysis of semiconductor device with unstructured geometries including novel semiconductor devices such as FinFET, GAAFET (Gate All Around FET), etc. Therefore, FEM is implemented as a basic discretization method in VEDA along with its two variants, DG-FEM (Discontinuous Galerkin FEM) and SUPG (Streamline Upwind Petrov Galerkin). It is observed that DG-FEM and SUPG stabilization technique improve accuracy and reduce computation time required

TH-2607_126102004

for the analysis of semiconductor devices. In DG-FEM, the basis functions employed for the discretization of partial differential equations are discontinuous piecewise polynomials, whereas in FEM it is a continuous form of basis function. A discontinuous basis function helps in conserving the flux more efficiently and are capable of handling complex geometries. SUPG method is observed to be better than the classical Scharfetter-Gummel method to solve a convection-diffusion equation. The dependency of FEM on mesh generation for discretization leads to a certain amount of numerical error produced during semiconductor device analysis. It motivates us to incorporate meshfree variant of FEM, EFG (Element Free Galerkin Method) for the analysis of semiconductor devices. EFG method employs a scattered set of nodes throughout the geometry to formulate linear algebraic equations using fundamental partial differential equations which represent semiconductor device transport model. The solution provided by our proposed EFG implementation is compared with other discretization techniques including the methodology adopted by commercial TCAD simulator, *Sentaurus*. The accuracy of the solution obtained by EFG based proposed methodology is $\approx 10\times$ better as compared to the solution provided by *Sentauras*.

The need of highly accurate solution of TCAD analysis imposes a penalty in terms of increased computation time. Therefore, various aspects of TCAD to improve computation time for the analysis are studied. Three different approaches viz. parallel computing, numerical analysis and initial guess, are aimed to improve computation time of device analysis. In order to address these issues, OpenMP, MPI and CUDA based platforms are employed to achieve high performance during semiconductor device analysis. A number of efficient numerical solvers, such as LU Decomposition, Conjugate Gradient, Gauss-Seidel, etc. are incorporated in VEDA to improve computational efficiency of the proposed framework. One of the major contribution of this thesis is to provide improved initial guess for the analysis of semiconductor devices, which eventually reduces computation time by achieving faster convergence. *Random Walk* and *Machine Learning* based methods are introduced in VEDA to provide improved initial guess during the analysis. The random walk method is applied on the equivalent electrical networks of PDEs representing transport model of semiconductor devices. Therefore, equivalent electrical circuits of these PDEs are also formulated in this thesis. Further, random walk method is incorporated in the proposed framework VEDA and it provides upto 20% faster solution

TH-2607_126102004

with the initial guess having maximum 4% inaccuracy as compared to the final solution.

The success in improving solution accuracy and reducing computation time of semiconductor devices analysis motivate us to explore machine learning based algorithms to further speed up the computation time while maintaining the accuracy. Therefore, various machine learning algorithms, such as decision tree, random forest, support vector machine, artificial neural network, deep neural network, etc., are studied so that an efficient methodology can developed to estimate a very good initial guess. In this thesis, we propose a process flow which incorporates machine learning based methods to overall improve semiconductor device analysis. It was observed that deep neural network provides the best possible initial guess for the TCAD analysis. The obtained results exhibit a minimum speedup of $1.86\times$ (13 iterations to 7 iterations) and maximum speedup of $3.5\times$ (21 iterations to 6 iterations) as compared to the sequential TCAD analysis. This also motivates us to use DNN based semiconductor device models instead of developing semiconductor device models for the analysis of electrical circuits. This further inspires us to develop an electrical circuit analysis framework to validate our proposed methodology and the use of machine learning based semiconductor device models. This leads us to build above mentioned framework which can make use of the machine learning based semiconductor device models. Various electrical circuits of different domains are designed using our proposed circuit simulator and the results obtained using this simulator are compared with the commercially available circuit analyzer *Spectre*. It is observed that the results provided by both the simulators are in good agreement tagging our proposed simulator an ideal platform for the analysis of future semiconductor devices. It should be noted that the proposed methodology provides an edge in terms of less memory utilization with reduced computation time over lookup table based circuit analysis techniques.

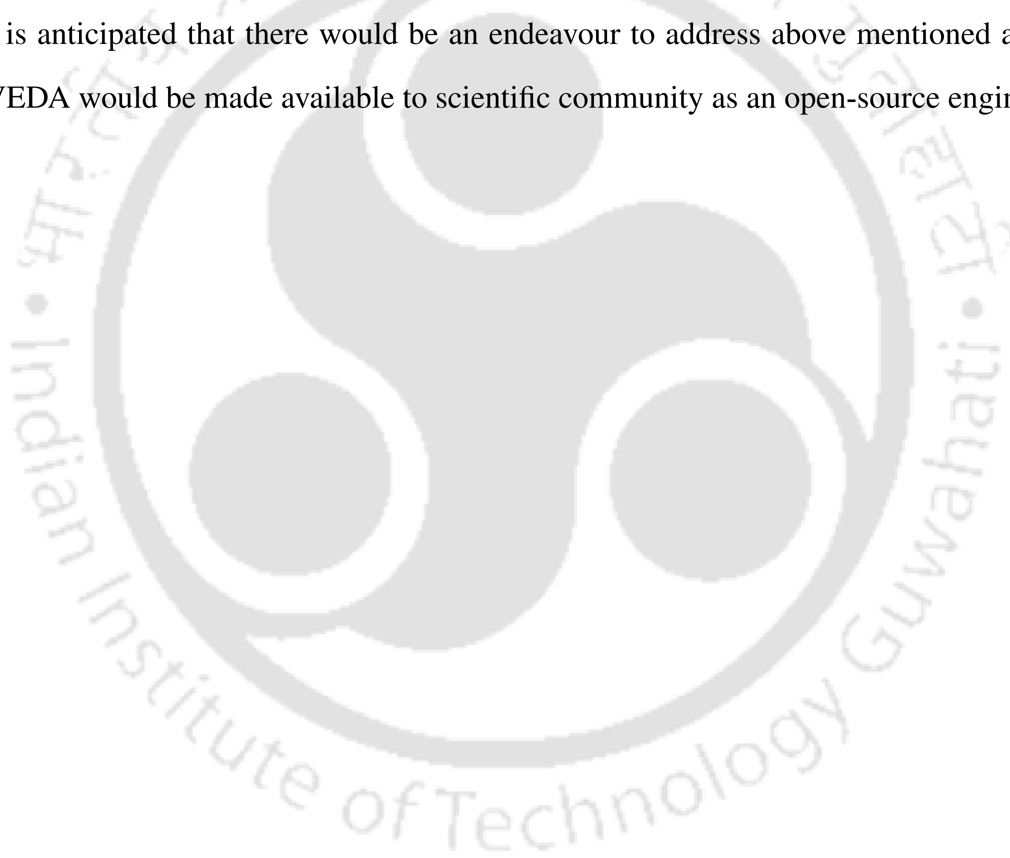
Future aspects of the work presented in this thesis may include the following directions.

- Incorporate various transport models of the semiconductor analysis, such as Hydrodynamics Model, Thermodynamics Model, Ballistic Model, etc.
- Incorporate the analysis of 2D devices, such as Graphene, Silicene, MOS_2 using the proposed methods

TH-2607_126102004

- Explore different paradigms of high performance computing to harness as much resource utilization as possible to further reduce computation time of the proposed methodologies
- Use machine learning algorithms to create good predictive models of novel semiconductor devices devices based on 2D materials to study effect of parametric variations on the performance of electrical circuits composed of novel devices in the early stage of the device development
- Perform transient analysis of analog/RF circuits using machine learning based algorithms

It is anticipated that there would be an endeavour to address above mentioned aspects in future and VEDA would be made available to scientific community as an open-source engine.



TH-2607_126102004

TH-2607_126102004

Bibliography

[1] D. Vasileska, S. M. Goodnick, and G. Klimeck, *Computational Electronics: Semiclassical and Quantum Device Modeling and Simulation*. CRC press, 2017.

[2] *Sentaurus Device User Guide, Version K-2015.06*. Synopsys, Santa Clara, June 2015.

[3] S. Selberherr, *Analysis and Simulation of Semiconductor Devices*. Springer-Verlag Wien, 1984.

[4] M. Pourfath, V. Sverdlov, and S. Selberherr, "Transport Modeling for Nanoscale Semiconductor Devices," in *2010 10th IEEE International Conference on Solid-State and Integrated Circuit Technology*, Nov 2010, pp. 1737–1740.

[5] G. Brown and B. Lindsay, "The Numerical Solution of Poisson's Equation for Two-Dimensional Semiconductor Devices," *Solid-State Electronics*, vol. 19, no. 12, pp. 991 – 992, 1976.

[6] D. Vasileska, D. Mamaluy, H. Khan, K. Raleva, and S. Goodnick, "Semiconductor Device Modeling," *Journal of Computational and Theoretical Nanoscience*, vol. 5, no. 6, pp. 999–1030, 6 2008.

[7] A. Mauri, A. Bortolossi, G. Novielli, and R. Sacco, "3D Finite Element Modeling and Simulation of Industrial Semiconductor Devices including Impact Ionization," *Journal of Mathematics in Industry*, vol. 5, no. 1, p. 1, Mar 2015.

[8] J. J. Barnes and R. J. Lomax, "Finite-Element Methods in Semiconductor Device Simulation," *IEEE Transactions on Electron Devices*, vol. 24, no. 8, pp. 1082–1089, Aug 1977.

[9] G. Evans, J. Blackledge, and P. Yardley, *Numerical Methods for Partial Differential Equations*, ser. Springer undergraduate mathematics series. Springer, 2000.

[10] J. T. Oden, "An Introduction to the Finite Element Method with Applications to Nonlinear Problems (R. E. White)," *SIAM Review*, vol. 31, no. 3, pp. 512–512, 1989.

[11] H. Ren and Y. Cheng, "The Interpolating Element-Free Galerkin (IEFG) Method for Two-Dimensional Potential Problems," *Engineering Analysis with Boundary Elements*, vol. 36, no. 5, pp. 873–880, 2012.

[12] J. Dolbow and T. Belytschko, "Numerical Integration of the Galerkin Weak Form in Meshfree Methods," *Computational Mechanics*, vol. 23, no. 3, pp. 219–230, 1999.

[13] T. Belytschko, Y. Y. Lu, and L. Gu, "Element-Free Galerkin Methods," *International Journal for Numerical Methods in Engineering*, vol. 37, no. 2, pp. 229–256, 1994.

[14] D. Neamen, *Semiconductor Physics And Devices*, 3rd ed. New York, NY, USA: McGraw-Hill, Inc., 2003.

[15] A. R. Firoozjaee, E. Hendi, and F. Farvizi, "Element Free Gelerkin Method for 2-D Potential Problems," *Applied Mathematics*, vol. 6, no. 01, p. 149, 2015.

TH-2607_126102004

[16] H. Qian, S. R. Nassif, and S. S. Sapatnekar, "Power Grid Analysis using Random Walks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 8, pp. 1204–1224, Aug 2005.

[17] J. Yang, Y. Cai, Q. Zhou, and J. Shi, "Fast Poisson Solver Preconditioned Method for Robust Power Grid Analysis," in *2011 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov 2011, pp. 531–536.

[18] T. Mitchell, *Machine Learning*, ser. McGraw-Hill International Editions. McGraw-Hill, 1997.

[19] J. Kelleher, B. Namee, and A. D'Arcy, *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*, ser. The MIT Press. MIT Press, 2015.

[20] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*, ser. Springer Series in Statistics. Springer New York, 2009.

[21] Jun Guo, Sheqin Dong, and S. Goto, "Random Walk Algorithm for Large Thermal RC Network Analysis," in *2009 IEEE 8th International Conference on ASIC*, Oct 2009, pp. 771–774.

[22] Y. Shi, Wei Yao, J. Xiong, and L. He, "Incremental and On-Demand Random Walk for Iterative Power Distribution Network Analysis," in *2009 Asia and South Pacific Design Automation Conference*, Jan 2009, pp. 185–190.

[23] S. Dash, V. Bangera, V. B. Y. Kumar, G. Trivedi, and S. B. Patkar, "Parallel Two Step Random Walk Algorithm to Analyze VLSI Power Grid Networks," in *2015 19th International Symposium on VLSI Design and Test*, June 2015, pp. 1–2.

[24] H. Qian and S. Sapatnekar, "A Hybrid Linear Equation Solver and its Application in Quadratic Placement," in *Proceedings of theICCAD-2005*, ser. IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD, 12 2005, pp. 904–908.

[25] V. Ladogubets, O. Beznosyk, and O. Finogenov, "Presentation of a System of Ordinary Differential Equations as an Equivalent Electrical Circuit," in *2010 Proceedings of VIth International Conference on Perspective Technologies and Methods in MEMS Design*, April 2010, pp. 116–120.

[26] A. Ferikoglu, Y. Sari, and R. Koker, "Simulating Solutions of Linear Differential Equations using various Active Circuits," in *2010 XIth International Workshop on Symbolic and Numerical Methods, Modeling and Applications to Circuit Design (SM2ACD)*, Oct 2010, pp. 1–4.

[27] Y. D. Save, H. Narayanan, and S. B. Patkar, "Solution of Partial Differential Equations by Electrical Analogy," *Journal of Computational Science*, vol. 2, no. 1, pp. 18 – 30, 2011.

[28] Y. D. Save, H. Narayanan, and S. B. Patkar, "Two Graph Based Circuit Simulator for PDE-Electrical Analogy," in *2012 25th International Conference on VLSI Design*, Jan 2012, pp. 400–405.

[29] F. Adly, P. D. Yoo, S. Muhaidat, Y. Al-Hammadi, U. Lee, and M. Ismail, "Randomized General Regression Network for Identification of Defect Patterns in Semiconductor Wafer Maps," *IEEE Transactions on Semiconductor Manufacturing*, vol. 28, no. 2, pp. 145–152, May 2015.

[30] G. Tello, O. Y. Al-Jarrah, P. D. Yoo, Y. Al-Hammadi, S. Muhaidat, and U. Lee, "Deep-Structured Machine Learning Model for the Recognition of Mixed-Defect Patterns in Semiconductor Fabrication Processes," *IEEE Transactions on Semiconductor Manufacturing*, vol. 31, no. 2, pp. 315–322, May 2018.

TH-2607_126102004

[31] K. Nakata, R. Orihara, Y. Mizuoka, and K. Takagi, "A Comprehensive Big-Data-Based Monitoring System for Yield Enhancement in Semiconductor Manufacturing," *IEEE Transactions on Semiconductor Manufacturing*, vol. 30, no. 4, pp. 339–344, Nov 2017.

[32] Z. Marinkovi, G. Crupi, D. M. M. . Schreurs, A. Caddemi, and V. Markovi, "Neural Procedure for Microwave MOSFET Modelling versus Bias and Gate Length," in *2017 13th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS)*, Oct 2017, pp. 166–169.

[33] A. F. Abo-Elhadeed, "Modeling Ballistic Double Gate MOSFETs using Neural Networks Approach," in *Proceedings of the 8th Spanish Conference on Electron Devices, CDE'2011*, Feb 2011, pp. 1–4.

[34] M. Abtin, P. Keshavarzi, K. Jaferzadeh, and A. Naderi, "Modeling Double Gate FinFETs by using Artificial Neural Network," in *2010 IEEE International Conference on Semiconductor Electronics (ICSE2010)*, June 2010, pp. 38–40.

[35] J. Cai, W. Zhou, J. Liu, C. Yu, L. Sun, and S. Chen, "A Global Modeling Technique for InP HBT Based on Machine Learning Method," in *2018 Asia-Pacific Microwave Conference (APMC)*, Nov 2018, pp. 375–377.

[36] *Virtuoso Spectre Circuit Simulator User Guide, Product Version 15.1.* Cadence, October 2015.

[37] R. A. Thakker, C. Sathe, M. S. Baghini, and M. B. Patil, "A Table-Based Approach to Study the Impact of Process Variations on FinFET Circuit Performance," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 29, no. 4, pp. 627–631, April 2010.

[38] R. A. Thakker, C. Sathe, A. B. Sachid, M. Shojaei Baghini, V. Ramgopal Rao, and M. B. Patil, "A Novel Table-Based Approach for Design of FinFET Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 7, pp. 1061–1070, July 2009.

[39] D. H. Cho and S. M. Kang, "An Accurate AC Characteristic Table Look-up Model for VLSI Analog Circuit Simulation Applications," in *1993 IEEE International Symposium on Circuits and Systems*, May 1993, pp. 1531–1534 vol.3.

[40] S. Datta, *Nanoscale Device Modeling: the Greens Function Method, Superlattices and Microstructures*, 2000, vol. 28(4).

[41] N. Arora, *MOSFET Models for VLSI Circuit Simulation: Theory and Practice.* Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1993.

[42] K. Tomizawa, *Numerical Simulation of Submicron Semiconductor Devices*, ser. Artech House materials science library. Artech House, 1993.

[43] J. Dolbow and T. Belytschko, "An Introduction to Programming the Meshless Element Free Galerkin Method," *Archives of Computational Methods in Engineering*, vol. 5, no. 3, pp. 207–241, 1998.

[44] G.-R. Liu and Y.-T. Gu, *An Introduction to Meshfree Methods and their Programming.* Springer Science & Business Media, 2005.

[45] C. K. C. Wu and M. E. Plesha, "Essential Boundary Condition Enforcement in Meshless Methods: Boundary Flux Collocation Method," *International Journal for Numerical Methods in Engineering*, vol. 53, no. 3, pp. 499–514, 2002.

[46] L. Semin, "Collocation Method for 2D Heat Conduction Equation," *Computational Technologies*, vol. 11, no. 1, 2006.

TH-2607_126102004

[47] J. Sánchez and P. Gonçalves, "Shape Function Object Modeling for Meshfree Methods," *Mecánica Computacional*, vol. 29, pp. 4753–4767, 2010.

[48] D. Caughey and R. Thomas, "Carrier Mobilities in Silicon Empirically related to Doping and Field," *Proceedings of the IEEE*, vol. 55, no. 12, pp. 2192–2193, 1967.

[49] A. Mitchell, D. Griffiths, and G. Watson, *Numerical Analysis: A.R. Mitchell 75th Birthday Volume*. World Scientific, 1996.

[50] A. Bjorck, *Numerical Methods in Matrix Computations*, ser. Texts in Applied Mathematics. Cham: Springer, 2015.

[51] G. Strang, *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 2016.

[52] N. Mohamed and M. Sujod, "Finite Elements in Semiconductor Devices," in *Information Management and Engineering, 2009. ICIME '09. International Conference on*, April 2009, pp. 108–110.

[53] S. C. Cockburn B, Karniadakis GE, *Discontinuous Galerkin Methods: Theory, Computation and Applications*. Springer-Verlag:Berlin, 2000.

[54] J. de Frutos, B. Garca-Archilla, V. John, and J. Novo, "An Adaptive {SUPG} Method for Evolutionary ConvectionDiffusion Equations," *Computer Methods in Applied Mechanics and Engineering*, vol. 273, pp. 219 – 237, 2014.

[55] S. Giere, T. Iliescu, V. John, and D. Wells, "SUPG Reduced Order Models for Convection-Dominated Convection-Diffusion-Reaction Equations," *Computer Methods in Applied Mechanics and Engineering*, vol. 289, pp. 454 – 474, 2015.

[56] I. Tan, G. L. Snider, L. D. Chang, and E. L. Hu, "A Self-Consistent Solution of Schrodinger-Poisson Equations using a Nonuniform Mesh," *Journal of Applied Physics*, vol. 68, no. 8, 1990.

[57] Haifeng Qian, S. R. Nassif, and S. S. Sapatnekar, "Power Grid Analysis using Random Walks," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 8, pp. 1204–1224, Aug 2005.

[58] B. Boghrati and S. Sapatnekar, "A Scaled Random Walk Solver for Fast Power Grid Analysis," in *2011 Design, Automation Test in Europe*, March 2011, pp. 1–6.

[59] V. B. Y. Kumar, K. Dhiman, M. Datar, A. Pacharne, H. Narayanan, and S. B. Patkar, "Relaxation based Circuit Simulation Acceleration over CPU-FPGA," in *2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID)*, Jan 2016, pp. 409–414.

TH-2607_126102004

List of Publications

Journal Publications

- Published Papers:

1. **Gaurav Kumar**, M. Singh, B. Anand and G. Trivedi, "*A Framework to Simulate Semiconductor Devices Using Parallel Computer Architecture*", Journal of Physics: Conference Series, vol. 759, no. 1, p. 012098, 2016.
2. M. Singh, **Gaurav Kumar**, S. Bordoloi and G. Trivedi, "*A study on modeling and simulation of Multiple-Gate MOSFETs*", Journal of Physics: Conference Series, vol. 759, no. 1, p. 012093, 2016.

- Manuscripts Under Review

1. **Gaurav Kumar**, A. Singh, A. Mahanta and G. Trivedi, "*VEDA: A Highly Accurate and Adaptive Framework to Simulate Semiconductor Devices based on Element-Free Galerkin Method,*" 2018 IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.
2. **Gaurav Kumar**, S. Dash and G. Trivedi, "*A Fast and Scalable Hybrid Framework to Simulate Semiconductor Devices using Random Walk Algorithm,*" 2018 IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.
3. **Gaurav Kumar**, Ashutosh Yadav, H. S. Jatana, A. Mahanta, G. Trivedi and P. Guha, "*Analysis of Semiconductor Devices using Artificial Neural Networks and its Application to Design Analog Circuits,*" 2018 IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.

Conference and Workshop Publications

1. **Gaurav Kumar**, M. Singh, A. Ray and G. Trivedi, "*An FEM based Framework to Simulate Semiconductor Devices using Streamline Upwind Petrov-Galerkin Stabilization Technique,*"

TH-2607_126102004

2017 27th International Conference Radioelektronika (RADIOELEKTRONIKA), Brno, 2017, pp. 1-5.

2. **Gaurav Kumar**, M. Singh, G. Trivedi and D. Nandi, "*Bandgap Generation in 2D Materials,*" 2017 Devices for Integrated Circuit (DevIC), Kalyani, 2017, pp. 551-555.

3. Sridevi Gugulothu, **Gaurav Kumar**, Harshal B. Nemade,Gaurav Trivedi, "*Design of a FEM Based Simulator for MEMS Devices*", TENCON 2017 - 2017 IEEE Region 10 Conference, Penang, 2017, pp. 1670-1675.

4. S. Gugulothu, **Gaurav Kumar**, S. Kundu, H. B. Nemade and G. Trivedi, "*Design of a Next Generation Framework for MEMS Devices,*" 2017 Devices for Integrated Circuit (DevIC), Kalyani, 2017, pp. 546-550.

5. A. Ray, **Gaurav Kumar**, S. Bordoloi, D. K. Sinha, P. Agarwal, and G. Trivedi, "*FEM based Device Simulator for High Voltage Devices*", VLSI Design and Test, B. K. Kaushik, S. Dasgupta, and V. Singh, Eds. Singapore: Springer Singapore, 2017, pp. 127135.

6. **Gaurav Kumar**, M. Singh, B. Anand and G. Trivedi, "*A Parallel Device Simulator based on Finite Element Method*", 2015 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, 2015, pp. 30-35.

7. **Gaurav Kumar**, M. Singh, B. Anand and G. Trivedi, "*A Parallel Device Simulator Based on Discontinuous Galerkin Finite Element Method*", 18th International Workshop on Physics of Semiconductor Devices (IWPSD 2015), Bangalore.

8. S. Bordoloi, **Gaurav Kumar**, M. Singh, and G. Trivedi , "*A Review on Issues in Device Modeling at Liquid Nitrogen Temperature*", XXVII IUPAP Conference on Computational Physics (CCP 2015), Guwahati. (Poster)

9. A. Ray, **Gaurav Kumar**, M. Singh, P. Agarawal and G. Trivedi, "*Silicon Carbide : An emerging material for high power device applications*", XXVII IUPAP Conference on Computational Physics, (CCP 2015), Guwahati. (Poster)

TH-2607_126102004

TH-2607_126102004