WAVELET-BASED SEMICONDUCTOR DEVICE SIMULATION

BY

PUN KONG-PANG

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF PHILOSOPHY

DIVISION OF ELECTRONIC ENGINEERING

THE CHINESE UNIVERSITY OF HONG KONG

JUNE 1997

To my family and my wife, Wah Fung

Acknowledgement

First of all, I would like to thank my thesis advisor, Prof. Fung-Yuel Chang, who has illustrated to me what it means to do brilliant research, and has shown great enthusiasm on educating. I thank Prof. Chang for giving the direction of my research, many valuable ideas and invaluable encouragement throughout the research. Beyond these, he has also given me many worthy advises on my life.

I learned wavelets from Prof. Chung-Chi Hsu, and solid state theory from Prof. Sai-Peng Wong. I would take this opportunity to express my deep thanks to them. Furthermore, I thank Prof. Philip Chan for his appreciation and constructive commands on this thesis.

I thank my colleagues, Yuen Pat Lau, Kelvin Cheung, Johnson Pang, Vincent Sit, Frankie Cheng, Ken Ma, Cheuk Him To, Kwong Yu Lai, Siu Kwong Ip, Dr. Juraj Povazanec for sharing their research experience with me, and Mr. Yeung in VLSI laboratory for the supporting of computer facilities. I thank Chung Yuk Or and Thomas Chan for their critical and constructive opinions on my research. I also thank my classmate Yick Man Tsang for working with me in hundreds of midnight.

Abstract

Wavelet-based method for one-dimensional semiconductor device simulation is developed in this thesis. Wavelet alternative is given to device simulator on two part: grid generator and numerical solver.

The physical device model adopted here is the classic drift-diffusion model. This model includes five fundamental equations: Poisson's equation, continuity equations for electron and hole, transport equations for electron and hole. After simplification and normalization, three partial differential equations (PDEs) are obtained. The device variables are electrostatic potential, quasi-Fermi potentials for electron and hole. Traditionally, the numerical solution of these partial differential equations are obtained by finite difference method, finite element method, etc. In this thesis, we use *cubic spline wavelet collocation method* for spatial discretization and finite difference method for time discretization. The concept of collocation method for PDEs is to expand unknown variable as a linear combination of a set of basis functions, then to evaluate the equation exactly at some collocation points. The set of basis functions we used is a wavelet family introduced by Wei Cai [1].

Why use wavelets? Because wavelets have multi-resolution property, and wavelet functions are localized both in spatial and frequency domains. We observed that many devices have variables varying very fast in a small region, and varying smoothly outside that region. Therefore, we could use higher resolution wavelets in that region to capture fast changing signals, and use lower resolution wavelets outside that region. This idea is implemented by deleting redundant collocation points/wavelet functions.

In the wavelet collocation method for PDEs, the collocation point number depends on the resolution level, for instance, L for scaling function space V_0, $L \times 2^j$ for wavelet space $W_j, j = 0, 1, \ldots$. However, it is not necessary to use full scale collocation points. An adaptive method is developed to delete the redundant collocation points and the corresponding basis functions, based on the localization property of wavelet functions. This collocation point generation process is equivalent to the grid generation process of other device simulators. However, this method is much attractive because the algorithm is very simple and efficient. Once the collocation points (grid) is determined, we use it for seeking the steady state and transient response of devices.

The major advantage of wavelet method for device simulation is that a large amount of wavelet basis/collocation points can be removed due to the localization property of wavelet functions, and this collocation point (grid) reduction scheme leads to a very efficient numerical solver. This method is specially good for devices with local rapid changing variables, which are very common.

In this thesis, two types of semiconductor device: p-n junction diode and bipolar transistor, as examples, are successfully simulated by this method. Part of our results was published in the proceedings of ISCAS'97 [2].

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The objectives of this research are to investigate the feasibility and find a suitable way of wavelet-based semiconductor device simulation, and to show how wavelet method is suitable for physical device simulation.

During this decade, wavelet approximations have attracted much attention as a potentially efficient numerical technique for solving partial differential equations. ([3, 4, 5]). Because of their advantageous properties of localizations in both spatial and frequency domains, wavelets seem to be a great candidate for adaptive and multi-resolution schemes to obtain solutions which vary dramatically both in space and time and develop singularities.

In the physical device simulation, it is quite often that the model develops singularity, for various reasons such as sharp changing in doping profile. The electric potential distribution in the devices usually takes sharp changing in a small region corresponding to sharp changing in charge distribution, and remains smooth outside this region. Not only does the electric potential take such phenomenon, but also do the carrier densities. It contains much more information inside this region than outside this region. A typical example is the abrupt doped p-n junction diode.

We think it should be very effective to simulate such devices by wavelet method due to its nice properties mentioned above. We can use higher resolution wavelets in a certain region to capture fast changing signals, while use lower resolution wavelets outside that region. Besides, we can develop some algorithms to determine that sharp region automatically.

To my best knowledge, we are the first group to apply wavelet method in physical device simulation.

1.1 Role of Device Simulation

Before going into the background theory and other details, we would like to describe the roles of device simulation firstly. So we can see our position clearly.

With the increasing complexity of integrated circuit fabrication processes and diminishing feature sizes of device structures, the use of computer programs have proven to be valuable aids in the development and characterization of new Integrated Circuit(IC) technologies.

The device modeling and simulation is an important and necessary stage in the whole computer aided IC design cycle. Figure 1.1 shows the simulation flow chart of computer aided IC design.

The first stage is process simulation. The process simulation deals with all aspects of IC fabrication. Given a description of the processing steps and layout geometry, process simulation determines the details of the resulting device structure, including the boundaries of various material layers of the structure and the distribution of the impurities within these layers.

Device simulation is the second stage. The output of first stage, doping profile and device geometry together with terminal voltages, are input of device simulator. Device simulator generates the device behaviors. Through some extraction program, the parameters of equivalent circuit of device is then generated, and input to the last stage, circuit simulation. One major advantage of using circuit simulator is that it is capable of carrying out large scale circuit simulation.

The benefit of using computer programs to supplement the technology development is to shorten the development cycle while reducing the development cost. During

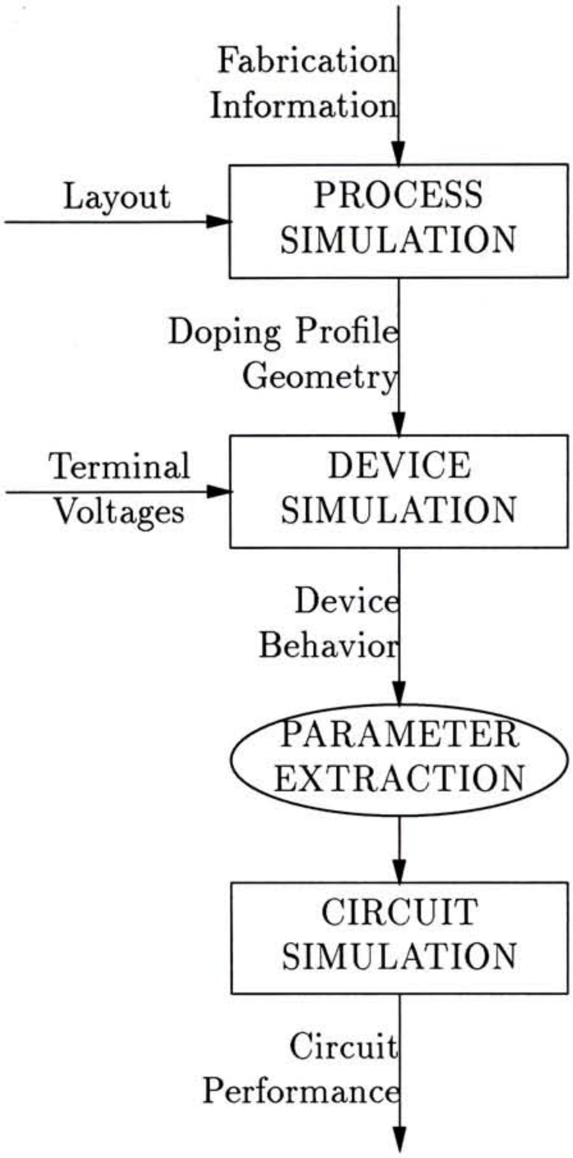


Figure 1.1: Interrelationship between Process, Device and Circuit Simulation.

the device simulation stage, the simulation speed is as important as the simulation accuracy. By applying wavelet method, simulation time should be largely reduced. It will benefit the designers who use such simulator.

1.2 Classification of Device Models

Models of semiconductor devices can be categorized into several different classes. The major divisions are shown in Figure 1.2. One of the most important distinctions is that between physical device models and circuit models. As the name suggests, circuit models employ circuit analogue for the devices, often representing the device as a "black box" there the transfer function is obtained from physical device model simulation as described before, or from measurement of an actual device. In the equivalent circuit models, the values of specific circuit elements are obtained either from measurement or by relating their values to specific physical parameters of a given device.

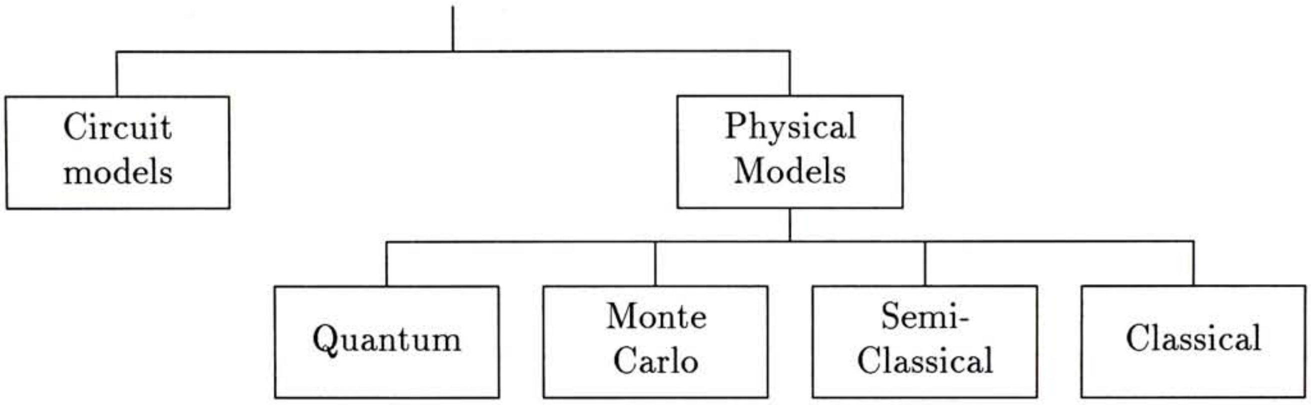


Figure 1.2: Classes of semiconductor device model

Equivalent circuit models have several advantages over physical models. First, they are relatively easy to solve. Therefore, they provide a computationally efficient means of modeling devices in applications such as large scale circuit simulation, where the use of complex models would results in an unacceptable computing requirement. However, circuit models are most often used to model devices which are already well characterized due to the limited predictive capability of them. Since it is often difficult to draw a direct relationship between the physical parameters and electrical behavior of a real device and the elements of its circuit model.

In this research, we use physical device model. This type of model utilizes a more exact representation of the device, and the important physical parameters, such as material permeability, carrier mobilities, are incorporated directly into the model. This can result in a more accurate model, assuming, as always, that any simplifying assumptions used in its formulation are carefully considered. These models can provide a greater predictive capability, because the actual device is modeled based on the fundamental physical phenomena which govern its operation, rather than a circuit representation of they external electrical characteristics of the device.

Figure 1.2 shows several sub-classes of physical models, each employing different assumptions in the formulation of the carrier transport model, and having different range of application. The applicability of a physical model to a specific device is often determined by the size of the important features of that device, for instance, the channel width of field effect transistor. Physical models generally have a lower size limit placed on them by the physical assumptions used in their formulation, and an upper size limit determined by the complexity and requirements of the solution.

Quantum mechanical models provide solutions which are directly based on the underlying quantum mechanical principles which govern electron transport in the semiconductors, and are required for very small device in which quantum effect are significant. The complexity and computer resource requirements of this type of model tends to place an upper limit to the dimension of simulated device. This limit is about a couple of hundred angstroms.

Monte Carlo models are statistical models which provide a solution for the Boltzmann transport equation, and yield a somewhat less complex formulation to describe carrier transport. These models are most often used for simulating devices in which the particulate nature of carriers is important, but in which the effects of very small scale phenomena, such as the Heisenberg uncertainty principle, becomes less significant. They are nevertheless computationally intensive, and are generally limited by avaibility of computational resources to modeling moderate scale devices with dimensions of a few tens of a micron. An important use of Monte Carlo model is for the calculations of material parameters, such as velocity-field and energy-field characteristic of carriers in bulk semiconductor.

Semi-classical models are based on an solutions for at least first three moments of the Boltzmann transport equation, and neglect the particulate nature of carriers by treating electrons and holes as a continuum. This means that the device be large enough that there are sufficient carriers so that electrons and holes can be characterized by local average quantities, such as density, energy and momentum. This is often the case of devices with dimensions above a few tens of a micron, and is true for a large number of the devices in common use today.

The simplest physical model is the classical drift-diffusion model. This model assumes that the driving forces of carriers are the drift and diffusion effect only. The classical and semi-classical models are most often used at present.

In this research, we choose drift-diffusion medel because it is the simplest and most popular one. In the subsequent chapters, we will focus on this model only. However, we should adopt more complicate model for a larger range of devices. Future research should be focused on this.

1.3 Sections of a Typical Simulator

A simulator can be broadly divided into several sections, and these are illustrated in Figure 1.3.

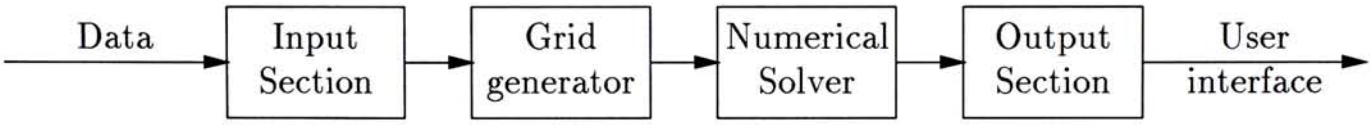


Figure 1.3: Sections of a typical simulator

First, there is some means of reading and interpreting the data which information such as the geometry of the device, the doping profiles throughout the device, the voltages which are applied to the contacts, and the selection of numerical methods.

Second, determine the topology of the numerical mesh used for the solution. In chapter 4, we will introduce an adaptive wavelet collocation method, which will generate a very efficient grid point based on self adaptive scheme. This scheme exploits the spatial localization property of wavelets to remove unnecessary collocation points(gird points).

The third section is numerical solver. It is generally the actual solution of semiconductor equations which requires the largest amount of computer resource of any part of the simulation process. The efficiency of this process is determined primarily by the method of solution, but this is often dictated by the type of solution required from the simulator, for example, whether steady state or transient solutions are required. We will introduce the wavelet collocation method to do it. To our best knowledge, we are the first one, so far the only one, to use wavelet method in device simulation. Later, we will see, this method is very suitable for those devices with abrupt doping profile or abrupt changing in other variables.

The last section is the output section. The method of presentation is an important aspect of the simulation process. In general, the results of simulations are graphic format. It is much easier to obtain information from graphs, surface and contour diagrams than from columns of numbers. However, in our research, our program will just give the data output. We concentrate on the grid generator and numerical solver only.

1.4 Arrangement of This Thesis

This thesis is arranged as follows. The first chapter is introduction. In the second chapter, the most important physical models, describing the carrier transports in semiconductor structures, the statistics and the intrinsic carrier density, the carrier mobility and the generation-recombination mechanisms are presented. The subsequent discussion of device simulation is based on this model. In addition, the limitations of this model are discussed in this chapter. In the third chapter, we will discuss various computational aspects of device simulation, including normalization and discretization of device equations, and methods of solving nonlinear algebraic systems. In the fourth chapter, the cubic spline wavelet collocation method, which is chosen to solve the differential equations, in addition with schemes of determining the collocation range, are discussed in detail. In the fifth chapter, results of device simulation by the suggested methods are presented, including a one dimensional p-n junction and n-p-n transistor. Finally, some concluding remarks are given and further research direction is pointed out.

Chapter 2

Classical Physical Model

In this chapter, we present the classical physical model, which is chosen to carry out the wavelet-based simulation algorithm. As mentioned in the previous chapter, this model is the simplest and most popular physical device model compared to Monte-Carlo and quantum models. And the computational resources required is less than others when simulation is carried out. The majority of contemporary devices can be adequately characterized using the classical model.

As mentioned in chapter 1, Monte Carlo models are statistical models which provide a solution for the Boltzmann transport equation. The simulations based on this model are very time consuming. The less complex method is the semi-classical model based on solutions for at least first three moments of the Boltzmann transport equation, and neglecting the particulate nature of carriers by treating electrons and holes as a continuum. If we further assume that the free carriers are in thermodynamic equilibrium with the crystal lattice, in other words, three temperature parameters of device model - electron temperature, hole temperature and the lattice temperature, are same, we got the classical model - drift-diffusion model. This model assumes that the driving forces of carriers are the drift and diffusion effect only.

The classical model deals with the description of the transport of charge carriers in semiconductor structures, under the influence of the potential distribution, the properties of the semiconductor material, the geometry and doping distribution. The semiconductor is connected to the outer world via contacts or terminals, where the bias voltages are applied. The basic equations to describe the transport and hence the electrical behavior of the device consist of Poisson's equation,

$$\nabla^2\psi = -\frac{q}{\epsilon}\left(p - n + N_D^+ - N_A^-\right), \tag{2.1}$$

and the current continuity equations

$$\begin{aligned} \frac{\partial n}{\partial t} &= \frac{1}{q}\nabla j_n - R(n,p) & (2.2) \\ \frac{\partial p}{\partial t} &= -\frac{1}{q}\nabla j_p - R(n,p), & (2.3) \end{aligned}$$

for the electron and hole currents j_n and j_p, respectively. The term $R(n,p)$ in equation 2.2 and 2.3 represents the net recombination rate in semiconductor and will be discussed later in this chapter.

The Poisson's equation describes the distribution of the electrostatic potential ψ in and around the semiconductor, where $N_D^+ - N_A^-$ denotes the net ionized impurity concentration, consisting of ionized positive donors and negative acceptors, n and p are the free electron and hole concentrations, and q is the magnitude of electron charge which is equal to $1.6 \times 10^{-19} Columbs$. The properties of the semiconductor and the isolating materials affect equation 2.1 via the dielectric constant ϵ. In addition, a doped semiconductor contains ionized impurities, which influence the distribution of the electrostatic potential. The electrostatic potential can be regarded as a driving force for the charge carriers which, on the other hand, can affect the potential distribution by the Coulomb interaction.

The current continuity equations 2.2 and 2.3 guarantee particle conservation. The change in the carrier density with time is connected with the generation-recombination mechanisms $R(n,p)$ and the local carrier flux. Under steady state conditions, the time derivative will vanish.

The electrostatic potential can be regarded as one driving force for the carriers to move through the semiconductor. This part of currents can be expressed as (Page 73 in [6])

$$\begin{aligned} j_{n,drift} &= q\mu_n nE = -q\mu_n n\nabla\psi & (2.4) \\ j_{p,drift} &= q\mu_p pE = -q\mu_p p\nabla\psi & (2.5) \end{aligned}$$

The last equalities in above equations are obtained by the relation between electrical field E and electrical potential ψ: $E = -\nabla\psi$. Yet, we will see, that it is responsible only for the drift part of the driving force. The other part is the gradient in the carrier concentrations, which results in a diffusive motion of carriers and is proportional to carrier concentration and its gradient. These two contributions to the carrier transport are written in the classical drift-diffusion approximation by

$$\begin{aligned} j_n &= -q\mu_n n\nabla\psi + qD_n\nabla n, & (2.6) \\ j_p &= -q\mu_p p\nabla\psi - qD_p\nabla p. & (2.7) \end{aligned}$$

The material properties of the semiconductor enter this equation by the mobility μ and the diffusion coefficient D. The sign in the diffusion term is corresponding to the sign of charge of either the electron or the hole.

As mentioned before, it is assumed by the drift-diffusion approximation that the free carriers are in thermodynamic equilibrium with the crystal lattice. This assumption becomes as well invalid for the case of small devices as for deep biased conditions which provide the carrier with enough energy to exceed by far the global lattice temperature. The temperature can be treated as a constant over the whole device structure if the heat flow is neglected. In case the carrier temperature is different from the lattice temperature another driving force correlated with the energy transport has to be added. This one is included in the extended drift-diffusion approximation

$$\begin{aligned} j_n &= q\mu_n n\left(-\nabla\psi + \frac{\nabla(nV_t)}{n}\right), & (2.8) \\ j_p &= q\mu_p p\left(-\nabla\psi - \frac{\nabla(pV_t)}{p}\right). & (2.9) \end{aligned}$$

The forces

$$F_n = -\nabla\psi + \frac{\nabla(nV_t)}{n}, \tag{2.10}$$

$$F_p = -\nabla\psi - \frac{\nabla(pV_t)}{p} \tag{2.11}$$

are called generalized driving forces. In this derivation, we have used the Einstein relation

$$D = \mu\frac{kT}{q} = \mu V_t \tag{2.12}$$

and the thermal voltage V_t, which describes the energy of the carriers. As long as the carriers are in thermodynamic equilibrium, T is a constant and equal to the lattice temperature T_l.

2.1 Carrier Densities

For pure semiconductors, they contain no doping atoms or contamination. The electrical conductance is resulted only from the breakage of bonds. Free electrons are exited to conduction band by thermal activation of lattice, and free holes are left in the valance band. Thus the concentration of electrons and holes are equal. These concentration are the so called intrinsic carrier concentration (n_i), and is an important parameter of semiconductor materials. Because of the thermal generation mechanism, it can be expected that the intrinsic carrier concentration is dependent on lattice temperature and band gap energy of semiconductor. This value is normally referring to the room temperature $T = 300K$.

If impurity atoms with one more or less valance band, are implanted or diffused into pure semiconductor, they act as dopant by either supplying an additional electron or accepting a valence electron. The impurity which supplies an additional electron is called donor and which accepts a valence electron is called acceptor. After doped with donor or acceptor, the semiconductor changes to $n-$ or $p-$ type respectively.

The carrier densities can be determined by the relative position of the Fermi levels E_f to the conduction band bottom and valence band top (E_c and E_v respectively), and can be approximated by the following equations (Page 198 of [7])

$$
\begin{aligned}
n &= N_c \exp\left(\frac{E_f - E_c}{kT}\right) && (2.13)\\
p &= N_v \exp\left(\frac{E_v - E_f}{kT}\right). && (2.14)
\end{aligned}
$$

where quantities N_c and N_v are the effective densities of conduction- and valance-band states respectively, as long as the Fermi level is several kT above the valence or below the conduction band edge. The assumption is always satisfied except in the case of heavily doped semiconductor.

By introducing the concept of quasi-Fermi potentials of electron and hole (ϕ_n and ϕ_p) (page 269 of [7]), we can relate the carrier concentrations to intrinsic concentration as

$$
\begin{aligned}
n &= n_i \exp\left(\frac{\psi - \phi_n}{Vt}\right), && (2.15)\\
p &= n_i \exp\left(\frac{\phi_p - \psi}{Vt}\right). && (2.16)
\end{aligned}
$$

In the thermal equilibrium both Fermi potentials have the same value

$$\phi_n = \phi_p \tag{2.17}$$

With this equation we obtain relation between electrons and holes in thermal equilibrium

$$p \cdot n = n_i^2, \tag{2.18}$$

which is called the mass-action law. In the non-equilibrium case, the value of this product depends on the bias conditions, expressed by the difference between the quasi-Fermi potentials ϕ_p and ϕ_n

$$p \cdot n = n_i \exp(\frac{\phi_p - \phi_n}{Vt}). \tag{2.19}$$

Substituting equations 2.15 and 2.16 into equations 2.8 and 2.9 under the constant carrier temperature, we can rewrite the electron and hole current densities as

$$j_n = -q\mu_n n \nabla \phi_n, \tag{2.20}$$

$$j_p = -q\mu_p p \nabla \phi_p. \tag{2.21}$$

These equations are very elegant and convenient expressions, and will be used as fundamental equations for device simulation later.

2.2 Space Charge

To solve the Poisson's equation, the space charges N_D^+ and N_A^- should be known. In the present of doping atoms the semiconductor becomes extrinsic. Most impurities used are arsenic, phosphorus, antimony and boron, which have energy levels near the band edges. The donors and accepted will be ionized to positively charged and negatively charged respectively. For low to medium doping concentrations, the donor or acceptor will be almost completely ionized at room temperature. In general, the ionized impurities can be calculated from the formula

$$N_D^+ = \frac{N_D}{1 + g_D \exp[(E_f - E_D)/kT]} \tag{2.22}$$

$$N_A^- = \frac{N_A}{1 + g_A \exp[(E_A - E_f)/kT]} \tag{2.23}$$

where N_D and N_A are concentrations of dopant, E_f is the Fermi potential, E_D and E_A are the impurity energy levels which are normally close to band edges, and g_D and g_A are degenerate factor for donor and acceptor respectively. Normally, g_D is given as 2 and g_A is 4. In the examples shown in chapter 5, we will let the dopant to be completely ionized. Typical value for $E_c - E_D$ and $E_A - E_v$ for the most common dopant in silicon are: 0.045eV for arsenic, 0.045eV for phosphorus, 0.039eV for antimony and 0.045eV for boron.

2.3 Carrier Mobilities

The carrier mobility is determined by the various scattering mechanisms. We will describe these processes in following paragraphs.

Under thermal equilibrium, mobile electrons and atoms of the lattice are always in random motion. From the theory of statistical mechanics, a carrier at a temperature T is estimated to have an average thermal energy of $3kT/2$. The thermal vibrations may be treated quantum-mechanically as discrete particles called phonon. The first kind of scattering is the collision between phonons and electrons or holes, and is called lattice scattering. Lattice scattering increases with increasing temperature because of the increased lattice vibration. It dominates other processes at and above room temperature in lightly doped silicon. Most semiconductor devices are operated in this temperature range.

Besides lattice scattering, there are other scattering mechanisms: (1) The ionized impurity atoms are charged centers that may deflect the free carriers. The effect of this process depends on the temperature and impurity concentration, and the effect on the mobility is most pronounced for heavily doped samples at low temperatures, where lattice scattering can be ignored. As the temperature is increased, the fast-moving carriers become less likely to be deflected by the charged ions and the scattering is decreased. Therefore, mobilities due to this scattering increases with temperature. (2) The neutral impurity atoms may introduce scattering if the concentration of these atoms is high. Usually, the effect of this scattering is negligible. (3) The Coulomb force between carriers (electron-electron, electron-hole) can cause scattering at high concentration of carrier densities.

The impurity or lattice scattering could each become the bottleneck for the electron movement so that

$$\frac{1}{\mu_{LI}} = \frac{1}{\mu_L} + \frac{1}{\mu_I} \tag{2.24}$$

where the subscripts denote the lattice and impurity scattering, respectively. The smaller of the two is the dominating process. Figure 2.1 shows how these two components of mobility vary with temperatures.

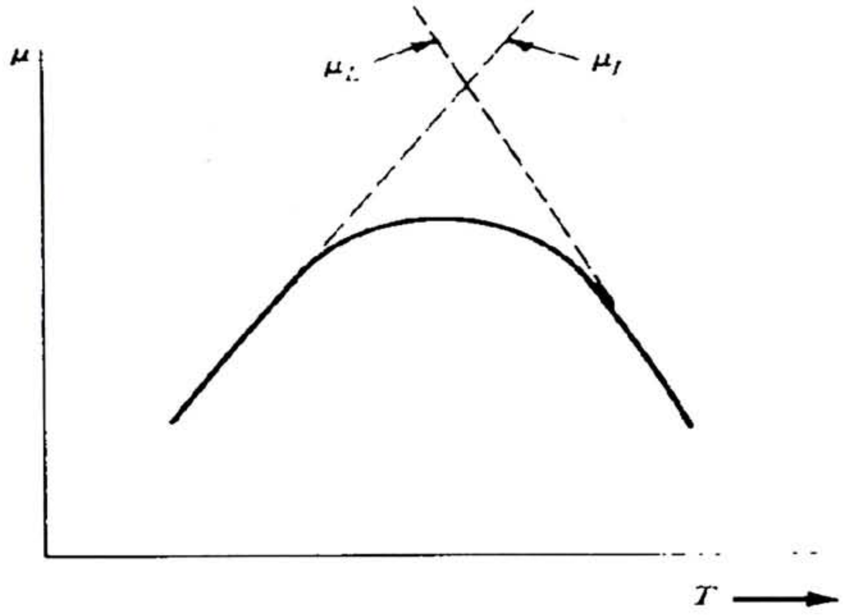


Figure 2.1: Mobility variation with temperature showing the effect of lattice and impurity scattering

In order to count the dependence on impurity concentration, the mobility can be expressed as

$$\mu_{LI} = \mu_{min} + \frac{\mu_0}{1 + (C/C^M_{ref})^\alpha}, \tag{2.25}$$

where μ_0 stands for the difference $\mu_{max} - \mu_{min}$. The quantity μ_{max} contains the lattice scattering, whereas μ_{min} and the term enclosed in brackets denotes the contribution from impurity scattering, with C as the total impurity concentration. These parameters are all temperature dependent. Values for them can be found in the literature [8].

General speaking, the drift velocity of carriers is proportional to applied electric field. This linear relationship is valid when the electric field is low. By increasing the applied field, the drift velocity eventually approaches a limit very close to the thermal velocity at room temperature. The velocity does not increase with the electric field because at high electric field, the energy gained by the electron will be lost through the emission of optical phonons. Thus, the mobility decreases as the electric field is increased, giving rise to a nonlinear mobility. This effect is called velocity saturation, and can be expressed by [9]

$$\mu_{LIE} = \mu_{LI} \left(\frac{1}{1+(E/E_c)^{\beta}} \right)^{1/\beta} . \tag{2.26}$$

where E_c denotes the critical field, and is approximately 1.5×10^4V/cm for n-type silicon. The β is 2 for electrons, and 1 for holes. The saturation velocity is related to the critical field by

$$v_{sat} = \mu_{LI} E_c . \tag{2.27}$$

Lastly, drift mobilities for Ge, Si and GaAs at 300K are shown in Table 2.1.

Table 2.1: Drift mobilities ($cm^2/V-s$) in Ge, Si, GaAs at 300K

	Ge		Si		GaAs	
	μ_n	μ_p	μ_n	μ_p	μ_n	μ_p
Value at 300K	3900	1900	1400	470	8000[a]	340
Temperature dependence	$T^{-1.66}$	$T^{-2.33}$	$T^{-2.5}$	$T^{-2.7}$	–	$T^{-2.3}$

[a] Electron mobility is dominated by polar optical-phonon scattering.

Data taken from page 220 of [7].

2.4 Generation and Recombination

In a semiconductor, electrons and holes are generated by thermal excitation of electrons from valance band to conduction band. This generation process is counterbalanced, under thermal equilibrium, by recombination processes in which electrons and holes annihilate each other. Whenever the carrier concentrations are disturbed from their equilibrium value, they will attempt to return to equilibrium state. If the excess carriers are present, the recombination process will dominate. It is the purpose of this section to describe various recombination process and give a model to one of the processes which we used in the project.

There are two basic kinds of recombination processes. In the first process, electrons in conduction band make direct transition to vacant states in valance band. In the second process, elections and holes recombine through intermediate states known as recombination centers. The recombination centers are usually impurities and lattice imperfections of some sort. We refer the first and second one as intrinsic and extrinsic recombination process respectively. It is because of that the first process is an inherent property of the semiconductor itself, while the second process depends very much on the nature of the impurity or the lattice imperfection.

The intrinsic recombination can be characterized with a lifetime, which depends greatly on the energy-band structure of a semiconductor. Figure 2.2 shows intrinsic recombination process in direct and indirect-gap semiconductors. The intrinsic recombination is important in direct-gap materials such as GaAs. On the other hand, the situation in indirect-gap semiconductor is quite different. Because the indirect radiative recombination process of Figure 2.2(b) is a two-step process, the probability of its occurrence is greatly reduced. It could be concluded that the intrinsic recombination does not play an important role for the recombination of excess carriers in indirect-gap semiconductors such as Ge and Si.

The extrinsic recombination takes place in the recombination centers or trap levels, localized in the forbidden band gap. This mechanism has been investigated by Hall, [10] Shockley and Read [11], leading to the well-know formula for the net recombination rate

$$R^{SRH} = \frac{np - n_i^2}{\tau_p(n + n_t) + \tau_n(p + p_t)}. \tag{2.28}$$

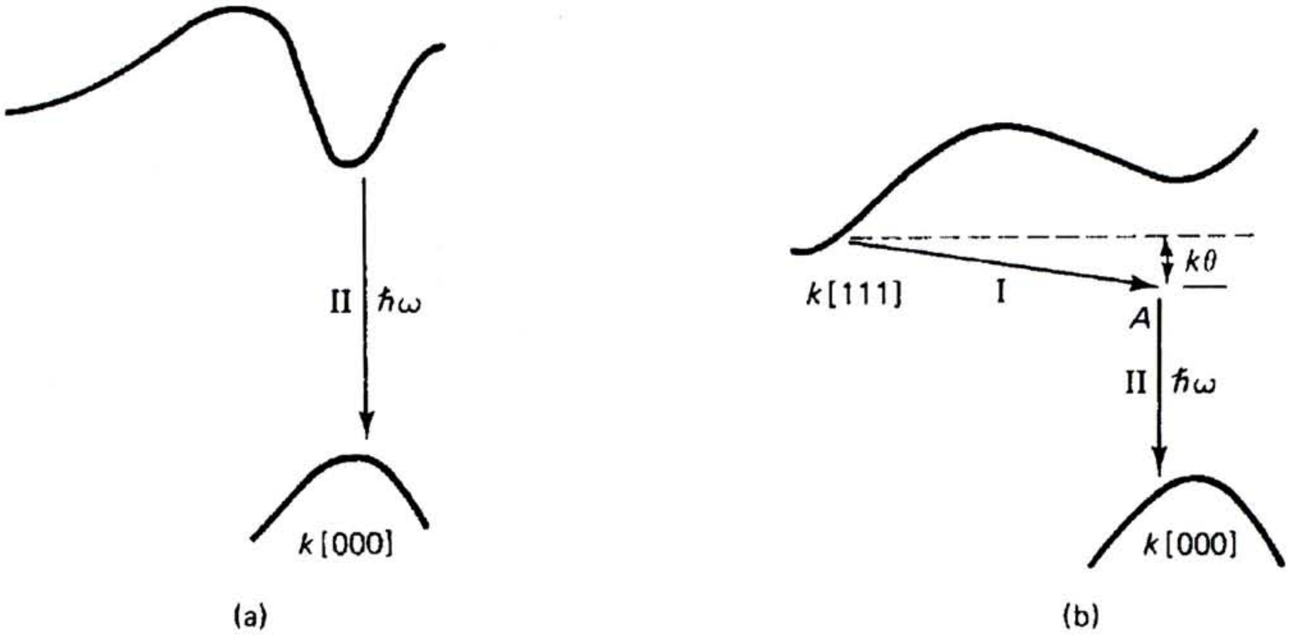


Figure 2.2: Diagram showing the intrinsic recombination process (a) in a direct-gap semiconductor and (b) in an indirect-gap semiconductor. The process shown in (b) requires phonon participation for momentum conservation.
(*Digested from figure 7.6 of [7]*)

The trap densities n_t and p_t follow from the trap energy level E_t according to Boltzmann statistics

$$\begin{aligned} n_t &= n_i \exp \frac{E_t - E_i}{kT} && (2.29) \\ p_t &= n_i \exp \frac{E_i - E_t}{kT}. && (2.30) \end{aligned}$$

The largest capture and emission probabilities are found near the middle of the band gap E_i. Use this value for E_t, the SRH recombination formula becomes

$$R^{SRH} = \frac{np - n_i^2}{\tau_p(n + n_i) + \tau_n(p + n_i)}. \tag{2.31}$$

The carrier lifetimes τ_n and τ_p are found to depends slightly on the total impurity concentration, and could be assumed to be constant for low to medium-doped semiconductor. In thermal equilibrium the Shockley-Read-Hall(SRH) term is vanishing.

Besides the SRH generation and recombination in bulk, there are also other generation recombination processes for indirect-gap semiconductor. They are: surface recombination, Auger process, impact ionization and others. However, we will not give the model for these process. Only a brief description will be given.

In the surface, due to the discontinuity in the lattice structure, there exist a large amount of energy states in the forbidden gap. These energy states, called surface state, greatly enhance the recombination rate at the surface region. In addition to the surface states, other imperfections exist resulting from absorbed ions, molecules, or mechanical damage in the layer next to the surface.

In the Auger recombination, the energy released by electron-hole recombination will be absorbed by a second electron in the conduction band. This second electron lose its extra energy to lattice by scattering events. Usually, Auger recombination is important when the carrier concentration is very high as a result of either high doping or high injection level.

At high electric fields and sufficiently high current densities, the carrier can gain enough energy for impact ionization, which is the inverse mechanism to Auger recombination. This generation mechanism is important for the modeling of MOS-transistor and power devices.

2.5 Modeling of Device Boundaries

The model described above deals with the charge distribution and carrier transport inside devices. These devices are connected to outside world via contacts or interfaces, where the biased voltages are applied or currents enter. Based on their electrical properties, device boundaries can be classified as (1) contacts, which allow a current flow into and out of the device;(2) contacts, where only voltages can be applied, such as the gate of MOSFET; (3) interfaces, where current flow disappears; (4) artificial boundaries, where neither electric field nor current flow exists, and is used to artificially isolate the device from the whole integrated circuits. Because we deal with one-dimensional devices, only ohmic contact, which belongs to the first type, will be discussed here. Models for other boundaries could be found in section 2.5 of [12].

A contact is, considered from the physical point of view, a boundary between different conducting materials. Normally, the semiconductor contact is made by metals or polysilicon, which has very high conductance. For those contacts used in emitter, base and collector terminals of bipolar transistor and drain and source terminals of MOSFET, which supply biased voltage or current flow, can be simply modeled as ohmic contact.

The ohmic contact assumes a constant value for the variable ψ, n and p, or ψ, ϕ_n and ϕ_p at the contact, and neglect the transition resistance at the interface. This is justified, as long as the work-function difference is small, or the doping concentration at the contact is so high that the electrons are able to tunnel through the barrier. The thermal equilibrium is also assumed in the contact region. With the applied voltage V_{appl} we can express the boundary condition as

$$\psi = V_{appl} + \psi_0 \tag{2.32}$$

$$\phi_n = V_{appl} + \phi_{n0} \tag{2.33}$$

$$\phi_p = V_{appl} + \phi_{p0} \tag{2.34}$$

where ψ_0, ϕ_{n0} and ϕ_{p0} are values under the absence of applied voltage (thermal equilibrium). If dopants are completely ionized, then $n = N_d$ and $p = N_A$. It could be assumed that the thermal equilibrium is hold at the boundaries, i.e., $\phi_n = \phi_p$, and if referring ϕ_{p0}/ϕ_{n0} as ground potential, we get,

$$\phi_{n0} = \phi_{p0} = 0. \tag{2.35}$$

Therefore, we can get values of ψ_0 by substituting these values of n, p, ϕ_{n0} and ϕ_{p0} into equations 2.15 and 2.16 and obtain:

$$\psi_0 = V_t \ln N_D/n_i, \quad \text{n-type} \tag{2.36}$$

$$\psi_0 = -V_t \ln N_A/n_i, \quad \text{p-type.} \tag{2.37}$$

2.6 Limits of Classical Device Modeling

In many practical applications, the models described above have proven to be useful. The parameters of these models depend partially on the technology, and have to be adjusted properly to match experimental data. In most simulators these parameters can be modified in the input-file, which may be sufficient for engineering purposes. For the investigation of new devices and non well-understood transport phenomena, this method does not provide adequate description.

However, the transport is not described completely by equations 2.1- 2.9. In certain applications additional transport mechanisms are required. There are some examples. For an accurate description of the switching behavior of power devices, where the current causes an inhomogeneous temperature distribution, the energy transport through phonons should be taken into account, leading to an additional equation for the heat-flow through the device. Besides, the hot electron effect should be taken into account for the modeling of sub-micron MOSFETs. The influence of quantum effect already arises at the horizon for the presence of two-dimensional electron gas (2DEG) devices.

In the future, we expect the semi-classical, Monte-Carlo and even the quantum models will become more and more important. However, the classical model will still serve as an efficient and computational cost-effective model for the time being.

Chapter 3

Computational Aspects

In the previous chapter, a description of classical drift-diffusion model, which is a set of coupled nonlinear partial differential equations(PDE), was given. In this chapter, we will discuss various computational aspects of solving such PDEs.

The first section is dedicated to the normalization of device equations. Because the normalization to dimensionless form is necessary to prevent overflow and underflow occuring in calculation due to the huge difference of magnitudes of different variables. It also provides elegance in formulation.

The model description is actually a set of continuous partial differential equations. However, all signals which can be manipulated by computer are in discrete form. In order to solve those PDEs by computer, we need to discretize them. There are numerous discretization methods, for example, finite difference, finite element, finite box, and collocation method. In this thesis, we propose a wavelet-based method to solve this problem, which will be described in detail in next chapter. Although our thesis is not based on these conventional methods, we will also give a brief introduction to them for comparison.

The equation set after discretization is a nonlinear system, which can be solved either by the famous Gummel's method or Newton's method. Both method will be discussed in the third section.

Table 3.1: List of normalization factors

Description	Normalized quantity	Normalization factor
Position coordinate	x	$L_D \triangleq \sqrt{\epsilon V_t/qn_i}$ (Debye length)(cm)
Electrostatic potential	ψ	V_t(Thermal voltage)(volt)
Quasi-Fermi level	ϕ_n, ϕ_p	V_t (volt)
Applied voltages	V_{appl}	V_t (volt)
Carrier concentrations	n, p	n_i (cm^{-3})
Dopant concentrations	N_D, N_A	n_i (cm^{-3})
Diffusion constants	D_n, D_p	$D_0 \triangleq 1(cm^2/sec)$
Mobilities	μ_n, μ_p	D_0/V_t $(cm^2/V-sec)$
Current densities	j_n, j_p	(qn_iD_0/L_D) (A/cm^2)
Carrier Life time	τ_n, τ_p	$1sec$

3.1 Normalization

The quantities in device equations differ from each other in huge magnitudes. Normally, carrier concentrations n and p vary from 10^{12} to $10^{19}/cm^{-3}$; electrostatic potential and varies in the range of several volts; quasi-Fermi levels ψ, ϕ_n and ϕ_p vary within 1 volt. These factors may cause overflow and underflow problems during the iteration process of solving the nonlinear system, which will be discussed later. Therefore, it is convenient to normalize these quantities and express device equations in dimensionless form. Most people use the normalization factors as listed in Table 3.1, which is suggested by A. De Mari [13].

After normalization, for one-dimensional case, the Poisson's equation 2.1, current continuity equations 2.2 and transport equations 2.20 become respectively,

$$\frac{\partial^2\psi}{\partial x^2} = -(p-n+N), \tag{3.1}$$

$$\frac{L_D^2}{D_0}\frac{\partial n}{\partial t} = \frac{\partial j_n}{\partial x} - R(n,p), \tag{3.2}$$

$$\frac{L_D^2}{D_0}\frac{\partial p}{\partial t} = -\frac{\partial j_p}{\partial x} - R(n,p), \tag{3.3}$$

$$j_n = -\mu_n n\frac{\partial \phi_n}{\partial x}, \tag{3.4}$$

$$j_p = -\mu_p p\frac{\partial \phi_p}{\partial x}, \tag{3.5}$$

where N is net doping concentration, n and p are

$$n = \exp(\psi - \phi_n), \tag{3.6}$$

$$p = \exp(\phi_p - \psi), \tag{3.7}$$

and the normalized Read-Shockley-Hall recombination rate is

$$R^{SRH} = \frac{L_D^2}{D_0} \frac{np - 1}{\tau_p(n+1) + \tau_n(p+1)}. \tag{3.8}$$

In addition, the boundary conditions (equations 2.36) becomes

$$\psi = V_{appl} + \ln N_D, \quad \text{n-type}, \tag{3.9}$$

$$\psi = V_{appl} - \ln N_A, \quad \text{p-type}, \tag{3.10}$$

$$\phi_n = \phi_p = V_{appl}. \tag{3.11}$$

Note that all the quantities in the above equations are normalized quantities.

By substituting equations 3.4 and 3.5 to equations 3.2 and 3.3, and using the relation $n = \exp(\psi - \phi_n)$, $p = \exp(\phi_p - \psi)$, we get

$$\frac{\partial}{\partial x}\left(\mu_n e^{(\psi - \phi_n)} \frac{\partial \phi_n}{\partial x}\right) = -\left(R + \frac{L_D^2}{D_0}\frac{\partial n}{\partial t}\right) \tag{3.12}$$

$$\frac{\partial}{\partial x}\left(\mu_p e^{(\phi_p - \psi)} \frac{\partial \phi_p}{\partial x}\right) = R + \frac{L_D^2}{D_0}\frac{\partial n}{\partial t}. \tag{3.13}$$

Above two equations together with equation 3.1, are the basic equation set, with three variables ψ, ϕ_n, ϕ_p, and boundary conditions 3.9, 3.10, 3.11.

If we further assume that carrier mobilities are constants independent of x, above two equations becomes,

$$\frac{\partial^2 \phi_n}{\partial x^2} + \frac{\partial \phi_n}{\partial x}\frac{\partial \psi}{\partial x} - (\frac{\partial \phi_n}{\partial x})^2 = -\frac{1}{\mu_n n}\left(R + \frac{L_D^2}{D_0}\frac{\partial n}{\partial t}\right) \tag{3.14}$$

$$\frac{\partial^2 \phi_p}{\partial x^2} + (\frac{\partial \phi_p}{\partial x})^2 - \frac{\partial \phi_p}{\partial x}\frac{\partial \psi}{\partial x} = \frac{1}{\mu_p p}\left(R + \frac{L_D^2}{D_0}\frac{\partial n}{\partial t}\right). \tag{3.15}$$

For steady state, the time varying terms vanish, and equations 3.14 and 3.17 become:

$$\frac{\partial^2 \phi_n}{\partial x^2} + \frac{\partial \phi_n}{\partial x}\frac{\partial \psi}{\partial x} - (\frac{\partial \phi_n}{\partial x})^2 = -\frac{1}{\mu_n n}R \tag{3.16}$$

$$\frac{\partial^2 \phi_p}{\partial x^2} + (\frac{\partial \phi_p}{\partial x})^2 - \frac{\partial \phi_p}{\partial x}\frac{\partial \psi}{\partial x} = \frac{1}{\mu_p p}R. \tag{3.17}$$

3.2 Discretization

To solve the devices equation by computer, we need to discretize them. There exist various discretization schemes, including finite difference and finite element which are most popular. In the following these important methods will be briefly described.

3.2.1 Finite Difference Method

According to Taylor theorem, any function $f(x)$, having 3 continuous derivatives on interval $[a, b]$, and let $x, x_0 \in [a, b]$, then $f(x)$ can be expressed as

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + f''(x_0)\frac{(x - x_0)^2}{2!} + R(x), \qquad (3.18)$$

$$\text{where } R(x) = \frac{(x - x_0)^3}{3!} f^{(3)}(\xi) \qquad (3.19)$$

for some ξ between x_0 and x.

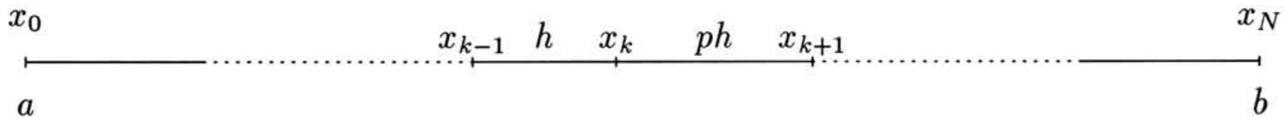


Figure 3.1: Grid points in the interval $[a, b]$

Now, let the grid points be $x_0, x_1, \cdots, x_{k-1}, x_k, x_{k+1}, \cdots, x_N$ where $x_0 = a$ and $x_N = b$, as shown in Figure 3.1. Let

$$h = x_k - x_{k-1} \qquad (3.20)$$

$$ph = x_{k+1} - x_k \qquad (3.21)$$

where p is a real number, which represents the ratio between two adjacent step size. If we approximate $f(x)$ as Taylor series expanded near x_k and evaluate its value at x_{k-1} and x_{k+1}, then

$$f(x_{k-1}) = f(x_k) - hf'(x_k) + \frac{h^2}{2} f''(x_k) + R_1, \qquad (3.22)$$

$$f(x_{x+1}) = f(x_k) + phf'(x_k) + \frac{(ph)^2}{2} f''(x_k) + R_2, \qquad (3.23)$$

for small grid size, remainder R_1, R_2 can be ignored. We will not discussed the error analysis of this method, which can be found in many books [14]. Eliminating $f''(x_k)$ in the above two equations, we can get the second order approximation to 1^{st} derivative at x_k,

$$f'(x_k) = \frac{f(x_{k+1}) - (1-p^2)f(x_k) - p^2 f(x_{k-1})}{p(1+p)h}. \tag{3.24}$$

In the same way, by eliminating $f'(x_k)$, we can get the second order approximation of $f''(x_x)$

$$f''(x_k) = \frac{f(x_{k+1}) - (1+p)f(x_k) + pf(x_{k-1})}{p(1+p)h^2/2}. \tag{3.25}$$

Here we use a ratio p, and this expression is more convenient than those using h_{k-1}, h_k. For uniform grid points, p is equal to 1.

3.2.2 Finite Element Method

Finite element method(FEM) is achieved by solving variational integral equations, which are equivalent to the differential equations 3.1 3.16 3.17. The advantages of FEM are [15]: (1)The application to problems with irregular geometry and different sizes of elements in different regions of the domain is no more difficult than for regular geometry and meshes. Such problems are more complex when tackled using FDM. (2) Once a computer code has been written for a particular order of trial function, only very minor changes to the code are needed in order to change the order of trial function. Conversely, however, in order to change the accuracy of a finite difference code, major modifications are usually required. (3) Procedures involving successive mesh refinements are more easily incorporated into finite element schemes.

Other considerations, however, may favor FDM. Here are some cases: (1) It is generally much easier to write a finite difference rather than a finite element computer code. (2) Error analysis is much easier to perform for FDM. (3) FDM generally requires smaller computer memory than finite element method.

3.3 Nonlinear Systems

3.3.1 Newton's Method

After discretization, equations (3.1), (3.16), (3.17) with boundary conditions (3.9,3.10 and 3.11) become a nonlinear system. Then, such nonlinear system could be solved by either Newton's method or Gummel's method. Using Newton's method for device simulation is first suggested by B.V. Gokhale [16]. Rewrite equation 3.1,3.16 and 3.17 as

$$\begin{aligned} f_\psi &= \frac{\partial^2 \psi}{\partial x^2} + (p - n + N) = 0, & (3.26)\\ f_n &= \frac{\partial^2 \phi_n}{\partial x^2} + \frac{\partial \phi_n}{\partial x}\frac{\partial \psi}{\partial x} - (\frac{\partial \phi_n}{\partial x})^2 + \frac{1}{n}R(n,p) = 0, & (3.27)\\ f_p &= \frac{\partial^2 \phi_p}{\partial x^2} + (\frac{\partial \phi_p}{\partial x})^2 - \frac{\partial \phi_n}{\partial x}\frac{\partial \psi}{\partial x} - \frac{1}{p}R(n,p) = 0. & (3.28)\end{aligned}$$

The Newton's method involves the linear system:

$$y^{k+1} = y^k - J(y^k)f(y^k), \tag{3.29}$$

$$\text{where} \quad y = [\psi, \phi_n, \phi_p]^T,$$

$$f = [f_\psi, f_n, f_p]^T,$$

and $J(y^k)$ is Jacobian evaluated at iteration k, where iteration number is expressed as superscript,

$$J = \begin{pmatrix} \frac{\partial f_\psi}{\partial \psi} & \frac{\partial f_\psi}{\partial \phi_n} & \frac{\partial f_\psi}{\partial \phi_p} \\ \frac{\partial f_n}{\partial \psi} & \frac{\partial f_n}{\partial \phi_n} & \frac{\partial f_n}{\partial \phi_p} \\ \frac{\partial f_p}{\partial \psi} & \frac{\partial f_p}{\partial \phi_n} & \frac{\partial f_p}{\partial \phi_p} \end{pmatrix}. \tag{3.30}$$

Note that all variables ψ, ϕ_n and ϕ_p are updated simultaneously in each iteration. This method starts from a trial solution y_0 and stops when the difference between the results of two consecutive iterations is less than a specified small number.

3.3.2 Gummel's Method and its modification

Different from Newton's method, in Gummel's method [17] three variable updating processes in each iteration are used instead of simultaneous updating process. Here, we suggest a modification to Gummel's method. In our method the electrical potential ψ is updated by the approach used in Gummel's original method; the quasi-Fermi potentials of electrons ϕ_n and holes ϕ_p are updated by one-step Newton-Raphson's method. The following paragraphs describe the detail of the improved Gummel's method.

Firstly, we choose trial waveforms ψ^0, ϕ_n^0 and ϕ_p^0 of the electrical potential , electron and hole quasi-Fermi potential respectively.

Then, Poisson's equation (3.1) is used to update electrical potential ψ^k, where k is iteration index. Let $\psi^{k+1} = \psi^k + \delta$. For small perturbation δ, neglecting the second order and higher order terms, $e^\delta \approx 1 + \delta$. Then,

$$\begin{aligned} \delta'' &- \delta(e^{(\psi^k-\phi_n^k)} + e^{(\phi_p^k-\psi^k)}) \\ &= -(\psi^k)'' + e^{(\psi^k-\phi_n^k)} - e^{(\phi_p^k-\psi^k)} - N. \end{aligned} \tag{3.31}$$

Equation (3.31) is a linear differential equation for δ, and is solved by finite difference method. The updated electrical potential ψ^{k+1} is

$$\psi^{k+1} = \psi^k + \delta = \psi^k + T^{-1}B, \tag{3.32}$$

where T is a "tridiagonal matrix", and $B = -(\psi^k)'' + e^{(\psi^k-\phi_n^k)} - e^{(\phi_p^k-\psi^k)} - N$ is a column vector. The next step is to update electron quasi-Fermi potential ϕ_n according to electron transport equation 3.28. To solve it numerically, this equation is discretized by finite difference scheme as mentioned in previous section. Let $\psi = \psi^{k+1}, \phi_p = \phi_p^k$ in equation (3.28). It becomes a nonlinear equation for ϕ_n now. To update ϕ_n, one-step Newton-Raphson's method is used:

$$\phi_n^{k+1} = \phi_n^k - J_n(\phi_n^k)f_n(\phi_n^k), \tag{3.33}$$

where $J_n(\phi_n)$ is Jacobian matrix of $f_n(\phi_n)$, and it is only a portion of Jacobian matrix in Newton's method. It is not necessary to complete the Newton-Raphson's scheme since this ϕ_n is only the intermediate result.

The last variable needed to update is ϕ_p according to hole's transport equation 3.28. Using ψ^{k+1} and ϕ_n^{k+1} for ψ and ϕ_n respectively in equation (3.28). Similar to ϕ_n, we update ϕ_p by one-step Newton-Raphson's scheme:

$$\phi_p^{k+1} = \phi_p^k - J_p(\phi_p^k) f_p(\phi_p^k), \tag{3.34}$$

where $J_p(\phi_p)$ is Jacobian matrix of $f_p(\phi_p)$.

After updating ψ, ϕ_n and ϕ_p, we check if the scheme converges by evaluating the error between two consecutive solutions. If the error is less than a specified small number, the solution is obtained.

We will now summarize the overall scheme.

Step 1: Choose the trial waveforms of ψ^0, ϕ_n^0 and ϕ_p^0.
Step 2: Update ψ by equation (3.32).
Step 3: Update ϕ_n by equation (3.33).
Step 4: Update ϕ_p by equation (3.34).
Step 5: Check convergence, if not converges, return to Step 2.

3.3.3 Comparison and discussion

It should be pointed out that Gummel's method is a decoupled system of Newton's method. From Jacobian matrix (equation 3.30) in Newton's method, we see that if the non-diagonal elements are zeros or close to zeros, the resultant linear system is actually that in Gummel's method.

The Jacobian matrix of Newton's method is a multi-band matrix, and that of Gummel's method is a tri-diagonal matrix. The dimension of later one is only 1/3 of former one.

In general, Newton's method is more robust in the means of convergence and slower than Gummel's method. In slightly coupled system, we prefer the Gummel's method because of higher computational speed. However, for heavily coupled system, Newton's method should be adopted due to its robustness.

Gummel's method is favourate in the following cases: (1) Low current densities where the diffusion term in the current continuity equation is dominating. (2) Generation-recombination term is independent of electric field $\nabla\psi$, i.e., the electric field strength is lower than the avalanche threshold; (3) The mobility is independent of electric field $\nabla\psi$. In these cases, the equations can be treated as decoupled. We like to study and compare Newton's method and Gummel's method under these circumstances.

As an example an abrupt p-n junction was simulated by two methods. The choice of an abrupt impurity distribution is motivated by accuracy considerations, since such a doping profile maximizes truncation errors and represents numerically the worst case.

The diode used here is taken from [13]. Its physical parameters are listed in Table 3.2.

Table 3.2: Physical parameters of a diode.

Material	germanium (relative permittivity = 16)
Temperature	$300^0 K$
Doping	Donar: $10^4(or 2.5 \times 10^{17} cm^{-3})$ $10^2(or 2.5 \times 10^{15} cm^{-3}$
Length	N-side, M-O=2.2 (or $2.1043 \times 10^{-4} cm$) P-side, L-M=4.0 (or $3.8267 \times 10^{-4} cm$)
Carrier mobilities	electron, $3600 cm^2/V - sec$ hole, $1700 cm^2/V - sec$

Simulation is carried out both by Newton's method and our method. Obviously, we should use the same trial waveform for the beginning of two methods in order to have a fair comparison. The grid points of the diode are not necessary of equal distance.

The simulation program has been written in Matlab, and run on SUN Ultra Sparc CPU. The iteration and simulation time for the diode under forward bias $V_b = 0.05V$ is shown in the Table 3.3. And the numerical solutions of ψ, ϕ_n and ϕ_p are shown in Figure 3.2.

Table 3.3: The iteration number and simulation time of Newton's and our methods

Grid point Number	65	89	121	185	241	313	369
Newton's method	6[a]	7	7	7	7	7	7
	3.88[b]	8.8	18.19	55.62	114.44	237.51	378.68
Our method	8	7	7	7	7	7	7
	3.46	4.96	8.68	20.02	36.73	63.49	95.08

[a] the number of iteration needed for program to converge.

[b] simulation time (in seconds) running on SUN Ultra Sparc.

From the data listed in Table 3.3, we observe that for small grid point number (less than 65), the simulation times of two methods are competitive. However, for large grid point number (large than 185), the simulation time of our method is only 1/4 of that of Newton's method. The fast computation of the modified Gummel's method is easy to understand. Since we just need to invert $N \times N$ matrix instead of inverting $3N \times 3N$ one, where N is grid point number. In real device simulation, the grid point number is always over one hundred. If this technique is applied, significant computational cost can be saved. For large bias voltage, these methods will not converge. To overcome this problem, it need to increase bias voltage from small value by several small step.

3.4 Linear System and Sparse Matrix

After device equations were discretized by either FDMs or FEMs and linearized either by Newton's or Gummel's method, the last step is to solve a linear system in the form of

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{3.35}$$

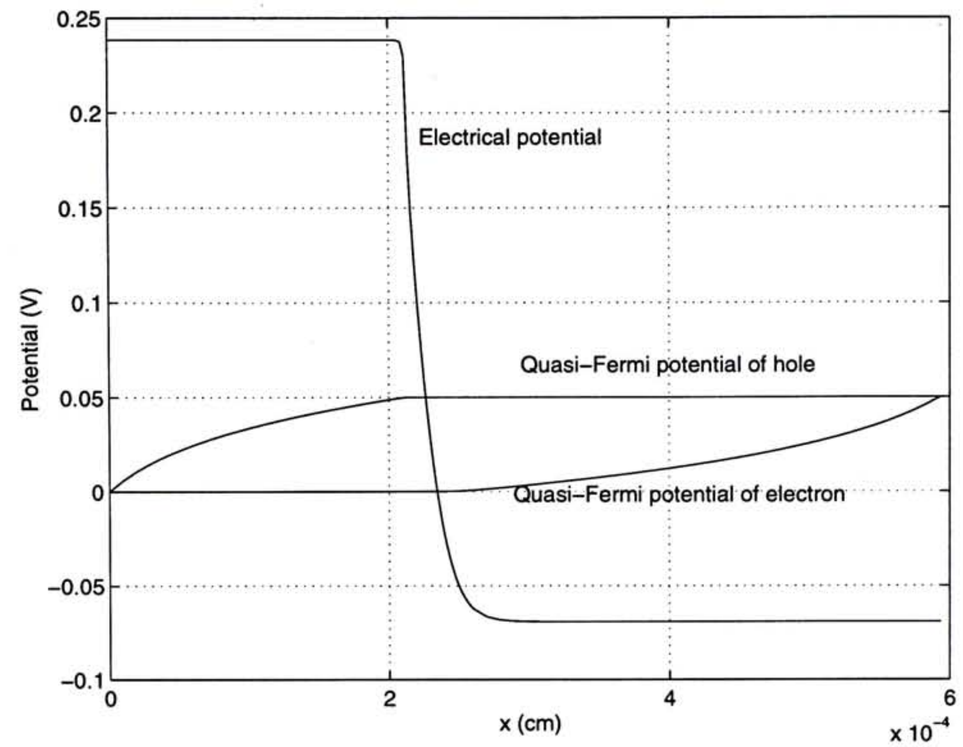


Figure 3.2: The numerical solution of ψ, ϕ_n and ϕ_p of an N-P diode with bias voltage $V_b = 0.05V$.

There are two kinds of method to solve it: direct methods and indirect methods. Direct methods mean to invert the matrix **A** directly. Indirect methods has many variations. In general, they use iterative scheme to approximate the solution of equation 3.35. The advantage of direct methods is robustness. However, for very large scale system, indirect methods are prefered because they are faster in such case. In our research, we use direct method for its robustness. These methods are well understood and can be easily found in many textbooks related to numerical analysis [14].

Usually, the matrix **A** is very sparse, i.e, it contains many zero elements or elements very close to zero. To save memory space, and more importantly, to reduce the computational time, spare matrix technique should be used. There are many solfware packages with linear equation solver with sparse matrix technique, for example, the Matlab$^{(R)}$ of the Math Works Inc. In this package, the sparse matrix command is "*sparse*()". A very simple command can be use to solve equation 3.35: "$x = sparse(A)\backslash b$".

Chapter 4

Cubic Spline Wavelet Collocation Method for PDEs

There are various wavelet methods for discretizing differential equations published in this decade, including wavelet-Galerkin's method[3, 18](finite element type), and wavelet collocation method[18]. In finite element type methods, piecewise polynomial trial functions may be replaced by wavelets, such as Doubechies wavelets. This method was originally proposed by Glowinski, Lawton, Ravachol and Tenenbaum[5]. Many interesting examples in their paper suggest that wavelets have great potential in the application to numerical solution of differential equations. However, computing integrals of products of derivatives of wavelets needed in wavelet Galerkin's method is very difficult. Chen [19] suggests a Haar wavelet method to PDEs, and show great potential in circuit analysis and in lumped and distributed-parameter systems. The idea of Chen's method is using orthogonal Haar wavelet functions to construct operational matrices, to convert a differential equation into an algebraic equation, and hence the solution procedures are much simplified.

Recently, Wei Cai develops an adaptive multiresolution cubic spline wavelet collocation methods for initial boundary value problems of nonlinear PDEs[1]. Through many examples in his paper he shows that this method is suitable for nonlinear PDEs.

Zhou introduces this method in linear and nonlinear circuit simulation [20]. Zhou exploits the multiresolution property of this method in many examples in his papers.

In our research, we use Cai's wavelet for discretizing device equations in spatial domain, and use finite difference method in time domain discretization. This idea is known as "the method of lines". Besides, we use adaptive method to identify collocation points and non-truncated wavelet coefficients. This procedure is equivalent to the grid point generation procedure in some device simulator using finite difference scheme as described in chapter 1. However, due to the localization property of wavelets, a major portion of wavelets will be located on the sharp-changing region which contains major device information. As a result, lots of computational efforts will be saved.

In this chapter, we will describe the adaptive cubic spline wavelet collocation method in parallel with the way of implementing it in Matlab^{TM} programming environment. The rest of this chapter is divided into the following sections. In section 1, we introduce the cubic scaling functions $\phi(x), \phi_b(x)$ and their wavelet functions $\psi(x), \psi_b(x)$. An MRA and its corresponding wavelet decomposition of the Sobolev space $H_0^2(I)$ are constructed using $\phi(x), \phi_b(x), \psi(x)$ and $\psi_b(x)$. Then, we show how to construct a wavelet approximation for functions in the Sobolev space $H^2(I)$. In addition, we will describe how to generate these functions in Matlab^{TM}. In section 3, we discuss the derivative matrix for approximating differential operators. In section 4, we present the wavelet collocation method for static and transient device simulation. In section 5, we present an adaptive scheme to reduce collocation points in order to achieve high computational efficiency.

4.1 Cubic spline scaling functions and wavelets

The wavelet and MRA proposed by Cai is constructed in a Sobolev space $H_0^2(I)$. Because the inner product considered here is in space $H_0^2(I)$, not in space $L^2(I)$, these two mother wavelet functions will no longer have vanishing moments of the first two orders as usual wavelets in space L^2. For simplicity, Cai still used the term "wavelet" in his paper with the understanding that it is different from the usual wavelet with its non-vanishing moment. Cai pointed out that despite the fact that these wavelet-like functions have non-vanishing moments, the projection f_i of any function $f_i \in H_0^2(I)$ on V_j still provides a "blurred" version of function f while the one on space W_j keeps its local details. Hence the magnitude of coefficients in the wavelet expansion of functions in $H_0^2(I)$ does reflect the local scales and change of the function to be approximated. It is these features that are needed for achieving adaptivities and MRAs in practical computations.

Now, we begin to describe the scaling function and its wavelet function which will be used in our research project.

Let I denote a finite interval, say $I = [0, L]$, L being a positive integer (for the sake of simplicity, it is assumed that $L > 4$), and $H^2(I)$ and $H_0^2(I)$ denote the following two Sobolev spaces:

$$\begin{aligned} H^2(I) &= \{f(x), x \in I |\ \|f^{(i)}\|_2 < \infty,\ i = 0, 1, 2\}. && (4.1) \\ H_0^2(I) &= \{f(x) \in H^2(I) |\ f(0) = f'(0) = f(L) = f'(L) = 0\}. && (4.2) \end{aligned}$$

$H_0^2(I)$ is a Hilbert space equipped with inner product

$$< f, g >= \int_I f''(x) g''(x) dx. \tag{4.3}$$

thus,

$$\|f\| = \sqrt{< f, f >} \tag{4.4}$$

provides a norm for $H_0^2(I)$.

Cai suggested a scaling function $\phi(x)$ and a boundary scaling function $\phi_b(x)$ to generate an MRA for the Sobolev space $H_0^2(I)$:

$$\begin{aligned} \phi(x) &= N_4(x) = \frac{1}{6}\sum_{j=0}^{4} C_j^4(-1)^j(x-j)_+^3. && (4.5) \\ \phi_b(x) &= \frac{3}{2}x_+^2 - \frac{11}{12}x_+^3 + \frac{3}{2}(x-1)_+^3 - \frac{3}{4}(x-2)_+^3, && (4.6) \end{aligned}$$

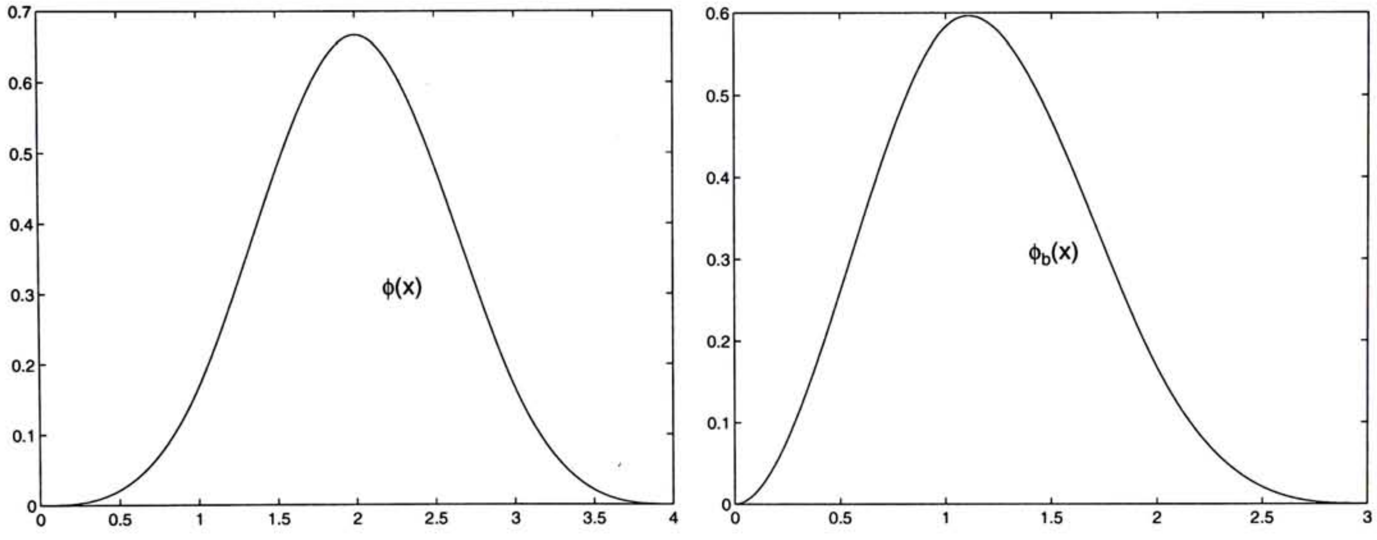
where $N_4(x)$ is the fourth order B-spline and for any real number n,

$$x_+^n = \begin{cases} x^n & \text{if } x \geq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{4.7}$$

The graphs of $\phi(x)$ and $\phi_b(x)$ are shown in Figure 4.1.Note that

$$\begin{aligned} &supp(\phi(x)) = [0,4]; \\ &supp(\phi_b(x)) = [0,3]; \end{aligned} \tag{4.8}$$

where *supp* means the nonzero interval of functions. The interior scaling function can



(a) (b)

Figure 4.1: (a) Interior scaling functions $\phi(x)$ and (b) boundary scaling function $\phi_b(x)$.

be expressed in another way as follows.

$$\phi(x) = \begin{cases} \frac{1}{6}x^3 & 0 \le x \le 1 \\ -\frac{1}{2}x^3 + \frac{1}{2}x^2 + \frac{1}{2}x + \frac{1}{6} & 1 \le x \le 2 \\ \frac{1}{2}x^3 - x^2 + \frac{2}{3} & 2 \le x \le 3 \\ -\frac{1}{6}x^3 + \frac{1}{2}x^2 - \frac{1}{2}x + \frac{1}{6} & 3 \le x \le 4 \\ 0 & \text{otherwise} \end{cases} \tag{4.9}$$

It is actually a piecewise-polynomial with knot $[0, 1, 2, 3, 4]$. In Matlab, we can generate these piecewise-polynomial easily. A few commands shown below can generate piecewise polynomial representation of $\phi(x)$.

```
knot1=[0:4];
coefficients1=[
1/6  0  0  0;
-1/2  1/2  1/2  1/6;
1/2  -1  0  2/3;
-1/6  1/2  -1/2  1/6; ];

pp=mkpp(knot1,coefficients1);
```

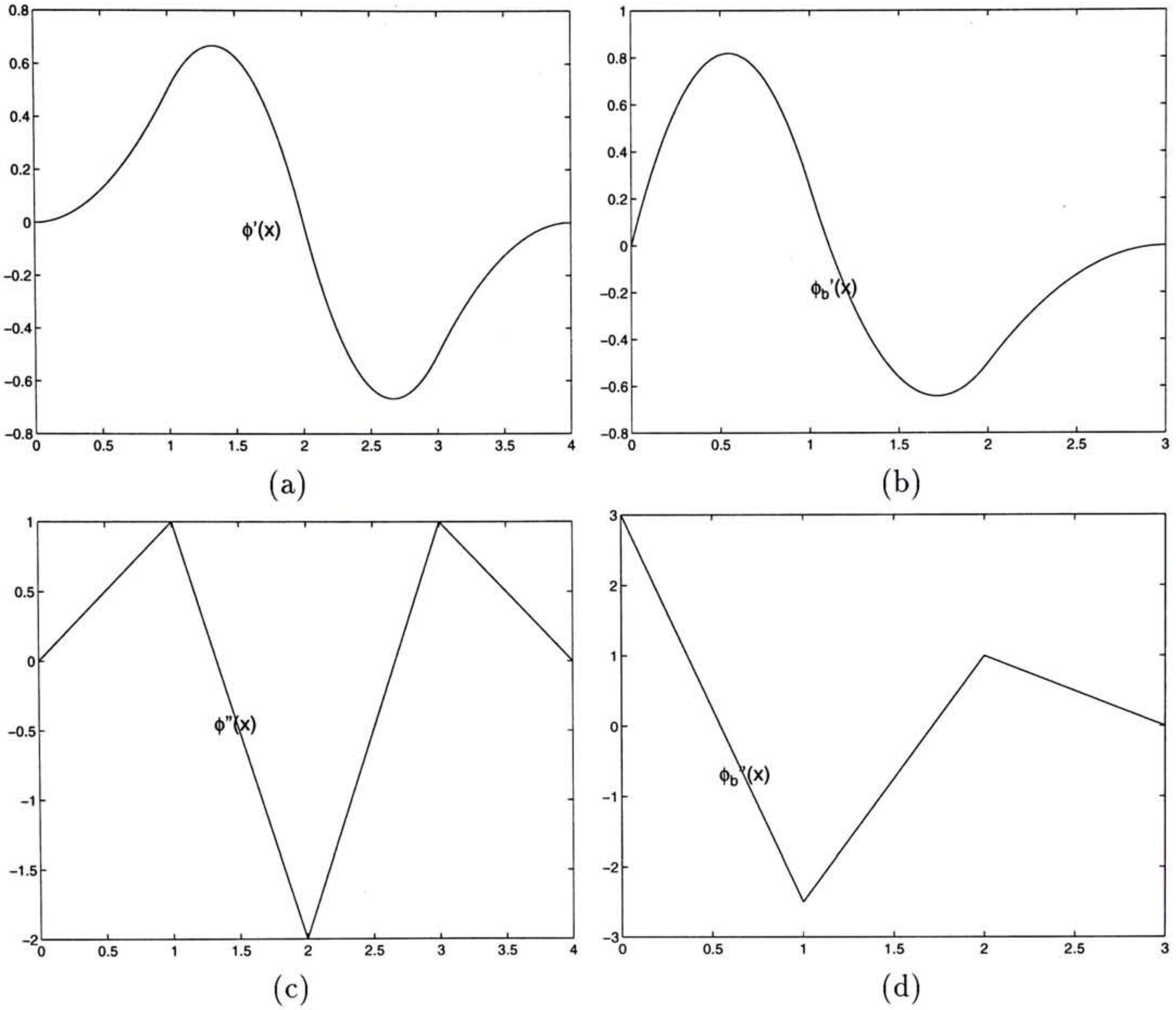
To evaluate its value in point x, use the command "ppval(x,pp)". Similarly, boundary scaling function can be expressed as:

$$\phi_b(x) = \begin{cases} -\frac{11}{12}x^3 + \frac{3}{2}x^2 & 0 \le x \le 1 \\ \frac{7}{12}x^3 - \frac{5}{4}x^2 + \frac{1}{4}x + \frac{7}{12} & 1 \le x \le 2 \\ -\frac{1}{6}x^3 + \frac{1}{2}x^2 - \frac{1}{2}x + \frac{1}{6} & 2 \le x \le 3 \\ 0 & \text{otherwise} \end{cases} \tag{4.10}$$

It can be generated by following commands.

```
knot2=[0:3];
coefficients2=[
-11/12  3/2  0  0;
7/12  -5/4  1/4  7/12;
-1/6  1/2  -1/2  1/6 ];

ppb=mkpp(knot2,coefficients2);
```



(a) (b) (c) (d)

Figure 4.2: (a) The first derivative of $\phi(x)$; (b) The second derivative of $\phi(x)$; (c) The first derivative of $\phi_b(x)$; (d) The second derivative of $\phi_b(x)$.

To evaluate its value in point x, use command "ppval(x,ppb)". Later in using collocation method, we will need the first and second derivatives of these scaling functions. To get their derivatives, use the command "fnder()". For example, commands

```
pp_d=fnder(pp);        pp_dd=fnder(pp_d);
```

generate piecewise polynomial representations of the first and second derivatives of $\phi(x)$('pp'). The first and second derivatives of $\phi(x)$ and $\phi_b(x)$ are shown in Figure 4.2

The scaling functions $\phi(x)$ and $\phi_b(x)$ satisfy the two-scale relationships given in (4.11).

$$\begin{aligned}\phi(x) &= \sum_{k=0}^{4} 2^{-3}C_k^4\phi(2x-k),\\ \phi_b(x) &= \beta_{-1}\phi_b(2x)+\sum_{k=0}^{2}\beta_k\phi(2x-k),\\ &\quad \text{where } \beta_{-1}=\frac{1}{4}, \beta_0=\frac{11}{16}, \beta_1=\frac{1}{2}, \beta_2=\frac{1}{8}.\end{aligned} \tag{4.11}$$

For any $j,k\in\mathcal{Z}$, define

$$\phi_{j,k}(x)=\phi(2^jx-k);\qquad \phi_{b,j}(x)=\phi_b(2^jx), \tag{4.12}$$

and let V_j be the linear span of $\{\phi_{j,k}(x),\ 0\le k\le 2^jL-4,\ \phi_{b,j}(x),\phi_{b,j}(L-x)\}$, namely,

$$V_j = span\{\phi_{j,k}(x),\ 0\le k\le 2^jL-4,\ \phi_{b,j}(x),\phi_{b,j}(L-x)\}. \tag{4.13}$$

Theorem 4.1 *Let $V_j, j\in\mathcal{Z}^+$ be the linear span of 4.13. Then V_j forms an MRA for $H_0^2(I)$ equipped with norm(4.4) in the following sense:*

(i) $V_0\subset V_1\subset V_2\subset\cdots$;

(ii) $clos_{H_0^2 I}(\cup_{j\in Z^+}) = H_0^2(I)$;

(iii) $\cap_{j\in Z^+}V_j = V_0$; and

(iv) for each j, $\{\phi_{j,k}(k),\phi_{b,j}(x),\phi_{b,j}(L-x)\}$ is a basis of V_j.

The proof of 4.1 was given by Cai [1], and is omitted here. For those interested in detail please refer to Cai's paper.

To construct a wavelet decomposition of Sobolev space $H_0^2(I)$ under the inner product 4.3, we consider the following two wavelet functions $\psi(x),\psi_b(x)$(shown in Figure 4.3):

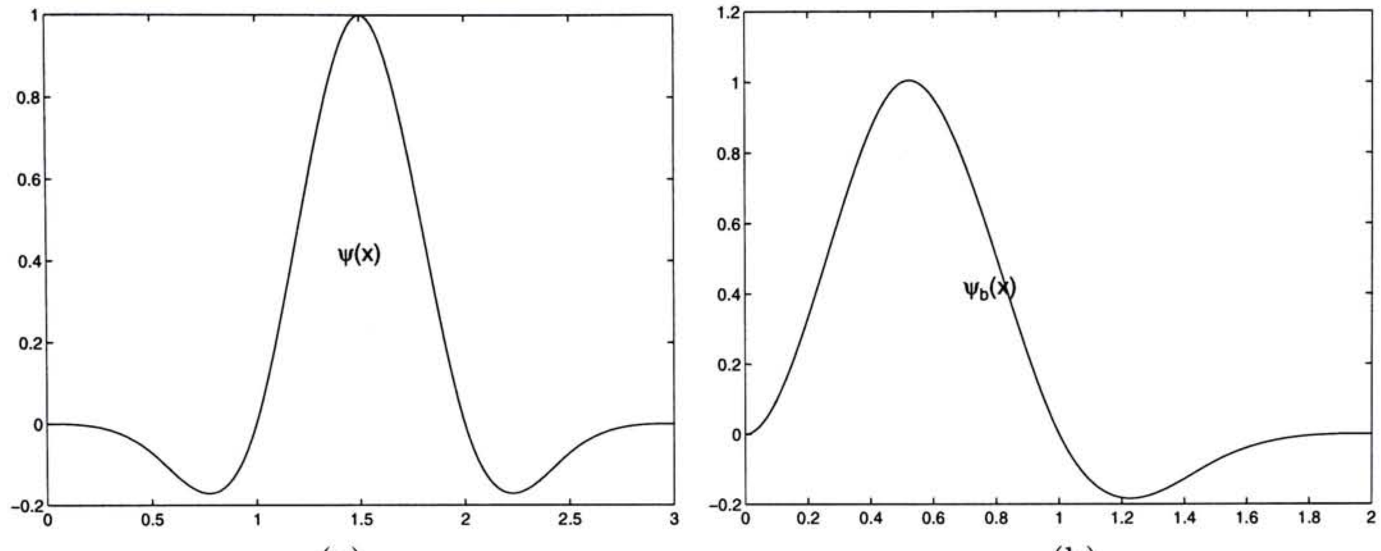
$$\psi(x) = -\frac{3}{7}\phi(2x)+\frac{12}{7}\phi(2x-1)-\frac{3}{7}\phi(2x-2)\ \in V_1. \tag{4.14}$$

$$\psi_b(x) = \frac{24}{13}\phi_b(2x)-\frac{6}{13}\phi(2x)\ \in V_1. \tag{4.15}$$

It can be verified that:

$$\psi(n) = \psi_b(n) = 0 \quad \text{for all } n \in Z. \tag{4.16}$$

This is a very important feature for constructing fast discrete wavelet transform. Following commands can generate the piecewise polynomial representation of wavelet function $\psi(x)$ in Matlab:



(a)

(b)

Figure 4.3: (a) Interior wavelet functions $\psi(x)$; (b)Boundary wavelet function $\psi_(x)$.

```
%generate polynomial coefficients of compressed scaling function
lk2=[
8*1/6  0  0  0;
-8*1/2  4*1/2  2*1/2  1/6;
8*1/2  -4*1  0  2/3;
-8*1/6  4*1/2  -2*1/2  1/6; ];

knot3=[0:0.5:3];
%generate coefficients of wavelet function from lk2
coefficient3=[
-3/7*lk2(1,:);
-3/7*lk2(2,:) +12/7*lk2(1,:);
-3/7*lk2(3,:) +12/7*lk2(2,:) -3/7*lk2(1,:);
-3/7*lk2(4,:) +12/7*lk2(3,:) -3/7*lk2(2,:);
+12/7*lk2(4,:) -3/7*lk2(3,:);
     -3/7*lk2(4,:);  ];
```

```
ww=mkpp(knot3,coefficient3);
```

To generate boundary wavelet function $\psi_b(x)$, use these commands:

```
%generate coefficients of compressed boundary scaling function
lk4=[
-8*11/12 4*3/2  0  0;
8*7/12  -4*5/4  2*1/4  7/12;
-8*1/6  4*1/2  -2*1/2  1/6; ];

knot4=[0:0.5:2];
%generate coefficients of wavelet function from lk4
coefficients4=[
24/13*lk4(1,:) -6/13*lk2(1,:);
24/13*lk4(2,:) -6/13*lk2(2,:);
24/13*lk4(3,:) -6/13*lk2(3,:);
       -6/13*lk2(4,:); ];

wwb=mkpp(knot4,coefficients4);
```

Now we define

$$\psi_{j,k}(x) = \psi(2^j x - k), \qquad j \geq 0,\ k = 0,\dots,n_j - 3, \tag{4.17}$$

$$\psi^l_{b,j}(x) = \psi_b(2^j x), \qquad \psi^r_{b,j}(x) = \psi_b(2^j(L - x)). \tag{4.18}$$

where $n_j = 2^j L$. For the sake of simplicity, we will adopt the following notation:

$$\psi_{j,-1}(x) = \psi^l_{b,j}(x). \quad \psi_{j,n_j-2}(x) = \psi^r_{b,j}(x). \tag{4.19}$$

So, when $k = -1$ and $n_j - 1$, wavelet function $\psi_{j,k}(x)$ will denote the two boundary wavelet functions, which can not be obtained by translating and dilating $\psi(x)$.

Finally, for each $j \geq 0$, we define

$$W_j = span\psi_{j,k}(x) \mid k = -1,\dots,n_j - 2. \tag{4.20}$$

We quota the Theorem 2 in Cai's paper [1] without proof:

Theorem 4.2 *The W_j, $j \geq 0$, defined in(4.20) is the orthogonal compliment of V_j in V_{j-1} under the inner product (4.3), i.e.,* *(i)* $V_{j+1} = V_j \oplus W_j$ *for* $j \in Z^+$, *where* $\oplus$ *stands for* $V_j \perp W_j$ *under the inner product (4.3) and* $V_{j+1} = V_j + W_j$, *therefore,*

(ii) $W_j \perp W_{j+1}$, $j \in Z^+$, *and*

(iii) $H_0^2(I) = V_0 \oplus_{j \in Z^+} W_j$.

This theorem constructs an MRA in Sobolev space $H_0^2(I)$. This theorem leads to the conclusion: function $f(x) \in H_0^2(I)$ can be approximated as closely as needed by a function $f_j(x) \in V_j = V_0 \oplus W_0 \oplus \cdots \oplus W_{j-1}$ for a sufficiently large j, and $f_j(x)$ has a unique orthogonal decomposition

$$f_j(x) = f_0 + g_0 + g_1 + \cdots + g_{j-1}. \tag{4.21}$$

where $f_0 \in V_0, g_i \in W_i,\ 0 \le i \le j-1$.

4.1.1 Approximation for a function in $H^2(I)$

Consider the following two splines:

$$\eta_1(x) = \begin{cases} (1-x)^3 & 0 \le x \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{4.22}$$

$$\eta_2(x) = \begin{cases} \frac{7}{6}x^3 - 3x^2 + 2x & 0 \le x < 1 \\ -\frac{1}{6}x^3 + \frac{1}{2}x^2 - \frac{1}{2}x + \frac{1}{6} & 1 \le x < 2 \\ 0 & \text{otherwise} \end{cases} \tag{4.23}$$

For any function $f(x) \in H^2(I)$, we have $f(x) \in C^1(I)$, by the Sobolev embedding theorem. Therefore, we can define the following interpolating spline $\mathbf{I}_{b,j}f(x)$, $j \ge 0$;

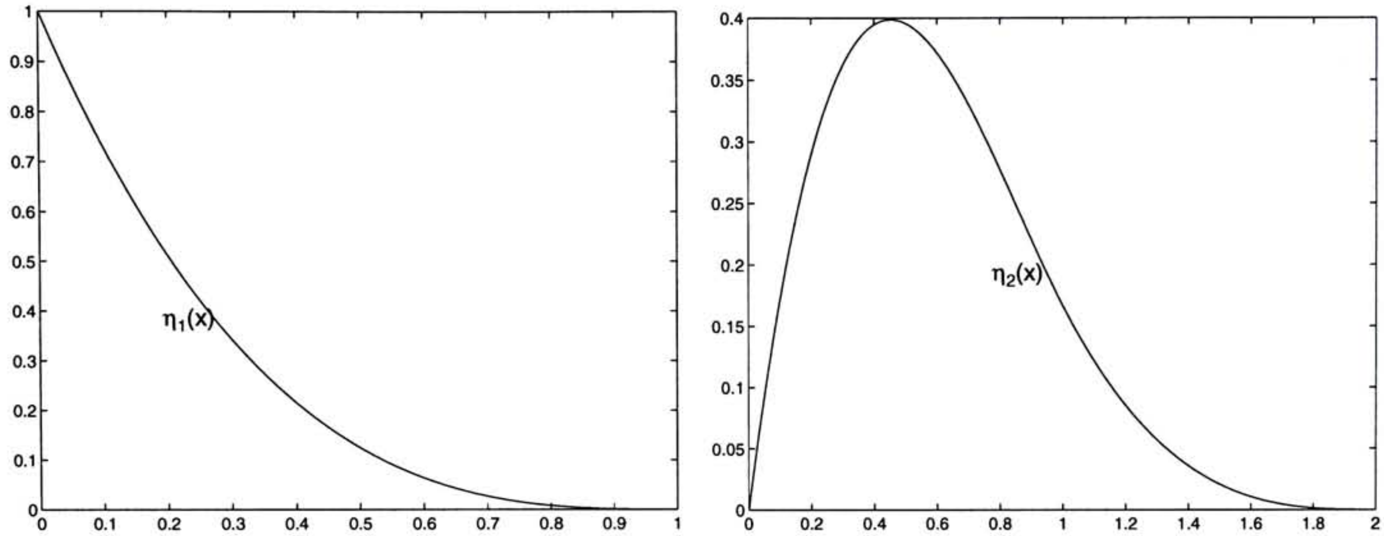
$$\mathbf{I}_{b,j}f(x) = \alpha_1\eta_1(2^j x) + \alpha_2\eta_2(2^j x) + \alpha_3\eta_1(2^j(L-x)) + \alpha_4\eta_2(2^j(L-x)). \tag{4.24}$$

where the coefficients $\alpha_1, \alpha_2, \alpha_3$ and α_4 are determined by certain interpolating conditions. Spline $\mathbf{I}_{b,j}f(x)$ is introduced to approximate the non-homogeneities of function $f(x)$ at the boundaries.

Now, for any function $f(x) \in H^2(I)$, we can find a function $f_j(x)$ in the form of

$$f_j(x) = \mathbf{I}_{b,j} f + f_0 + g_0 + g_1 + \cdots + g_{j-1}. \quad f_0 \in V_0,\ g_i \in W_i,\ 0 \le i \le j-1. \quad (4.25)$$

which approximates $f(x)$ as closely as needed provided that j is large enough. According to Cai's paper [1], the approximation will be $O(2^{-4})$ if $f_j(x)$ is chosen as the interpolating spline of $f(x)$.



(a) (b)

Figure 4.4: Boundary function for approximating function in space $H^2(I)$: (a)$\eta_1(x)$;(b)$\eta_2(x)$.

To generate $\eta_1(x)$ and $\eta_2(x)$ in Matlab, use these commands:

```
knot5=[0 1];
coefficients5=[-1 3 -3 1];

bb1=mkpp(knot5,coefficients5);
%first boundary function

knot6=[0:2];
coefficients6=[
7/6  -3  2  0;
-1/6  1/2  -1/2  1/6  ];

bb2=mkpp(knot6,coefficients6);
%second boundary function
```

These two boundary functions are shown in Figure 4.4.

4.2 Wavelet interpolation

In this section, we will introduce a wavelet interpolation method which maps discrete sample values of a function in Sobolev space $H^2(I)$ to its wavelet interpolant expansions. Such expansion with the wavelet decomposition will enable us to compute an approximation of the first and second derivatives of the function.

4.2.1 Interpolant operator $\mathbf{I}_{V0}$ in V_0

Consider any function $f(x) \in H^2(I)$ and denote the boundary and interior knots by

$$x_k^{(-1)} = k, \quad k = 0, 1, \dots, L \tag{4.26}$$

and the values of $f(x)$ on $\{x_k^{(-1)}\}_{k=0}^{L}$ by

$$f_k^{(-1)} = f(x_k^{(-1)}). \quad k = 0, 1, \dots, L. \tag{4.27}$$

The cubic interpolant $(\mathbf{I}_{v0} + \mathbf{I}_b)f(x)$ of data $\{x_k^{(-1)}\}_{k=0}^{L}$ can be expressed as

$$\begin{aligned}(\mathbf{I}_{v0} + \mathbf{I}_b)f(x) = {} & \hat{f}_{-1,-3}\eta_1(x) + \hat{f}_{-1,-2}\eta_2(x) + \hat{f}_{-1,-1}\phi_b(x) + \sum_{k=0}^{L-4} \hat{f}_{-1,k}\phi_k(x) \\ & + \hat{f}_{-1,L-3}\phi_b(L-x) + \hat{f}_{-1,L-2}\eta_2(L-x) + \hat{f}_{-1,L-1}\eta_1(L-x). \end{aligned} \tag{4.28}$$

The $(\mathbf{I}_{v0} + \mathbf{I}_b)f(x)$ interpolates data $f_k^{(-1)}$, $k = 0, 1, \dots, L$, i.e.,

$$(\mathbf{I}_{v0} + \mathbf{I}_b)f(x_k^{(-1)}) = f_k^{(-1)}, \quad k = 0, 1, \dots, L. \tag{4.29}$$

Let $\mathbf{B}$ be the transform matrix between $\mathbf{f}^{(-1)} = (f_0^{(-1)}) \ \dots \ f_L^{(-1)})^T$ and the coefficients $\hat{\mathbf{f}}^{(-1)} = (f_{-1,-3}, \dots f_{-1,L-1})^T$, i.e.,

$$\mathbf{f}^{(-1)} = \mathbf{B}\hat{\mathbf{f}}^{(-1)}, \tag{4.30}$$

where

$$\mathbf{B} = \begin{pmatrix} 1 & & & & & & & & & \\ & \frac{1}{6} & \frac{7}{12} & \frac{1}{6} & & & & & & \\ & & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & & & & & \\ & & & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & & & & \\ & & & \ddots & \ddots & \ddots & & & & \\ & & & & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & & & \\ & & & & & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & & \\ & & & & & & \frac{1}{6} & \frac{7}{12} & \frac{1}{6} & \\ & & & & & & & & & 1 \end{pmatrix} \qquad (4.31)$$

as shown in Figure 4.5.

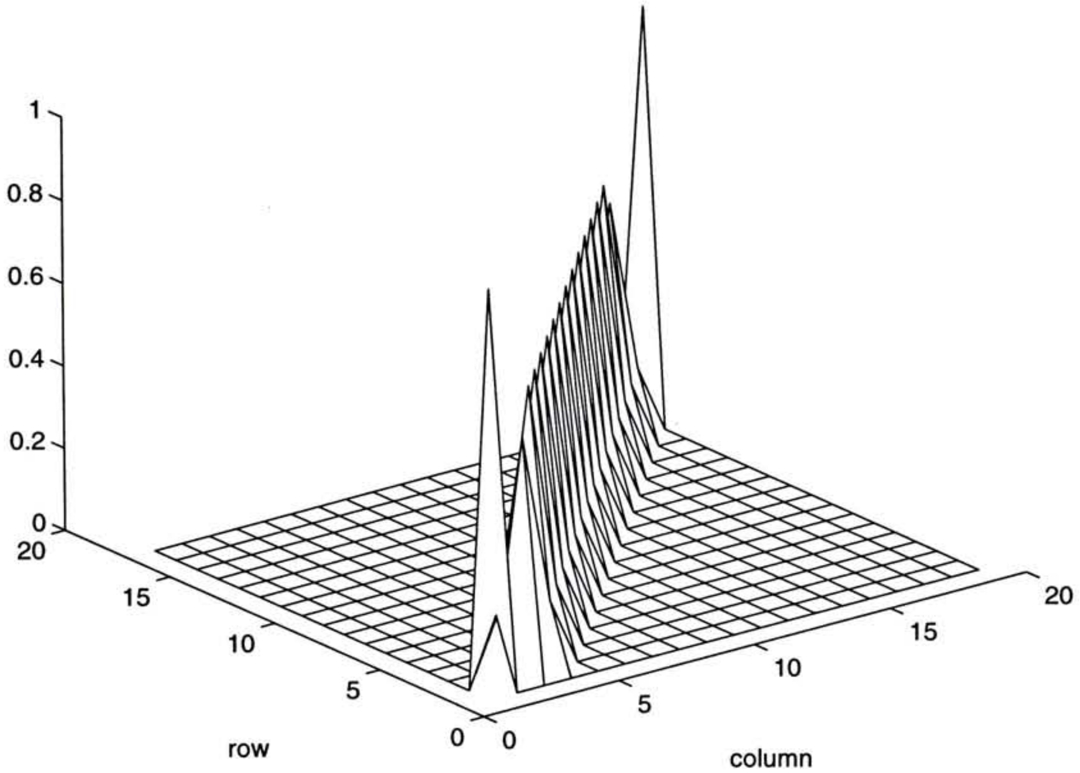


Figure 4.5: Mesh display of matrix $\mathbf{B}(L = 16$ here).

Note that the element of $\hat{\mathbf{f}}^{(-1)}$ is two more than that of $\mathbf{f}^{(-1)}$. To get $\hat{\mathbf{f}}$, we need two more equations besides equation 4.31. There are many methods to get it, for example, Not-a-knot conditions and derivative end conditions described in Cai [1]. However, it is omitted here because we are not interpolating any known function, we are going to solve device equations using collocation method. Later, we will see that the equation number is exactly equal to coefficients number in wavelet collocation method.

4.2.2 Interpolation operator $\mathbf{I}_{w_j} f$ in W_j

Similarly, we can define the interpolation operator $\mathbf{I}_{w_j} f(x)$ in $W_j, j \geq 0$, for any function $f(x)$ in $H^2(I)$. For this purpose, we choose the following interpolation points in I:

$$x_k^{(j)} = \frac{k+1.5}{2^j}. \quad -1 \leq k \leq n_j - 2, \tag{4.32}$$

where $n_j = 2^j L$. These interpolation points in different spaces are illustrated in Figure 4.6.

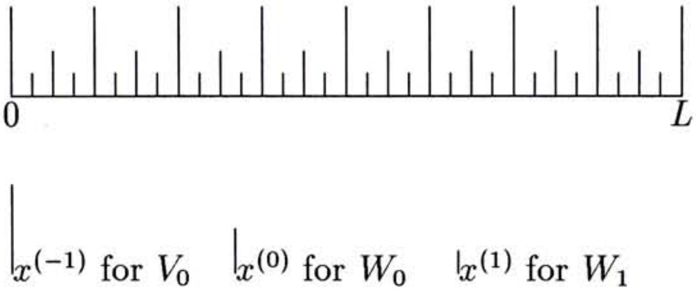


Figure 4.6: Illustration of interpolation points in different spaces($L = 8$ here).

The interpolation $\mathbf{I}_{w_j} f(x)$ of a function $f(x) \in H_0^2(I)$ in $W_j, j \geq 0$, can be expressed as a linear combination of $\psi_{j,k}(x)$, $k = -1, \ldots, n_j - 2$:

$$\mathbf{I}_{w_j} f(x) = \sum_{k=-1}^{n_j - 2} \hat{f}_{j,k} \psi_{j,k}(x). \tag{4.33}$$

Now let us assume that the values of a function $f(x) \in H^2(I)$ are given on all the interpolation points $\{x_k^j\}$ defined in (4.26) and (4.32). We intend to find the wavelet interpolation $\mathcal{P}_J f(x) \in V_0 \oplus W_0 \oplus W_1 \cdots \oplus W_{J-1}$ for $J-1 \leq 0$, i.e.,

$$\begin{aligned}
\mathcal{P}_J f(x) &= \hat{f}_{-1,-3}\eta_1(x) + \hat{f}_{-1,-2}\eta_2(x) + \hat{f}_{-1,-1}\phi_b(x) + \sum_{k=0}^{L-4} \hat{f}_{-1,k}\phi_k(x) \\
&\quad + \hat{f}_{-1,L-3}\phi_b(L-x) + \hat{f}_{-1,L-2}\eta_2(L-x) + \hat{f}_{-1,L-1}\eta_1(L-x) \\
&\quad + \sum_{j=0}^{J-1} \left[\sum_{k=-1}^{n_j-2} \hat{f}_{j,k}\psi_{j,k}(x) \right] \\
&= f_{-1}(x) + \sum_{j=0}^{J-1} f_j(x), \qquad (4.34)
\end{aligned}$$

where

$$f_{-1}(x) = (\mathbf{I}_{v0} + \mathbf{I}_b) f(x), \quad f_j(x) = \sum_{k=-1}^{n_j-2} \hat{f}_{j,k}\psi_{j,k}(x) \in W_j, j \geq 0, \qquad (4.35)$$

and the following interpolating conditions hold:

$$\begin{aligned}
\mathcal{P}_J f(x_k^{(-1)}) &= f(x_k^{(-1)}), \qquad 0 \leq k \leq L; \qquad (4.36) \\
\mathcal{P}_J f(x_k^{(j)}) &= f(x_k^{(j)}), \qquad j \geq 0, -1 \leq k \leq n_j - 2. \qquad (4.37)
\end{aligned}$$

Let us denote the values of $f(x)$ on all interpolation points by

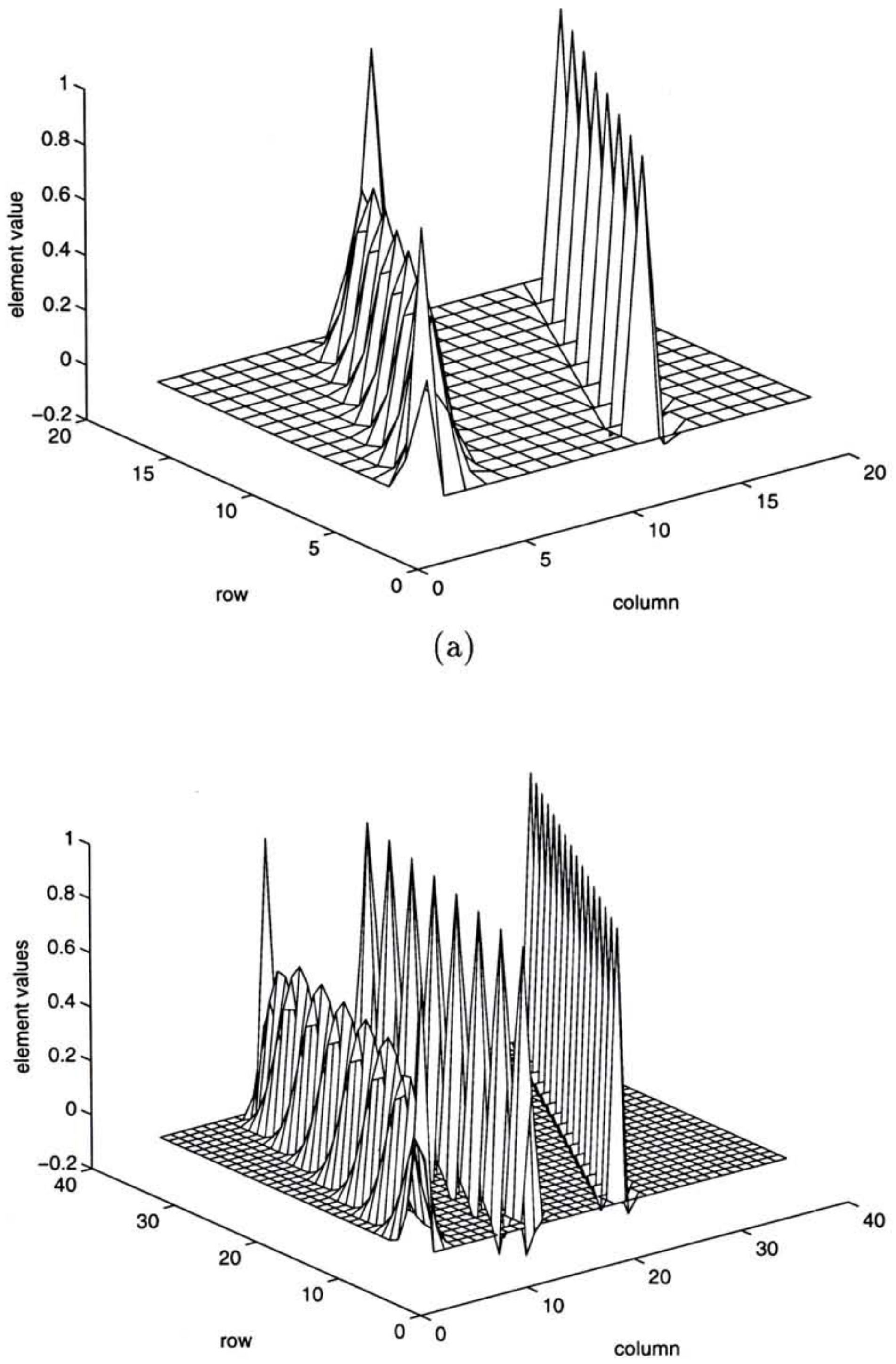
$$\mathbf{f} = (f(0), f(\Delta x), f(2\Delta x), \ldots, f(L))^T, \quad \Delta x = 2^{-(J+1)}L,$$

and denote $\hat{\mathbf{f}} = (\hat{\mathbf{f}}^{(-1)}, \hat{\mathbf{f}}^{(0)}, \ldots, \hat{\mathbf{f}}^{(J-1)})^T$, where:

$$\begin{aligned}
\hat{\mathbf{f}}^{(-1)} &= \{f_{-1,k}\}_{k=-3}^{L-1}, \\
\hat{\mathbf{f}}^{(j)} &= \{f_{j,k}\}_{k=-3}^{n_j-2}, \qquad j \geq 0.
\end{aligned}$$

We define matrix $\mathbf{D}^0$, which transform $\hat{\mathbf{f}}$ to $\mathbf{f}$, i.e.,

$$\mathbf{f} = \mathbf{D}^0 \hat{\mathbf{f}}. \qquad (4.38)$$



(a)

(b)

Figure 4.7: The mesh of matrix $\mathbf{D}^0$ for $L = 8$, and (a) $J = 0$; (b) $J = 1$.

where

$$\mathbf{D}^0 = \begin{pmatrix} \eta_1(0) & \cdots & \phi_k(0) & \cdots & \psi_{1,k}(0) & \cdots & \psi_{j,k}(0) & \cdots & \psi_{J-1,n_J-2}(0) \\ \eta_1(\Delta x) & \cdots & \phi_k(\Delta x) & \cdots & \psi_{1,k}(\Delta x) & \cdots & \psi_{j,k}(\Delta x) & \cdots & \psi_{J-1,n_J-2}(\Delta x) \\ \eta_1(2\Delta x) & \cdots & \phi_k(2\Delta x) & \cdots & \psi_{1,k}(2\Delta x) & \cdots & \psi_{j,k}(2\Delta x) & \cdots & \psi_{J-1,n_J-2}(2\Delta x) \\ \eta_1(3\Delta x) & \cdots & \phi_k(3\Delta x) & \cdots & \psi_{1,k}(3\Delta x) & \cdots & \psi_{j,k}(3\Delta x) & \cdots & \psi_{J-1,n_J-2}(3\Delta x) \\ \vdots & \vdots & \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ \eta_1(L) & \cdots & \phi_k(L) & \cdots & \psi_{1,k}(L) & \cdots & \psi_{j,k}(L) & \cdots & \psi_{J-1,n_J-2}(L) \end{pmatrix}, \tag{4.39}$$

and $\Delta x = 2^{-(J+1)}L$. The dimension of matrix $\mathbf{D}^0$ is:

$$row = L + 1 + \sum_{j=0}^{j=J} 2^j L,$$

$$column = L + 3 + \sum_{j=0}^{j=J} 2^j L.$$

Figure 4.7 illustrate two examples of the mesh of matrix $\mathbf{D}^0$ for $L = 8, J = 1$ and $L = 8, J = 1$, which are 17×19 and 33×35 matrices respectively.

In next section, we will describe matrices which transform the wavelet coefficients to derivatives of functions at collocation points. Note that the superscript of $\mathbf{D}^0$ means that the matrix transforms wavelets coefficients to 0th derivative of function at collocation points.

Theorem 4.3 *Let $f(x) \in H_0^2(0, L), M = max_I(f(x))$, and $\mathbf{I}_{wj}f(x)$ be its interpolation in W_j defined in (4.33). If for $\epsilon > 0, -1 \leq k_1 < k_2 \leq n_j - 2$,*

$$|f(x_k^{(j)})| \leq \epsilon \quad for\ k_1 \leq k \leq k_2,$$

defining

$$\tilde{\mathbf{I}}_{wj}f(x) = \sum_{-1 \leq k \leq n_j - 2, k \bar{\in} [k_1 + l, k_2 - l]} \hat{f}(x)_{j,k} \psi j, k(x), \tag{4.40}$$

where $l = l(\epsilon) = \min(\frac{n_j}{2}, -\frac{\log \epsilon}{\log \alpha})$, then we have

$$|\tilde{\mathbf{I}}_{wj}f(x) - \mathbf{I}_{wj}f(x)| \leq C(M)\epsilon. \tag{4.41}$$

where $C(M) = \frac{6K}{\alpha - 1}(\alpha + M)$. $K = 1.1726$, and $\alpha = 7 + \sqrt{192} \doteq 13.928$.

For the proof, please refer to [1].

As a consequence of Theorem 4.3, the coefficients $\hat{f}_{j,k}$ of the wavelet interpolation operator $\mathbf{I}_J f(x)$ can be ignored if the magnitudes of function $f(x)$ at points $x_k^{(j)} \in [x_{k_1+l}^{(j)}, x_{k_2+l}^{(j)}]$ are less than some given error tolerance ϵ. As J becomes larger, the coefficients $\hat{f}_{j,k}$ of the wavelet expansion will be less than the error tolerance ϵ in a larger region. Then many terms of $\psi_{j,k}(x)$ can be discarded in the wavelet expansion of $f(x)$. We will use this fact to achieve adaptivity for the solution of PDEs.

4.3 Derivative Matrices

Because a function $u(x)$, $u(x) \in H^2(I)$ can be represented by its wavelet coefficients (4.34), its derivatives can be calculated from the wavelet coefficients too.

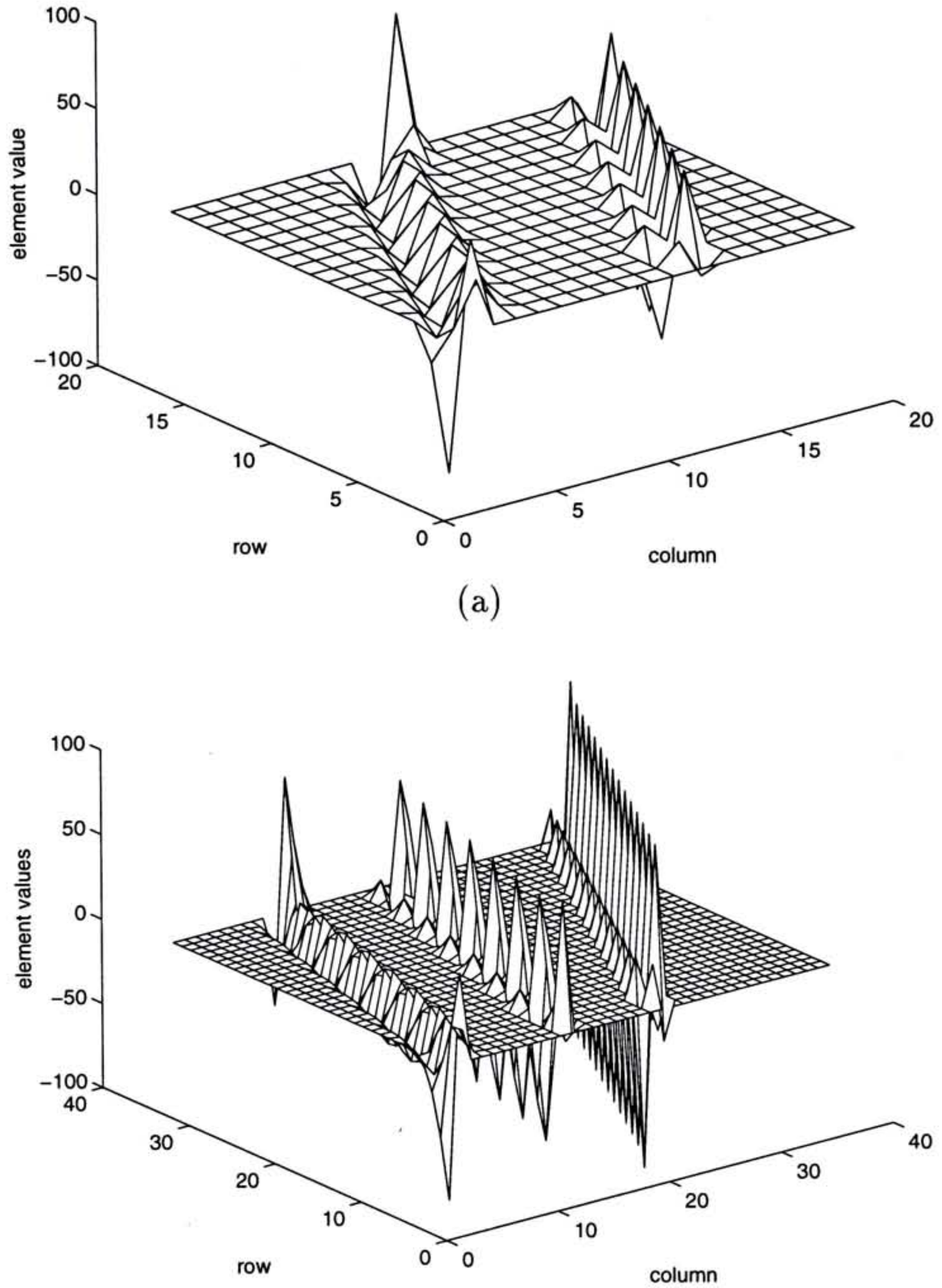
4.3.1 First derivative matrix

Recall that

$$\begin{aligned}
\mathcal{P}_J f(x) &= \hat{f}_{-1,-3}\eta_1(x) + \hat{f}_{-1,-2}\eta_2(x) + \hat{f}_{-1,-1}\phi_b(x) + \sum_{k=0}^{L-4} \hat{f}_{-1,k}\phi_k(x) \\
&\quad + \hat{f}_{-1,L-3}\phi_b(L-x) + \hat{f}_{-1,L-2}\eta_2(L-x) + \hat{f}_{-1,L-1}\psi_{1,k}(x) \\
&\quad + \sum_{j=0}^{J-1}\left[\sum_{k=-1}^{n_j-2} \hat{f}_{j,k}\psi_{j,k}(x)\right].
\end{aligned}$$

Therefore,

$$\begin{aligned}
\frac{d}{dx}\mathcal{P}_J f(x) &= \hat{f}_{-1,-3}\frac{d}{dx}\eta_1(x) + \hat{f}_{-1,-2}\frac{d}{dx}\eta_2(x) + \hat{f}_{-1,-1}\frac{d}{dx}\phi_b(x) + \sum_{k=0}^{L-4} \hat{f}_{-1,k}\frac{d}{dx}\phi_k(x) \\
&\quad + \hat{f}_{-1,L-3}\frac{d}{dx}\phi_b(L-x) + \hat{f}_{-1,L-2}\frac{d}{dx}\eta_2(L-x) + \hat{f}_{-1,L-1}\frac{d}{dx}\eta_1(L-x) \\
&\quad + \sum_{j=0}^{J-1}\left[\sum_{k=-1}^{n_j-2} \hat{f}_{j,k}\frac{d}{dx}\psi_{j,k}(x)\right]. \qquad (4.42)
\end{aligned}$$



(a)

(b)

Figure 4.8: The mesh of matrix $\mathbf{D}^1$ for $L = 8$, and (a) $J = 0$, (b)$J = 1$.

Let us denote the values of $\frac{d}{dx}f(x)$ on all interpolation points by

$$\mathbf{f_x} = \left(\frac{df(x)}{dx}|_{x=0}, \frac{df(x)}{dx}|_{x=\Delta x}, \frac{df(x)}{dx}|_{x=2\Delta x}, \dots, \frac{df(x)}{dx}|_{x=L} \right)^T, \quad \Delta x = 2^{-(J+1)}L,$$

and use the notation $\hat{\mathbf{f}} = (\hat{\mathbf{f}}^{(-1)}, \hat{\mathbf{f}}^{(0)}, \dots, \hat{\mathbf{f}}^{(J-1)})^T$ for wavelet coefficients. We define matrix $\mathbf{D}^1$, which approximate the first derivative of function at interpolation points $x_{j,k}$ defined in (4.26) and (4.32), by its wavelet coefficients:

$$\mathbf{f_x} = \mathbf{D}^1 \hat{\mathbf{f}}, \tag{4.43}$$

where

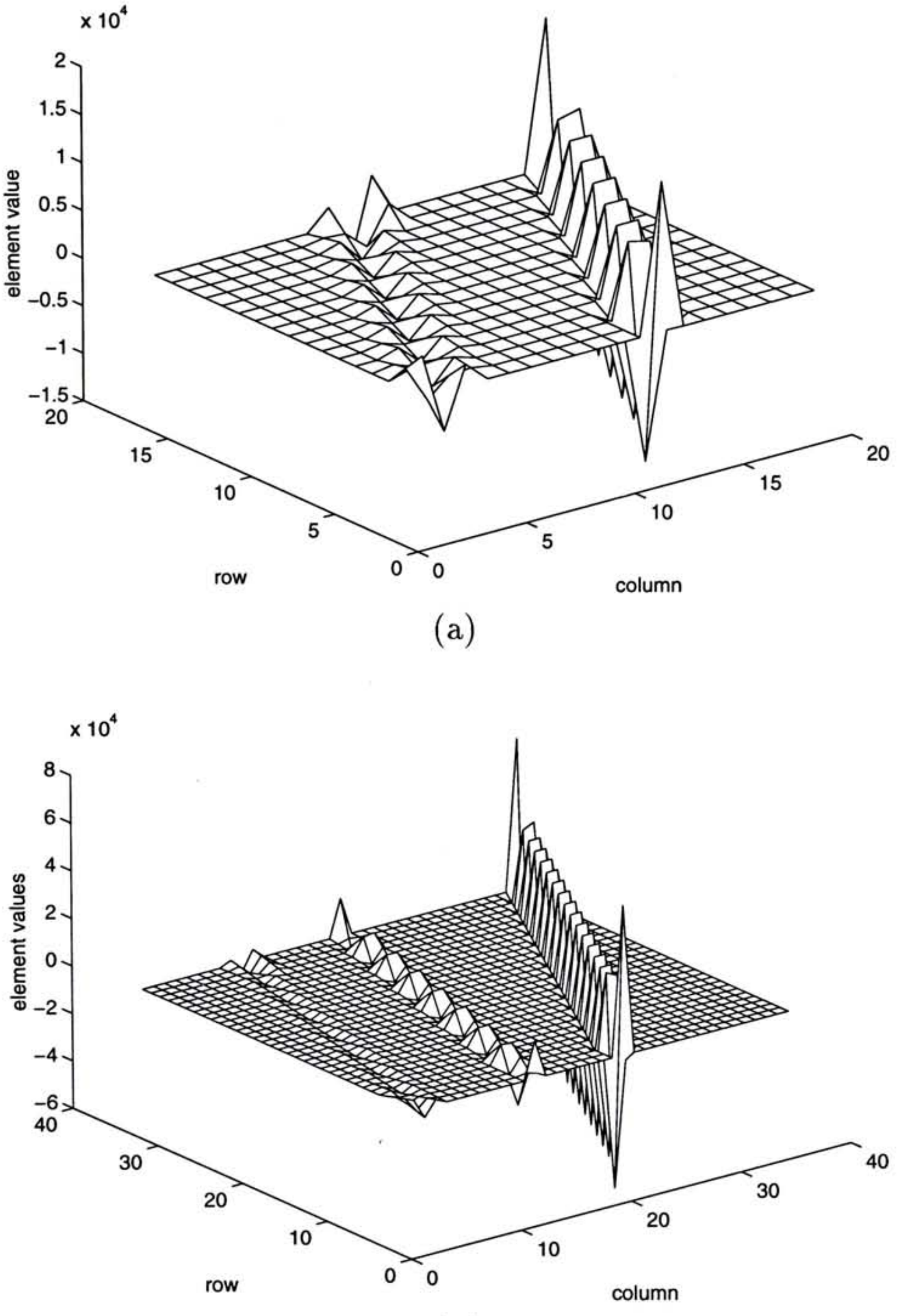
$$\mathbf{D}^1 = \begin{pmatrix} \frac{d}{dx}\eta_1(0) & \cdots & \frac{d}{dx}\phi_k(0) & \cdots & \frac{d}{dx}\psi_{1,k}(0) & \cdots & \frac{d}{dx}\psi_{j,k}(0) & \cdots & \frac{d}{dx}\psi_{J-1,n_J-2}(0) \\ \frac{d}{dx}\eta_1(\Delta x) & \cdots & \frac{d}{dx}\phi_k(\Delta x) & \cdots & \frac{d}{dx}\psi_{1,k}(\Delta x) & \cdots & \frac{d}{dx}\psi_{j,k}(\Delta x) & \cdots & \frac{d}{dx}\psi_{J-1,n_J-2}(\Delta x) \\ \frac{d}{dx}\eta_1(2\Delta x) & \cdots & \frac{d}{dx}\phi_k(2\Delta x) & \cdots & \frac{d}{dx}\psi_{1,k}(2\Delta x) & \cdots & \frac{d}{dx}\psi_{j,k}(2\Delta x) & \cdots & \frac{d}{dx}\psi_{J-1,n_J-2}(2\Delta x) \\ \frac{d}{dx}\eta_1(3\Delta x) & \cdots & \frac{d}{dx}\phi_k(3\Delta x) & \cdots & \frac{d}{dx}\psi_{1,k}(3\Delta x) & \cdots & \frac{d}{dx}\psi_{j,k}(3\Delta x) & \cdots & \frac{d}{dx}\psi_{J-1,n_J-2}(3\Delta x) \\ \vdots & \vdots & \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ \frac{d}{dx}\eta_1(L) & \cdots & \frac{d}{dx}\phi_k(L) & \cdots & \frac{d}{dx}\psi_{1,k}(L) & \cdots & \frac{d}{dx}\psi_{j,k}(L) & \cdots & \frac{d}{dx}\psi_{J-1,n_J-2}(L) \end{pmatrix} \tag{4.44}$$

with $\Delta x = 2^{-(J+1)}L$. The dimension of matrix $\mathbf{D}^1$ is same as that of $\mathbf{D}^0$. Figure 4.8 illustrate two examples of the mesh of matrix $\mathbf{D}^2$ for $L = 8, J = 0$ and $L = 8, J = 1$.

4.3.2 Second derivative matrix

Similarly,

$$\begin{aligned} \frac{d^2}{dx^2}\mathcal{P}_J f(x) &= \hat{f}_{-1,-3}\frac{d^2}{dx^2}\eta_1(x) + \hat{f}_{-1,-2}\frac{d^2}{dx^2}\eta_2(x) + \hat{f}_{-1,-1}\frac{d^2}{dx^2}\phi_b(x) + \sum_{k=0}^{L-4} \hat{f}_{-1,k}\frac{d^2}{dx^2}\phi_k(x) \\ &\quad + \hat{f}_{-1,L-3}\frac{d^2}{dx^2}\phi_b(L-x) + \hat{f}_{-1,L-2}\frac{d^2}{dx^2}\eta_2(L-x) + \hat{f}_{-1,L-1}\frac{d^2}{dx^2}\eta_1(L-x) \\ &\quad + \sum_{j=0}^{J-1}\left[\sum_{k=-1}^{n_j-2} \hat{f}_{j,k}\frac{d^2}{dx^2}\psi_{j,k}(x)\right]. \end{aligned} \tag{4.45}$$



(a)

(b)

Figure 4.9: The mesh of matrix $\mathbf{D}^2$ for $L = 8$, and (a) $J = 0$; (b) $J = 1$.

Let us denote the values of $\frac{d^2}{dx^2}f(x)$ on all interpolation points by

$$\mathbf{f_{xx}} = \left(\frac{d^2 f(x)}{dx^2}|_{x=0}, \frac{d^2 f(x)}{dx^2}|_{x=\Delta x}, \frac{d^2 f(x)}{dx^2}|_{x=2\Delta x}, \dots, \frac{d^2 f(x)}{dx^2}|_{x=L} \right)^T, \quad \Delta x = 2^{-(J+1)}L,$$

and use the notation $\hat{\mathbf{f}} = (\hat{\mathbf{f}}^{(-1)}, \hat{\mathbf{f}}^{(0)}, \dots, \hat{\mathbf{f}}^{(J-1)})^T$ for the wavelet coefficients. We define matrix $\mathbf{D}^2$, which approximate the second derivative of function at interpolation points $x_{j,k}$ defined in (4.26) and (4.32), by its wavelet coefficients:

$$\mathbf{f_{xx}} = \mathbf{D}^2 \hat{\mathbf{f}}. \tag{4.46}$$

where

$$\mathbf{D}^2 = \begin{pmatrix} \frac{d^2}{dx^2}\eta_1(0) & \cdots & \frac{d^2}{dx^2}\phi_k(0) & \cdots & \frac{d^2}{dx^2}\psi_{1,k}(0) & \cdots & \frac{d^2}{dx^2}\psi_{j,k}(0) & \cdots & \frac{d^2}{dx^2}\psi_{J-1,n_J-2}(0) \\ \frac{d^2}{dx^2}\eta_1(\Delta x) & \cdots & \frac{d^2}{dx^2}\phi_k(\Delta x) & \cdots & \frac{d^2}{dx^2}\psi_{1,k}(\Delta x) & \cdots & \frac{d^2}{dx^2}\psi_{j,k}(\Delta x) & \cdots & \frac{d^2}{dx^2}\psi_{J-1,n_J-2}(\Delta x) \\ \frac{d^2}{dx^2}\eta_1(2\Delta x) & \cdots & \frac{d^2}{dx^2}\phi_k(2\Delta x) & \cdots & \frac{d^2}{dx^2}\psi_{1,k}(2\Delta x) & \cdots & \frac{d^2}{dx^2}\psi_{j,k}(2\Delta x) & \cdots & \frac{d^2}{dx^2}\psi_{J-1,n_J-2}(2\Delta x) \\ \frac{d^2}{dx^2}\eta_1(3\Delta x) & \cdots & \frac{d^2}{dx^2}\phi_k(3\Delta x) & \cdots & \frac{d^2}{dx^2}\psi_{1,k}(3\Delta x) & \cdots & \frac{d^2}{dx^2}\psi_{j,k}(3\Delta x) & \cdots & \frac{d^2}{dx^2}\psi_{J-1,n_J-2}(3\Delta x) \\ \vdots & \vdots & \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ \frac{d^2}{dx^2}\eta_1(L) & \cdots & \frac{d^2}{dx^2}\phi_k(L) & \cdots & \frac{d^2}{dx^2}\psi_{1,k}(L) & \cdots & \frac{d^2}{dx^2}\psi_{j,k}(L) & \cdots & \frac{d^2}{dx^2}\psi_{J-1,n_J-2}(L) \end{pmatrix}, \tag{4.47}$$

with $\Delta x = 2^{-(J+1)}L$. The dimension of matrix $\mathbf{D}^2$ is same as that of $\mathbf{D}^0$. Figure 4.9 illustrate two examples of the mesh of matrix $\mathbf{D}^2$ for $L = 8, J = 0$ and $L = 8, J = 1$.

4.4 Wavelet Collocation Method for Solving Device Equations

In this section, we describe the collocation method for time-dependent device equations, and the adaptive method to utilize the spatial localization property of wavelet basis to save computational cost.

Recall from chapter 3, that the one dimensional time-dependent device equations are:

$$\begin{aligned} f_\psi &= \frac{\partial^2\psi}{\partial x^2} + (p - n + N) = 0, & (4.48)\\ f_n &= \frac{\partial^2\phi_n}{\partial x^2} + \frac{\partial\phi_n}{\partial x}\frac{\partial\psi}{\partial x} - (\frac{\partial\phi_n}{\partial x})^2 + \frac{1}{n\mu_n}\left(R(n,p) + \frac{L_D^2}{D_0}\frac{\partial n}{\partial t}\right) = 0, & (4.49)\\ f_p &= \frac{\partial^2\phi_p}{\partial x^2} + (\frac{\partial\phi_p}{\partial x})^2 - \frac{\partial\phi_p}{\partial x}\frac{\partial\psi}{\partial x} - \frac{1}{p\mu_p}\left(R(n,p) + \frac{L_D^2}{D_0}\frac{\partial p}{\partial t}\right) = 0, & (4.50) \end{aligned}$$

and the boundary conditions depend on specified devices. For an abrupt doped p-n junction, with donor concentration N_D and acceptor concentration N_A, the boundary conditions are:

$$\begin{gathered} \psi(0,t) = \ln N_D; \quad \phi_n(0,t) = \phi_p(0,t) = 0; \\ \psi(L,t) = -\ln N_A + v_b(t); \quad \phi_n(L,t) = \phi_p(0,t) = v_b(t); \end{gathered} \tag{4.51}$$

where $x \in [0, L]$ and $v_b(t)$ is applied terminal voltage.

Let's generalize device equations as:

$$\begin{cases} u_{xx} + u_t + g(u_x, u, x) + b(x) = 0. \quad x \in [0, L],\ t \geq 0 \\ u(x,0) = u_0(x); \\ u(0,t) = g_0(t), \\ u(L,t) = g_1(t). \end{cases} \tag{4.52}$$

The boundary conditions are Dirichlet type. We will use wavelet collocation method for spatial discretization and finite difference method for time discretization.

To get the transient solution, we need to solve the static solution first, i.e., calculate $u(x,0)$ first. In this step, we solve boundary value differential equations by wavelet collocation method. Once we get the static solution, we can go forward by Euler's formula.

4.4.1 Steady state solution

For steady state solution, we are solving the boundary value problem:

$$\begin{cases} f(u) = u_{xx} + g(u_x, u, x) + b(x) = 0, \quad x \in [0, L], \\ u(0) = c_1, \\ u(L) = c_2. \end{cases} \tag{4.53}$$

The idea of wavelet collocation method is to represent function $u(x)$ as linear combination of wavelet basis defined in (4.12)(4.17)(4.18)(4.22) and (4.23), and evaluate the equation exactly at collocation points defined in (4.26)(4.32) to get the wavelet coefficients. From the description of previous section, the values of the wavelet expansion of $u(x)$ at collocation points can be approximated by

$$\mathbf{u} = \mathbf{D}^0\hat{\mathbf{u}}. \tag{4.54}$$

where $\hat{\mathbf{u}}$ is the wavelet coefficients of $u(x)$. And its first and second derivatives with respect to x at collocation points can be approximated by

$$\mathbf{u_x} = \mathbf{D}^1\hat{\mathbf{u}}, \tag{4.55}$$

$$\mathbf{u_{xx}} = \mathbf{D}^2\hat{\mathbf{u}}. \tag{4.56}$$

By using these approximation, we can rewrite (4.53) as

$$\begin{cases} \mathbf{D}^2\hat{\mathbf{u}} + g(\mathbf{D}^1\hat{\mathbf{u}}, \mathbf{D}^0\hat{\mathbf{u}}) + \mathbf{b} = \mathbf{0}, \\ \hat{u}(1) = c_1, \\ \hat{u}(N) = c_2. \end{cases} \tag{4.57}$$

where N is the index corresponding to function $\xi_1(L - x)$. The boundary conditions of equation 4.57 come from the fact that only the first basis ($\eta_1(x)$) is non-zero (equal to 1) at $x = 0$, and only the last basis ($\eta_1(L - x)$) is non-zero (equal to 1) at $x = L$. Or, from boundary conditions of (4.53), we get:

$$\mathbf{D}^0\hat{\mathbf{u}}(1) = c_1, \mathbf{D}^0\hat{\mathbf{u}}(N) = c_2.$$

and from the fact that $\mathbf{D}^0(1) = [1\ 0 \ldots 0]$, $\mathbf{D}^0(1) = [0\ 0 \ldots 1]$, $\hat{u}(1) = c_1, \hat{u}(N) = c_2$.

In (4.57), the number of unknowns, i.e., element number of $\hat{\mathbf{u}}$, are $L+3+\sum_{j=1}^{J} 2^j L$. And the number of equations, which is equal to the number of collocation points plus two boundary conditions, is also $L+3+\sum_{j=1}^{J} 2^j L$.

To solve nonlinear system (4.57), we can use either Newton's method or Gummel's method as described in Chapter 3. Let's denote the solution of (4.57) by $\hat{\mathbf{u}}_0$. Subsequently, the solution $u_0(x)$ of (4.53) is $\mathbf{D}^0\hat{\mathbf{u}}_0$.

4.4.2 Transient solution

Once we get the steady state solution $u_0(x,0)$ of (4.52), we can go ahead for transient solution. We suggest backward Euler for time discretization. Based on this scheme, the time discretized formula of (4.52) is:

$$\begin{cases} f^t(u) = u^t_{xx} + \frac{u^t - u^{t-1}}{\Delta t} + g(u^t_x, u^t, x) + b^t(x) = 0. \quad x \in [0, L], t = 1, 2, 3 \ldots \\ u(x)^0 = u_0(x); \\ u^t(0) = g^t_0, \\ u^t(L) = g^t_1. \end{cases} \tag{4.58}$$

where Δt is the time step, and $t = 1, 2, 3 \ldots$ is the time index of Euler scheme.

Apply the wavelet collocation method to such boundary value problem again, we obtain the following nonlinear algebraic equations:

$$\begin{cases} \mathbf{D}^2\hat{\mathbf{u}}^t + g(\mathbf{D}^1\hat{\mathbf{u}}^{\mathbf{t}}, \mathbf{D}^0\hat{\mathbf{u}}^{\mathbf{t}}) + \frac{1}{\Delta t}\mathbf{D}^0\hat{\mathbf{u}}^t + \mathbf{b}^t - \frac{1}{\Delta t}\mathbf{D}^0\hat{u}^{t-1} = \mathbf{0}, \\ \hat{\mathbf{u}}^0 = \hat{\mathbf{u}}_0, \\ \hat{u}^t(1) = g^t_0, \\ \hat{u}^t(N) = g^t_1. \end{cases} \tag{4.59}$$

where N is the index corresponding to function $\eta_1(L-x)$, $g^t_0 = g_0(t \times \Delta t), g^t_1 = g_1(t \times \Delta t)$ and $\hat{\mathbf{u}}^{t-1}$ is the wavelet coefficients of previous time step. Simplify (4.59) as

$$\mathbf{A}\hat{\mathbf{u}}^{\mathbf{t}} + \mathbf{g}(\hat{\mathbf{u}}^{\mathbf{t}}) + \bar{\mathbf{b}}^{\mathbf{t}} = \mathbf{0}. \quad \hat{u}^t(1) = g^t_0,\ \hat{u}^t(N) = g^t_1. \tag{4.60}$$

where

$$
\begin{aligned}
\mathbf{A} &= \mathbf{D}^2 + \frac{1}{\Delta t}\mathbf{D}^0, \\
\bar{\mathbf{b}}^t &= \mathbf{b}^t - \frac{1}{\Delta t}\mathbf{D}^0\hat{\mathbf{u}}^{t-1}.
\end{aligned}
$$

Start from initial static solution $\hat{\mathbf{u}}_0$, $\hat{\mathbf{u}}^1, \hat{\mathbf{u}}^2, \hat{\mathbf{u}}^3, \ldots$ can be solved, and we get the transient solution.

4.5 Reducing Collocation Points

In this section, we will introduce a scheme to reduce collocation points and corresponding wavelet basis. Note that in (4.57) and (4.60), we use full set of wavelet basis. As discussed after theorem 4.3, most of the wavelet coefficients $\hat{u}_{j,k}$ for large j can be neglected within a given tolerance ϵ. So we can adjust the number of wavelet basis and corresponding collocation points, reducing significantly the cost of the scheme while providing enough resolution in the regions where the solution varies significantly. This procedure is equivalent to the grid point generation process in some device simulator employing other numerical methods, for instance, Finite Difference Method and Finite Element Methods.

4.5.1 Error evaluation

Suppose we have obtained the solution $u(x)$ of (4.53) up to resolution level J. In other words, we get the solution vector:

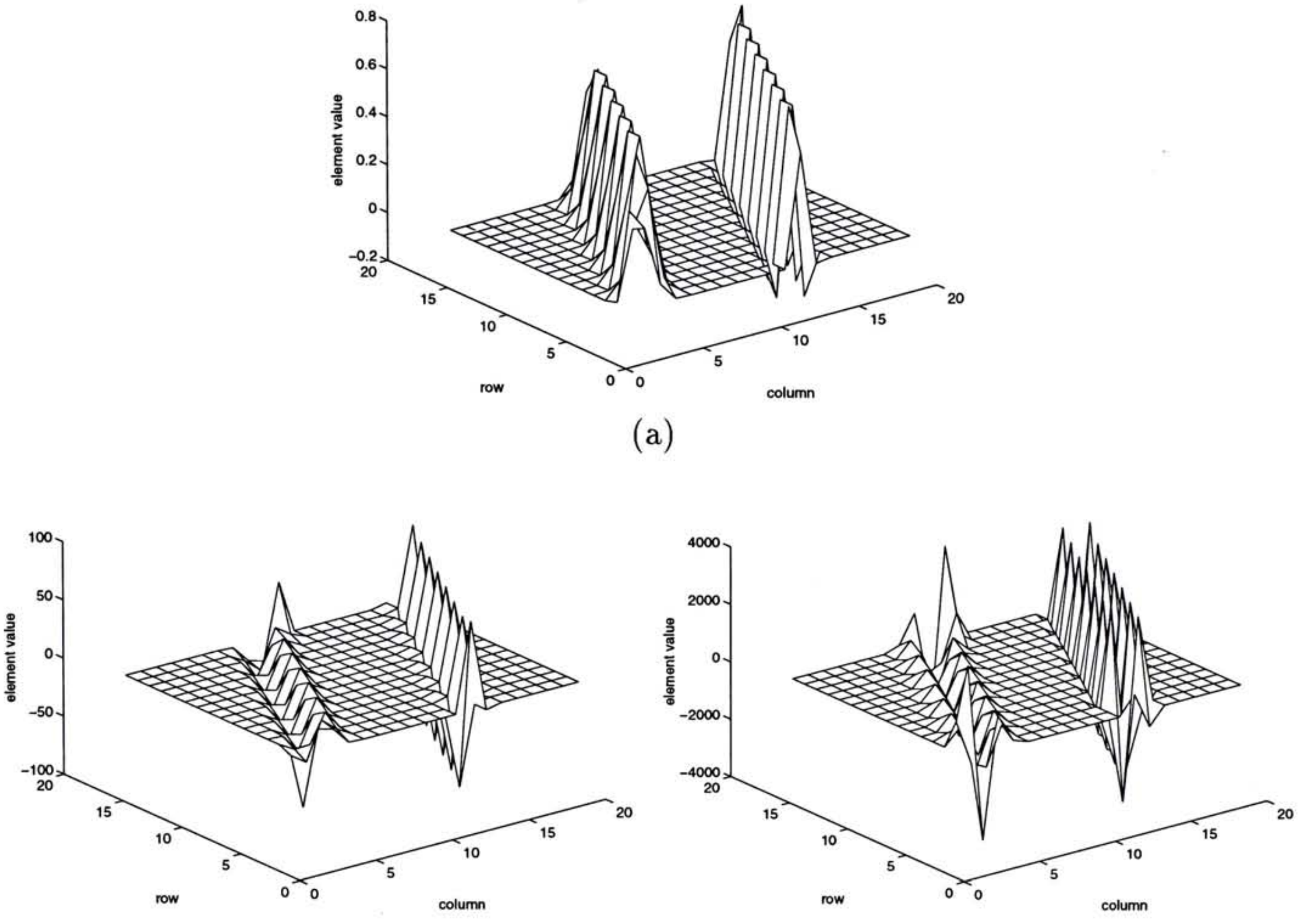
$$\hat{\mathbf{u}}_J = (\hat{\mathbf{u}}^{(-1)}, \hat{\mathbf{u}}^{(0)}, \hat{\mathbf{u}}^{(1)}, \ldots, \hat{\mathbf{u}}^{(J-1)})^T$$

so that

$$u(x) = \hat{\mathbf{u}}_J\theta(x),\ \theta(x) = \{\eta_1, \eta_2, \phi_b, \{\phi_k\}_{k=0}^{L-4}, \phi_b^r, \eta_2^r, \eta_1^r, \{\psi_k\}_{k=-1}^{n_0-2}, \ldots, \{\psi_k\}_{k=-1}^{n_{(J-1)}-2}\}$$

satisfy (4.53) exactly at collocation points

$$\{x\}_k^{(j)}, j = -1, \ldots J-1; \quad k = 0, 1, \ldots, L, \text{ if } j = -1; k = -1, \ldots, n_j - 2, \text{ if } j \geq 0.$$



(a)

(b)

(c)

Figure 4.10: The mesh of matrices (a)$\bar{\mathbf{D}}^0$; (b)$\bar{\mathbf{D}}^1$; (c)$\bar{\mathbf{D}}^2$, for $L = 8, J = 0$.

Now we want to evaluate (4.53) for solution $\hat{\mathbf{u}}_J$ at resolution level $J+1$, i.e, at collocation points $\{x\}_k^{(J)}$, $k = -1, \ldots, n_J - 2$.

Denote the values of function $u(x)$ at points $\{x\}_k^{(J)}$, $k = -1, \ldots, n_J - 2$ as $\mathbf{u}_J^{J+1}$(u expanded in level J and evaluated in level $J+1$), and the first and second derivatives of u with respective to x at these points as $\mathbf{u}'^{J+1}_J$ and $\mathbf{u}''^{J}_{J+1}$ respectively. $\mathbf{u}_J^{J+1}, \mathbf{u}'^{J+1}_J$ and $\mathbf{u}''^{J+1}_J$ can be calculated from $\hat{u}_J$ by some transform matrices:

$$\begin{aligned} \mathbf{u}_J^{J+1} &= \bar{\mathbf{D}}^0 \hat{\mathbf{u}}_J, \\ \mathbf{u}'^{J+1}_J &= \bar{\mathbf{D}}^1 \hat{\mathbf{u}}_J, \\ \mathbf{u}''^{J+1}_J &= \bar{\mathbf{D}}^2 \hat{\mathbf{u}}_J, \end{aligned}$$

where

$$\bar{\mathbf{D}}^0 = \begin{pmatrix} \eta_1(\Delta x) & \cdots & \phi_k(\Delta x) & \cdots & \psi_{1,k}(\Delta x) & \cdots & \psi_{j,k}(\Delta x) & \cdots & \psi_{J-1,n_J-2}(\Delta x) \\ \eta_1(3\Delta x) & \cdots & \phi_k(3\Delta x) & \cdots & \psi_{1,k}(3\Delta x) & \cdots & \psi_{j,k}(3\Delta x) & \cdots & \psi_{J-1,n_J-2}(3\Delta x) \\ \eta_1(5\Delta x) & \cdots & \phi_k(5\Delta x) & \cdots & \psi_{1,k}(5\Delta x) & \cdots & \psi_{j,k}(5\Delta x) & \cdots & \psi_{J-1,n_J-2}(5\Delta x) \\ \eta_1(7\Delta x) & \cdots & \phi_k(7\Delta x) & \cdots & \psi_{1,k}(7\Delta x) & \cdots & \psi_{j,k}(7\Delta x) & \cdots & \psi_{J-1,n_J-2}(7\Delta x) \\ \vdots & \vdots & \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ \eta_1(L-\Delta x) & \cdots & \phi_k(L-\Delta x) & \cdots & \psi_{1,k}(L-\Delta x) & \cdots & \psi_{j,k}(L-\Delta x) & \cdots & \psi_{J-1,n_J-2}(L-\Delta x) \end{pmatrix}, \tag{4.61}$$

and $\Delta x = 2^{-(J+2)}L$. The definitions of $\mathbf{u}'^{J+1}_J$ and $\mathbf{u}''^{J+1}_J$ are the same as $\mathbf{u}^{J+1}_J$ but performing first and second differentiations of all the basis functions respectively. Figure 4.10 shows meshes of these matrices for $L = 8, J = 0$. Practically, these transform matrices can be easily obtained, by subtracting even rows and first half columns from pre-calculated full transform matrices $\mathbf{D}^0, \mathbf{D}^1$ and $\mathbf{D}^2$ for resolution level $J + 1$.

Then, we can evaluate (4.53) at collocation points $\{x\}^{(J)}_k$, $k = -1, \dots, n_J - 2$ by following matrix system:

$$\mathbf{f} = \bar{\mathbf{D}}^2\hat{\mathbf{u}} + g(\bar{\mathbf{D}}^1\hat{\mathbf{u}}, \bar{\mathbf{D}}^0\hat{\mathbf{u}}) + \mathbf{b}. \tag{4.62}$$

In (4.62), $\mathbf{f}$ represents the vector $\{f(x^J_k)\}$, $k = -1, \dots, n^J - 2$.If

$$\{f(x^J_k)\} \le \epsilon, \text{ for all } k = -1, \dots, n^J - 2, \tag{4.63}$$

where ϵ is a predefined tolerance, we stop moving up to resolution level $J + 1$, else we will move up and adopt a scheme to reduce the collocation points and the corresponding wavelet basis as will be described in the following sub-section.

4.5.2 Deleting collocation points

The next step is to delete some collocation points where scaling functions and wavelet basis up to resolution level J have provided good enough approximation to the solution of equation (4.53).

We locate the range for the index k,

$$(k_1', l_l') \dots (k_m', l_m'). \quad m = m(j, \epsilon), \tag{4.64}$$

such that

$$f(x_k^J) \le \epsilon. \quad k_i' \le k \le l_i', \ i = 1 \dots m. \tag{4.65}$$

This means that $u(x) = \hat{\mathbf{u}}_J \theta(x)$ gives good enough approximation of (4.53) at collocation points $\{x\}_k^J$, $k_i' \le k \le l_i'$, $i = 1 \dots m$. Remember the support of $\psi(x)$ is $[0, 3]$. Therefore, when we go up to resolution level $J + 1$, we can ignore $\hat{u}_{J,k}$ for $k_i \le k \le l_i$, $i = 1 \dots m$, $k_i = k_i' + 3, l_i = l_i' - 3$, namely, we redefine $u_J(x)$ as

$$u_J(x) := \sum_{-1 \le k \le n_J - 2, k \bar{\in} \mathcal{K}_J} \hat{u}_{J,k} \psi_{J,k}(x), \tag{4.66}$$

where $\mathcal{K}_J = \bigcup_{1 \le i \le m} [k_i, l_i]$.

The new collocation points and unknowns in resolution level $J + 1$ will be

$$\{x_k^{(J)}\}, \hat{u}_{J,k}, \quad k \in \{-1, \dots, n_J - 2\}/\mathcal{K}_J. \tag{4.67}$$

So, new transform matrices $\mathbf{D}^0, \mathbf{D}^1$ and $\mathbf{D}^2$ up to level $J+1$ have rows corresponding to collocation points

$$\{\{x_k^{(-1)}\}, \{x_k^{(0)}\}_{-1 \le k \le n_0 - 2, k \bar{\in} / \mathcal{K}_j}, \dots, \{x_k^{(J)}\}_{-1 \le k \le n_J - 2, k \bar{\in} / \mathcal{K}_J}\}, \tag{4.68}$$

and columns corresponding to wavelet coefficients

$$\hat{\mathbf{u}} = (\hat{\mathbf{u}}^{(-1)}, \hat{\mathbf{u}}^{(0)}, \dots, \hat{\mathbf{u}}^{(J-1)})^T, \tag{4.69}$$

where

$$\begin{aligned} \hat{\mathbf{u}}^{(-1)} &= \{u_{-1,k}\}_{k=-3}^{L-1}, \\ \hat{\mathbf{u}}^{(j)} &= \{u_{j,k}\}_{-1 \le k \le n_j - 2, k \bar{\in} / \mathcal{K}_j}, \qquad j \ge 0. \end{aligned}$$

Practically, these reduced transform matrices can be easily obtained, by subtracting certain rows and columns from pre-calculated full transform matrices $\mathbf{D}^0, \mathbf{D}^1$ and $\mathbf{D}^2$, which are corresponding to row index (4.68) and column index (4.69).

These reduced transform matrices are used to solve (4.53) in resolution level $J+1$. And then we repeat the error evaluation and collocation point reducing processes for $J \leftarrow J+1$, until (4.63) is satisfied or J reaches a predefined maximum value.

Finally, we obtain new transform matrices,$\mathcal{D}^0, \mathcal{D}^1, \mathcal{D}^2$, which are used for transient equation (4.60):

$$\begin{cases} \mathcal{D}^2\hat{\mathbf{u}}^t + g(\mathcal{D}^1\hat{\mathbf{u}}^{\mathbf{t}}, \mathcal{D}^0\hat{\mathbf{u}}^{\mathbf{t}}) + \frac{1}{\Delta t}\mathcal{D}^0\hat{\mathbf{u}}^t + \mathbf{b}^t - \frac{1}{\Delta t}\mathcal{D}^0\hat{u}^{t-1} = \mathbf{0}, \\ \hat{\mathbf{u}}^0 = \hat{\mathbf{u}}_0, \\ \hat{u}^t(1) = g_0^t, \\ \hat{u}^t(N) = g_1^t. \end{cases} \tag{4.70}$$

where $\hat{\mathbf{u}} = (\hat{\mathbf{u}}^{(-1)}, \hat{\mathbf{u}}^{(0)}, \ldots, \hat{\mathbf{u}}^{(J-1)})^T$ is defined in (4.69), and $N = dim\ \hat{u}$. Since the sizes of these transform matrices will be reduced significantly for large j, we will save a large amount of computional cost as will be seen in some examples.

From the description above, we see that the collocation points generation scheme of wavelet collocation method is very nature and easy to implement. In next chapter, we will discuss in detail on some examples. One of these examples is abrupt dopped $p-n$ junction. We will see that, a large portion of wavelet basis are allocated in the region around the physical junction where sharp transition of device variables including electrostatic potential, carrier densities occurs, and only necessary basis are allocated in regions far away from the physical junction where device variables change smoothly. Completely utilizing the localization property of wavelets to save computational resources is the advantage of adaptive wavelet collocation method.

Chapter 5

Numerical Results

In this chapter, we will illustrate two examples. One is the simplest semiconductor device: p—n junction diode, and the other is a bipolar transistor. First, we will give the equilibrium solution, together with the illustration of grid generation process. After that, we will give the transient solutions under certain terminal voltages. Althought only two examples are illustrated, our method can be easily applied to other devices subject to the validity of drift-diffusion model.

5.1 P-N Junction Diode

The first example we carried out is an abrupt doped $p - n$ junction diode. We choose abrupt doped diode but not linear or gradient doped diode because the former represents the numerical worst case, and it contains rapid changing information which is suitable for the proposed simulation method.

The diagram of the selected diode is shown in Figure 5.1, and its physical parameters are listed in Table 5.1.

Recall from Chapter 3, that the one dimensional time-dependent device equations

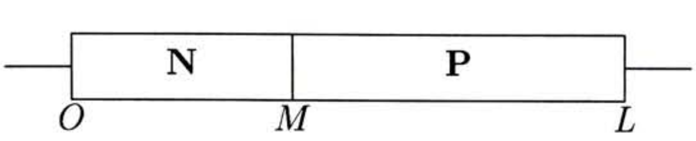


Figure 5.1: A one dimensional $p - n$ junction diode

Table 5.1: Physical parameters of a diode.

Material	Silicon (relative permittivity = 11.8)
Temperature	$300^0 K$
Intrinsic Impurity Concentration(n_i)	$1.5 \times 10^{10}/cm^3$
Doping	Donar: $10^6 (or 1.5 \times 10^{16} cm^{-3})$ $10^4 (or 1.5 \times 10^{14} cm^{-3})$
Intrinsic Debye Length($\sqrt{\frac{V_t \epsilon}{q n_i}}$)	$0.00335618 cm$ or $33.5618 \mu m$
Length	N-side, M-O=0.2 (or $6.7 \mu m$) P-side, L-M=0.4 (or $13.4 \mu m$)
Carrier mobilities	electron, $1500 cm^2/V - sec$ hole, $450 cm^2/V - sec$
Minority life time	electron $\tau_n = 10^{-4} sec$ hole $\tau_p = 10^{-5} sec$

are:

$$\begin{aligned} f_\psi &= \frac{\partial^2 \psi}{\partial x^2} + (p - n + N(x)) = 0, & (5.1) \\ f_n &= \frac{\partial^2 \phi_n}{\partial x^2} + \frac{\partial \phi_n}{\partial x}\frac{\partial \psi}{\partial x} - (\frac{\partial \phi_n}{\partial x})^2 + \frac{1}{n\mu_n}\left(R(n,p) + \frac{L_D^2}{D_0}\frac{\partial n}{\partial t}\right) = 0, & (5.2) \\ f_p &= \frac{\partial^2 \phi_p}{\partial x^2} + (\frac{\partial \phi_p}{\partial x})^2 - \frac{\partial \phi_p}{\partial x}\frac{\partial \psi}{\partial x} - \frac{1}{p\mu_p}\left(R(n,p) + \frac{L_D^2}{D_0}\frac{\partial p}{\partial t}\right) = 0, & (5.3) \end{aligned}$$

with boundary conditions

$$\begin{aligned} \psi(0,t) &= \ln N_D; \quad \phi_n(0,t) = \phi_p(0,t) = 0; \\ \psi(L,t) &= -\ln N_A + v_b(t); \quad \phi_n(L,t) = \phi_p(0,t) = v_b(t); \end{aligned}$$

where $n = \exp(\psi - \phi_n)$, $p = \exp(\phi_p - \psi)$, $v_b(t)$ is the applied terminal voltages of diode and

$$R = \frac{L_D^2}{D_0}\frac{np - 1}{\tau_p(n+1) + \tau_n(p+1)}. \tag{5.4}$$

Remember in the wavelet collocation method described in Chapter 4, the interval length L should be an integer larger than 4. We further normalize the x coordinate by factor $\frac{2^{J_0}}{L}$:

$$x \leftarrow \frac{2^{J0}}{L} x, \quad J0 \text{ is an integer larger than 2,}$$

so that the interval becomes $[0, 2^{J0}]$. Therefore, derivative matrices $\mathbf{D^0}, \mathbf{D^1}$ and $\mathbf{D^2}$ should be updated as:

$$\begin{aligned}
\mathbf{D^0} &\leftarrow \mathbf{D^0}, \\
\mathbf{D^1} &\leftarrow \left(\frac{2^{J0}}{L}\right)\mathbf{D^1}, \\
\mathbf{D^2} &\leftarrow \left(\frac{2^{J0}}{L}\right)^2\mathbf{D^2}.
\end{aligned}$$

Using these transform matrices for spatial discretization and finite difference method for time discretization as described in Chapter 4, we obtain:

$$\mathbf{f}_\psi = \mathbf{D^2}\hat{\psi}^{\mathbf{t}} + \mathbf{p^t} - \mathbf{n^t} + \mathbf{N} = 0, \quad (5.5)$$

$$\mathbf{f_n} = \mathbf{D^2}\hat{\phi_{\mathbf{n}}}^t + \mathbf{D^1}\hat{\phi_{\mathbf{n}}}^t\mathbf{D^1}\hat{\psi}^t - \left(\mathbf{D^1}\hat{\phi}_n^t\right)^2 + \frac{1}{\mathbf{n}^t\mu_n}\left(\mathbf{R} + \frac{L_D^2}{D_0}\frac{\mathbf{n}^t - \mathbf{n}^{t-1}}{\Delta t}\right) = 0, (5.6)$$

$$\mathbf{f_p} = \mathbf{D^2}\hat{\phi_{\mathbf{p}}}^t - \mathbf{D^1}\hat{\phi_{\mathbf{p}}}^t\mathbf{D^1}\hat{\psi}^t + \left(\mathbf{D^1}\hat{\phi}_p^t\right)^2 - \frac{1}{\mathbf{p}^t\mu_p}\left(\mathbf{R} + \frac{L_D^2}{D_0}\frac{\mathbf{p}^t - \mathbf{p}^{t-1}}{\Delta t}\right) = 0, (5.7)$$

with boundary conditions

$$\begin{aligned}
\hat{\psi}^t(1) &= \ln N_D; \quad \hat{\phi}_n^t(1) = \hat{\phi}_p^t(1) = 0; \\
\hat{\psi}(N) &= -\ln N_A + v_b(t); \quad \hat{\phi}_n^t(N) = \hat{\phi}_p^t(N) = v_b(t);
\end{aligned}$$

(N is the index corresponding to basis function $\eta_1(L-x)$)

where

superscript $t = 1, 2, 3, \ldots$ are time index,

$\hat{\psi}, \hat{\phi}_n, \hat{\phi}_p$ are wavelet coefficients up to resolution level J of ψ, ϕ_n, ϕ_p respectively,

$dim\ \hat{\psi} = dim\ \hat{\phi}_n = dim\ \hat{\phi}_p = L + 3 + \sum_{j=0}^{J-1} 2^j L = 2^J L + 3$,

$dim\ \mathbf{f}_\psi = dim\ \mathbf{f_n} = dim\ \mathbf{f_p} = 2^J L + 1$,

$\mathbf{n} = exp(\mathbf{D^0}(\hat{\psi} - \hat{\phi}_{\mathbf{n}}))$,

$\mathbf{p} = exp(\mathbf{D^0}(\hat{\phi}_{\mathbf{p}} - \hat{\psi}))$, and **R** is given by (5.4). Note that the vector products in (5.6)(5.7) represent element by element product, and vector devisions mean element by element devision.

When using Gummel's method as described in Chapter 3 to solve algebraic equations (5.5)(5.6)(5.7), we need to evaluate Jacobian $\left[\frac{\partial \mathbf{f}_\psi}{\partial \hat{\psi}}\right]$, $\left[\frac{\partial \mathbf{f}_\mathbf{n}}{\partial \hat{\phi}_\mathbf{n}}\right]$ and $\left[\frac{\partial \mathbf{f}_\mathbf{p}}{\partial \hat{\phi}_\mathbf{p}}\right]$. There are two methods to evaluate these Jacobian. For very small $\Delta\hat{\psi}$, we can evaluate $\frac{\partial \mathbf{f}_\psi}{\partial \hat{\psi}}$ by

$$\frac{\partial \mathbf{f}_\psi}{\partial \hat{\psi}} \simeq \frac{f_\psi(\hat{\psi}+\Delta\hat{\psi}) - f_\psi(\hat{\psi})}{\Delta\hat{\psi}}.$$

Evaluation of $\left[\frac{\partial \mathbf{f}_\mathbf{n}}{\partial \hat{\phi}_\mathbf{n}}\right]$ and $\left[\frac{\partial \mathbf{f}_\mathbf{p}}{\partial \hat{\phi}_\mathbf{p}}\right]$ are in the same way. We call it perturbation method. The second method is to write explicit expressions of $\left[\frac{\partial \mathbf{f}_\psi}{\partial \hat{\psi}}\right]$, $\left[\frac{\partial \mathbf{f}_\mathbf{n}}{\partial \hat{\phi}_\mathbf{n}}\right]$ and $\left[\frac{\partial \mathbf{f}_\mathbf{p}}{\partial \hat{\phi}_\mathbf{p}}\right]$. The main advantage of perturbation method is that it needs no extra programming (easy to implement), and the main disadvantage is that it needs much longer time than explicit method. On the other side, the advantages of explicit method is its high computational speed, and the disadvantages are that it is difficult to write explicit form in many case, and difficult to program. In this example, we will use the explicit method because of its speed. The explicit forms of Jacobian of (5.5)(5.6)(5.7) will be described in following paragraphs.

The explicit expressions of $\left[\frac{\partial \mathbf{f}_\psi}{\partial \hat{\psi}^t}\right]$, $\left[\frac{\partial \mathbf{f}_\mathbf{n}}{\partial \hat{\phi}^t_n}\right]$ and $\left[\frac{\partial \mathbf{f}_\mathbf{p}}{\partial \hat{\phi}^t_p}\right]$ are:

$$\begin{aligned}
\frac{\partial \mathbf{f}_\psi(i)}{\partial \hat{\psi}^t(j)} &= \mathbf{D}^2(i,j) - \mathbf{p}^t(i)\mathbf{D}^0(i,j) - \mathbf{n}^t(i)\mathbf{D}^0(i,j), && (5.8)\\
\frac{\partial \mathbf{f}_\mathbf{n}(i)}{\partial \hat{\phi}^t_n(j)} &= \mathbf{D}^2(i,j) + \mathbf{D}^1(i,:)\hat{\psi}^t\mathbf{D}^1(i,j) - 2\mathbf{D}^1(i,:)\hat{\phi}^t_n\mathbf{D}^1(i,j) + \mathcal{R}_n(i,j), && (5.9)\\
\frac{\partial \mathbf{f}_\mathbf{p}(i)}{\partial \hat{\phi}^t_p(j)} &= \mathbf{D}^2(i,j) - \mathbf{D}^1(i,:)\hat{\psi}^t\mathbf{D}^1(i,j) + 2\mathbf{D}^1(i,:)\hat{\phi}^t_p\mathbf{D}^1(i,j) - \mathcal{R}_p(i,j), && (5.10)\\
& \quad i = 1,2,\ldots,2^J L+1, \quad j = 1,2,\ldots,2^J L+3,
\end{aligned}$$

where

$\mathbf{D}^1(i,:)$ represents row i of $\mathbf{D}^1$,

$\mathbf{n}(i) = exp(\hat{\psi}(i) - \hat{\phi}_n(i)), \quad \mathbf{p}(i) = exp(\hat{\phi}_p(i) - \hat{\psi}(i)),$

and

$$\begin{aligned}
\mathcal{R}_n(i,j) &= -\left\{\frac{[\tau_n(\mathbf{p}^t(i)+1)+\tau_p(\mathbf{n}^t(i)+1)]\frac{1}{\mathbf{n}^t(i)} - \tau_p\,[\mathbf{p}^t(i)\mathbf{n}^t(i)-1]}{[\tau_n(\mathbf{p}^t(i)+1)+\tau_p(\mathbf{n}^t(i)+1)]^2} + \frac{\mathbf{n}^{t-1}(i)}{\Delta t}\frac{1}{\mathbf{n}^t(i)}\right\}\frac{L_D^2\mathbf{D}^0(i,j)}{\mu_n D_0}, \\
\mathcal{R}_p(i,j) &= \left\{\frac{[\tau_n(\mathbf{p}^t(i)+1)+\tau_p(\mathbf{n}^t(i)+1)]\frac{1}{\mathbf{p}^t(i)} - \tau_n\,[\mathbf{p}^t(i)\mathbf{n}^t(i)-1]}{[\tau_n(\mathbf{p}^t(i)+1)+\tau_n(\mathbf{n}^t(i)+1)]^2} + \frac{\mathbf{p}^{t-1}(i)}{\Delta t}\frac{1}{\mathbf{p}^t(i)}\right\}\frac{L_D^2\mathbf{D}^0(i,j)}{\mu_p D_0}.
\end{aligned}$$

The boundary conditions of ψ are merged to $\mathbf{f}_\psi$ as last two equations. In this sense, we write:

$$\mathbf{f}_\psi(K+1) = \hat{\psi}^t(1) - \ln N_D = 0,$$
$$\mathbf{f}_\psi(K+2) = \hat{\psi}(N) + \ln N_A - v_b(t) = 0,$$

where K is the dimension of old $\mathbf{f}_\psi$, and N is the index corresponding to basis function $\eta_1(L-x)$. Similarly, we write

$$\begin{aligned}
\mathbf{f}_n(K+1) &= \hat{\phi}_n^t(0) = 0, \\
\mathbf{f}_n(K+2) &= \hat{\phi}_n^t(N) - v_b = 0, \\
\mathbf{f}_p(K+1) &= \hat{\phi}_p^t(0) = 0, \\
\mathbf{f}_p(K+2) &= \hat{\phi}_p^t(N) - v_b = 0.
\end{aligned}$$

The corresponding Jacobian for these equations are:

$$\frac{\partial \mathbf{f}_\psi(K+1)}{\partial \hat{\psi}(i)} = \begin{cases} 1 & i=1, \\ 0 & \text{otherwise;} \end{cases} \qquad \frac{\partial \mathbf{f}_\psi(K+2)}{\partial \hat{\psi}(i)} = \begin{cases} 1 & i=N, \\ 0 & \text{otherwise.} \end{cases}$$

It is same for $\frac{\partial \mathbf{f}_n(K+1)}{\partial \hat{\phi}_n(i)}$, $\frac{\partial \mathbf{f}_n(K+2)}{\partial \hat{\phi}_n(i)}$, $\frac{\partial \mathbf{f}_p(K+1)}{\partial \hat{\phi}_p(i)}$ and $\frac{\partial \mathbf{f}_p(K+1)}{\partial \hat{\phi}_p(i)}$. After this merging process, these Jacobian matrices have equal number of rows and columns, which can be inverted.

Now, let's give a brief review on the modified Gummel's method to solve algebraic equations (5.5)(5.2)(5.3) at each time step.

(1) Guess a trial solution $\psi^{\mathbf{0}}, \phi_{\mathbf{n}}^{\mathbf{0}}, \phi_{\mathbf{p}}^{\mathbf{0}}$;

(2) Update ψ by one step Newton's method:

$$\psi^{k+1} \leftarrow \psi^k - \left[\frac{\partial \mathbf{f}_\psi}{\partial \psi}\right]^{-1} \mathbf{f}_\psi(\psi^{\mathbf{k}}, \phi_{\mathbf{n}}^{\mathbf{k}}, \phi_{\mathbf{p}}^{\mathbf{k}});$$

(3) Update ψ by one step Newton's method:

$$\phi_n^{k+1} \leftarrow \phi_n^k - \left[\frac{\partial \mathbf{f}_n}{\partial \phi_n}\right]^{-1} \mathbf{f}_n(\psi^{\mathbf{k+1}}, \phi_{\mathbf{n}}^{\mathbf{k}}, \phi_{\mathbf{p}}^{\mathbf{k}});$$

(4) Update ψ by one step Newton's method:

$$\phi_p^{k+1} \leftarrow \phi_p^k - \left[\frac{\partial \mathbf{f}_p}{\partial \phi_p}\right]^{-1} \mathbf{f}_p(\psi^{\mathbf{k+1}}, \phi_{\mathbf{n}}^{\mathbf{k+1}}, \phi_{\mathbf{p}}^{\mathbf{k}});$$

(5) Evaluate error ξ:

$$\xi = max\left(|max(\psi^{k+1} - \psi^k)|, |max(\phi_n^{k+1} - \phi_n^k)|, |max(\phi_p^{k+1} - \phi_p^k)|\right),$$

If $\xi < \epsilon$, where ϵ is a predefined tolerance, we stop the iteration process, otherwise, repeat step 2 to step 4 until this condition is satisfied.

Note that the superscript here is the iteration index.

We will show the steady state and transient solution of the selected diode in following subsections.

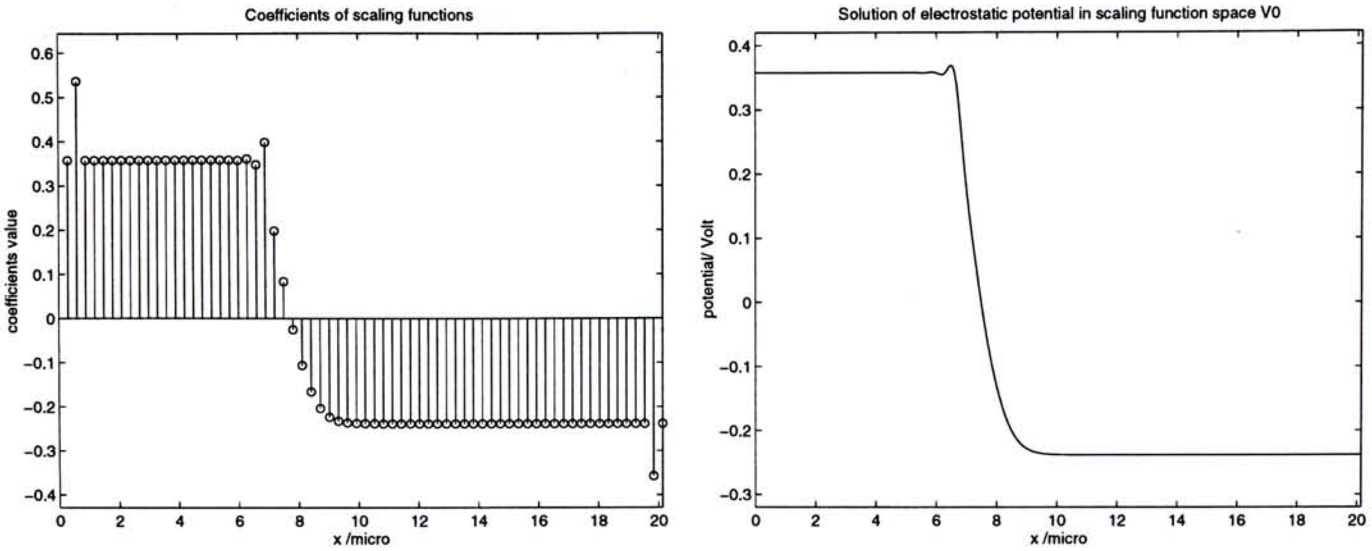
5.1.1 Steady state solution

Finding steady state solution is the first step for transient solution. Besides, we generate the reduced transform $\mathcal{D}^0$, $\mathcal{D}^1$ and $\mathcal{D}^2$ in this step. In other words, we generate the grid here.

For steady state, time varying terms $\frac{\partial n}{\partial t}$, $\frac{\partial p}{\partial t}$ in (5.2)(5.3) vanish. Consequently, $\frac{L_D^2}{D_0}\frac{\mathbf{n}^t-\mathbf{n}^{t-1}}{\Delta t}$, and $\frac{L_D^2}{D_0}\frac{\mathbf{p}^t-\mathbf{p}^{t-1}}{\Delta t}$ in (5.6) and (5.7) are omitted.

We will only discuss the equilibrium solution, i.e, unbiased diode here. Because the grid generation scheme is based on equilibrium solution. Arbitrary dc biased solution can be easily and quickly obtained when the reduced transform matrices are generated.

Let $J0 = 6$, then there are $2^6 + 1 = 65$ collocation points in the scaling function level. Remember the collocation point in this level is denoted as $x^{(-1)}$, and all of these



(a) (b)

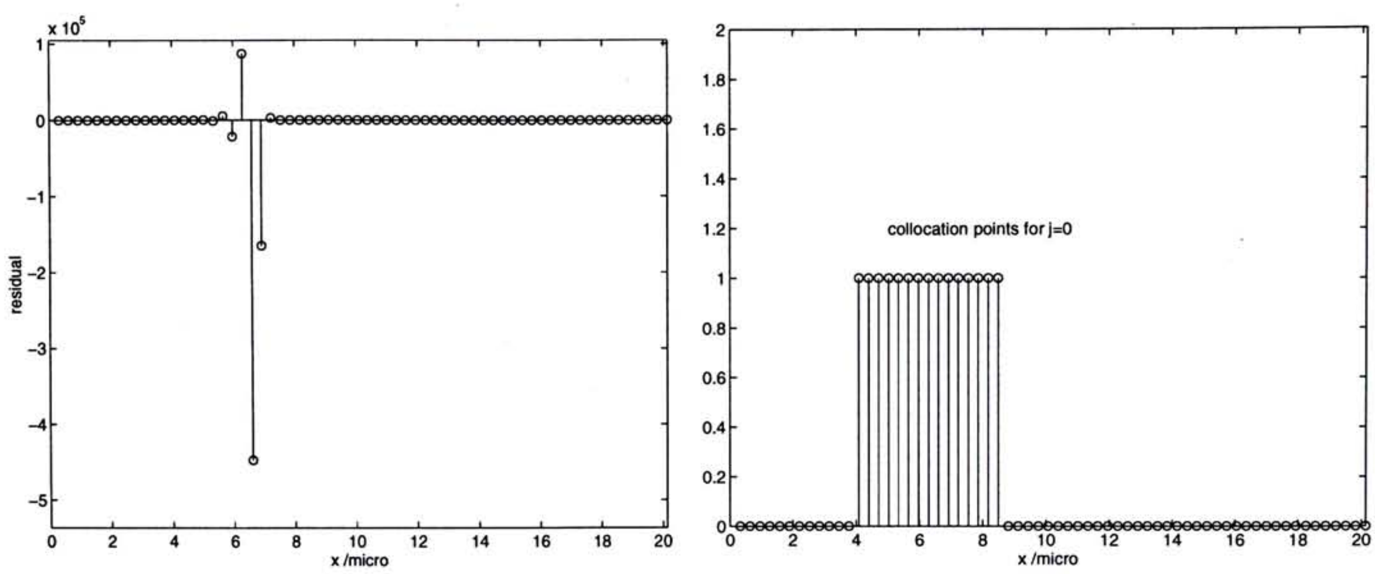
Figure 5.2: Solution of $\psi(x)$ in scaling function space: (a) scaling function coefficients; (b) waveform.

points should not be deleted. The solved scaling function coefficients $\hat{\psi}$ of $\psi(x)$ are shown in Figure 5.2(a). The solution $\psi(x)$ at this resolution level can be obtained by scaling function interpolation shown in equation (4.34). Its waveform is shown in Figure 5.2(b). ϕ_n and ϕ_p are zero in equilibrium.

From the figure, we see that this level gives the "average" information of $\psi(x)$. Later, we will see that the local "detail" of $\psi(x)$ will be given in wavelet space.

The next step is to evaluate (5.1)(5.2)(5.3) at points $x^{(0)}$ in order to identify the necessary range of collocation points for space $W0$, outside where the solution in previous level has achieved enough accuracy. Here, we evaluate only one equation (5.1) which is critical. The residual of (5.1) at $x^{(0)}$ is shown in Figure 5.3(a). According to the adaptive method described in Chapter 4, we determine the range of collocation points as shown in Figure 5.3(b). Note that we leave some safety margin when determine range of collocation points. From the graph we see that 49 collocation points were deleted out of 64, so we use only 15 collocation points for wavelet space $W0$.

The solution of wavelet coefficients of ψ in $W0$ are shown in Figure 5.4(a), and the corresponding spatial domain waveform after interpolation is shown in Figure 5.4(b). For solution in $V0 \oplus W0$, we simply add solutions in these two spaces. The result is



(a) (b)

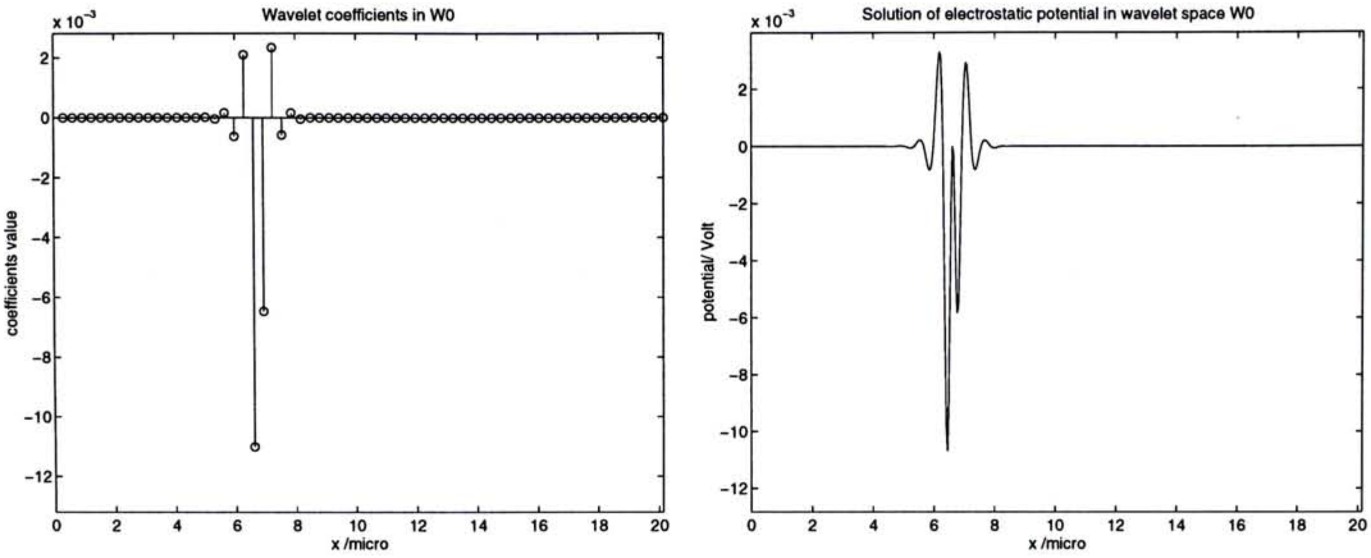
Figure 5.3: (a) Residual of (5.1) in $x^{(0)}$ for solution in $V0$; (b) Range of collocation points for space $W0$.

shown in Figure 5.9(a).

Again, we use the solution calculated in space $V0 \oplus W0$ to evaluate (5.1) at points $x^{(1)}$, and determine the range of necessary collocation points in wavelet space $W1$. The residual of (5.1) is shown in Figure 5.3(a). The determined the range of collocation points is shown in Figure 5.3(b). Note that there are safety margin in the range of collocation points. And note that 113 collocation points were deleted out of 128, so we use only 15 collocation points for wavelet space $W1$.

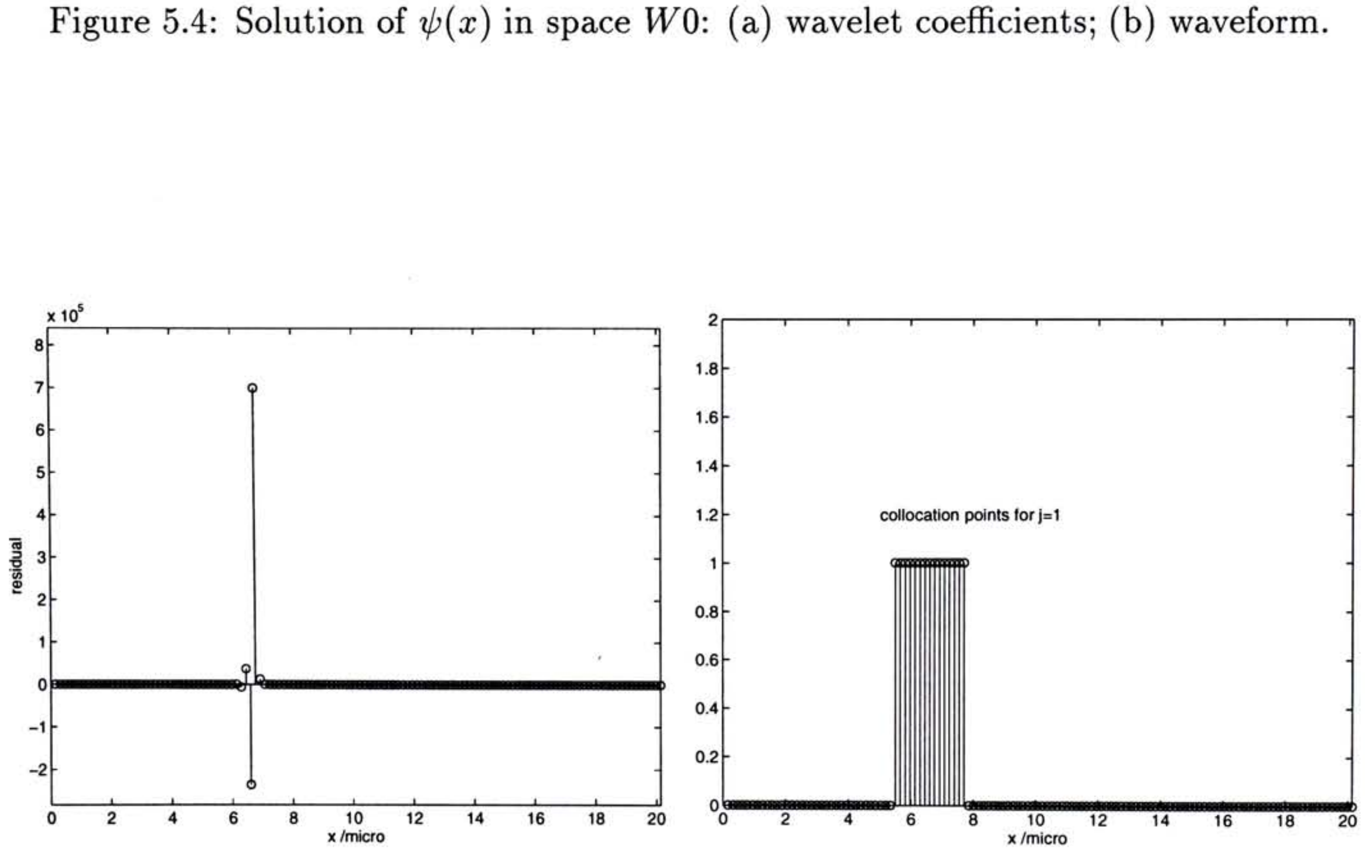
The solution of $\psi(x)$ in wavelet space $W1$ are shown in Figure 5.6, where Figure 5.6(a) is the wavelet coefficients, and (b) is the waveform after wavelet interpolation(equation 4.34). The solution of $\psi(x)$ in space $V0 \oplus W0 \oplus W1$ is shown in Figure 5.9(a).

It's very nature to ask when we stop increasing the resolution level? As described in Chapter 4, if the maximum magnitude of the solution in a wavelet space is less than a predefined tolerance, we stop. Otherwise we increase the resolution level until this criteria is satisfied or predefined maximum resolution level is reached. In this example, we let tolerance be $0.01V$, and the maximum resolution is $W2$.



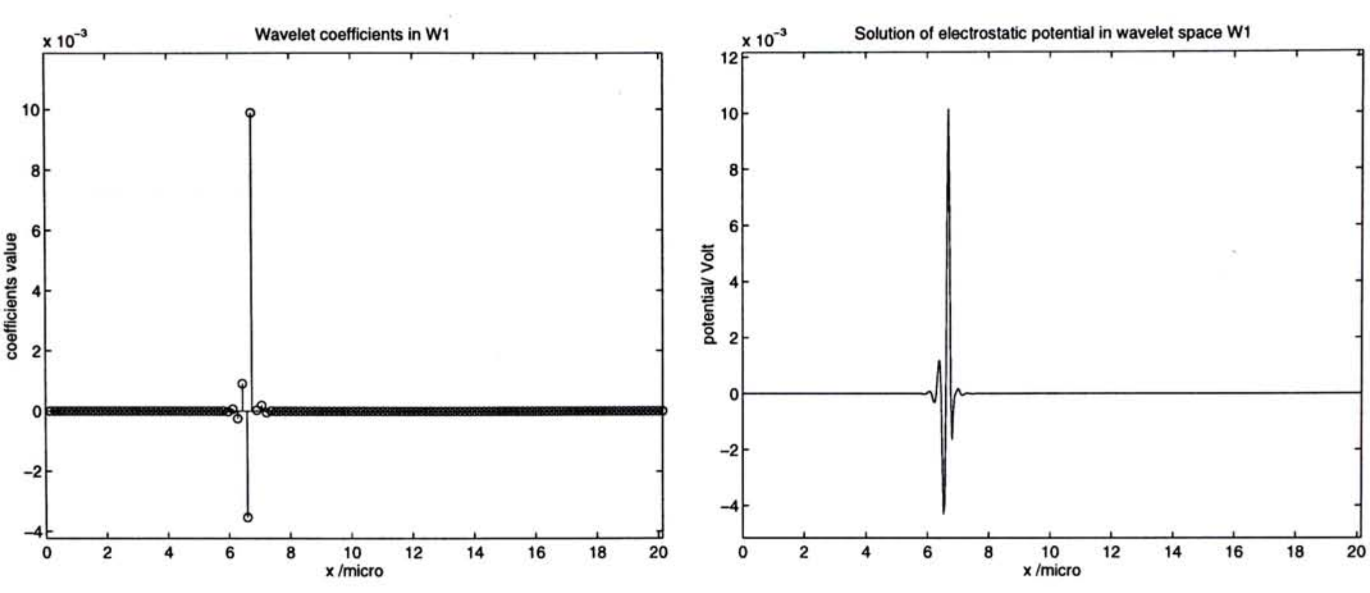
(a) (b)

Figure 5.4: Solution of $\psi(x)$ in space $W0$: (a) wavelet coefficients; (b) waveform.



(a) (b)

Figure 5.5: (a) Residual of (5.1) in $x^{(1)}$ for solution in $W0$; (b) Range of collocation points for space $W1$.



(a) (b)

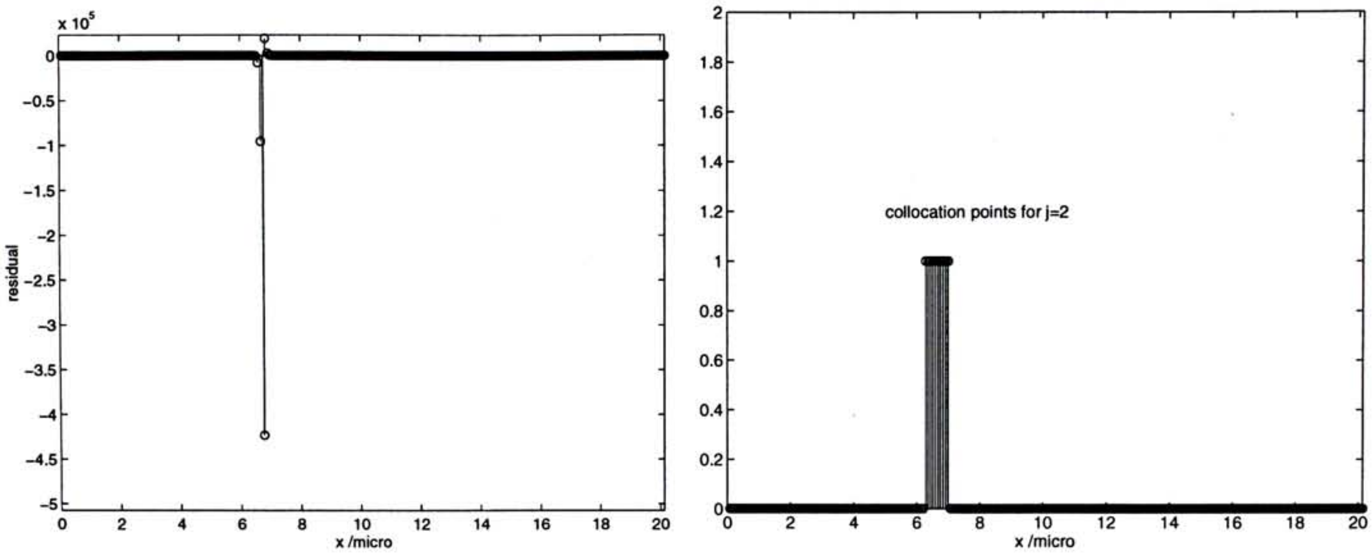
Figure 5.6: Solution of $\psi(x)$ in space $W1$: (a) wavelet coefficients; (b) waveform.

The solution in $W1$ do not satisfy this condition, we increase the resolution level. The residual and range of collocation points are shown in Figure 5.7(a),(b). Here 246 collocation points out of 256 were deleted, so we have only 10 collocation points for $W2$. The solution of wavelet coefficients and waveform are shown in Figure 5.8. From the graph, we see that the maximum amplitude of solution in this space is small than tolerance $0.01V$, so we stop here. The solution in $V0 \oplus W0 \oplus W1 \oplus W2$ is shown in Figure 5.10.

The solutions in all resolution levels are put together in Figure 5.11 for comparison. From the graph we observe: (1) outside junction range, solutions of all levels are the same; (2) near the junction where sharp changes occur, low resolution solution has ripples, and the ripples are reduced quickly when the resolution increases; (3) The solutions of the highest two resolution levels are almost the same, i.e, the solution is converged.

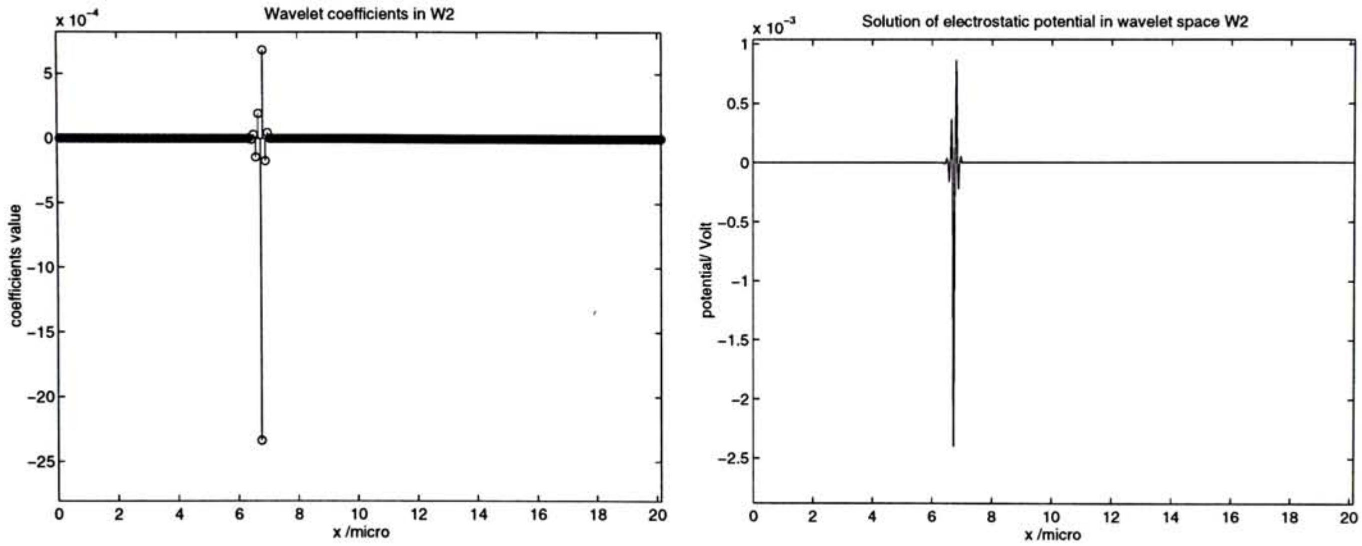
From Figure 5.11(a), we also see that the device variable ψ varies very fast in a small region, outside where it varies smoothly. This result verifies our assumption.

The number of total collocation points up to space $W2$ is $65+15+15+10 = 105$. Without adaptive algorithm, we need $(2^2+2^1+2^0)*L+L+1 = 8*L+1 = 8*64+1 = 513$



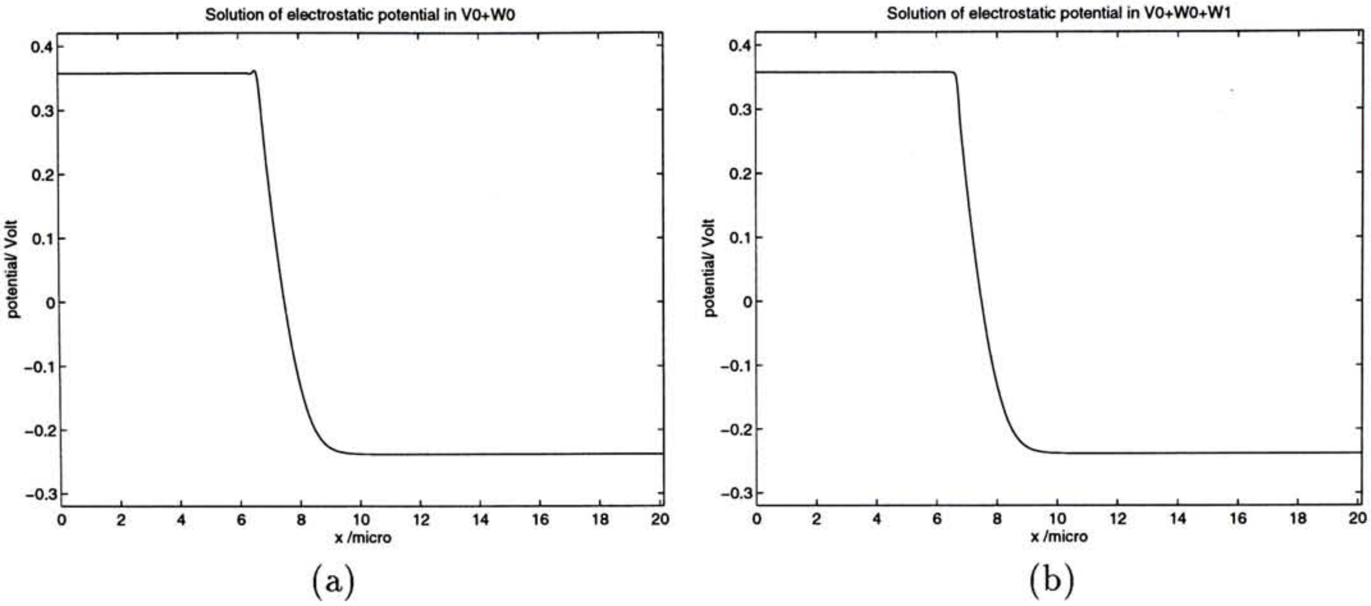
(a) (b)

Figure 5.7: (a) Residual of (5.1) in $x^{(2)}$ for solution in $W1$; (b) Range of collocation points for space $W2$.



(a) (b)

Figure 5.8: Solution of $\psi(x)$ in space $W2$: (a) wavelet coefficients; (b) waveform.



(a) (b)

Figure 5.9: Solution of $\psi(x)$ in spaces (a) $V0 \oplus W0$ (b) $V0 \oplus W0 \oplus W1$.

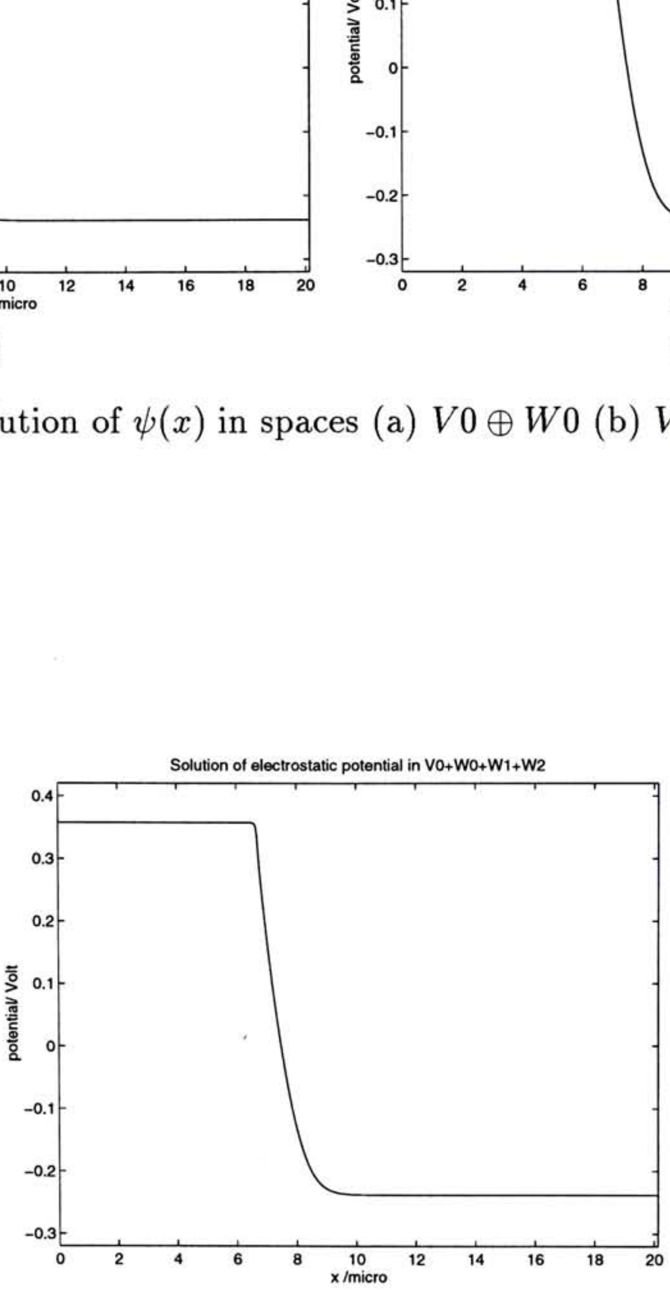
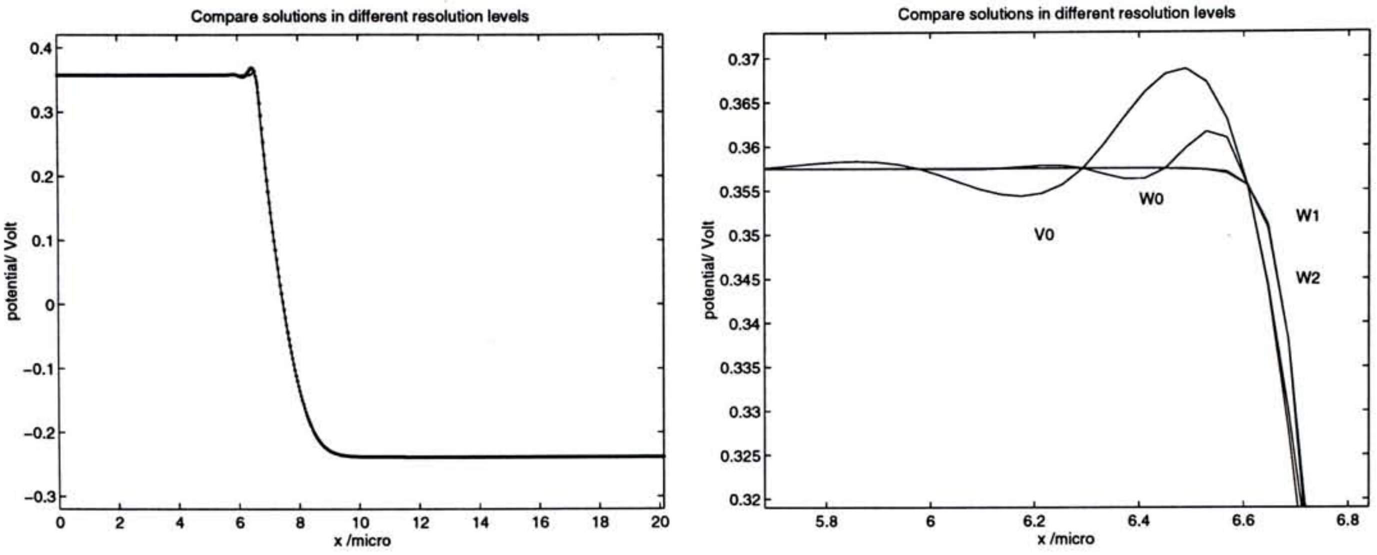


Figure 5.10: Solution of $\psi(x)$ in space $V0 \oplus W0 \oplus W1 \oplus W2$.



(a)

(b)

Figure 5.11: Solution of $\psi(x)$ in all resolution levels, (a) normal scale; (b) zoom in.

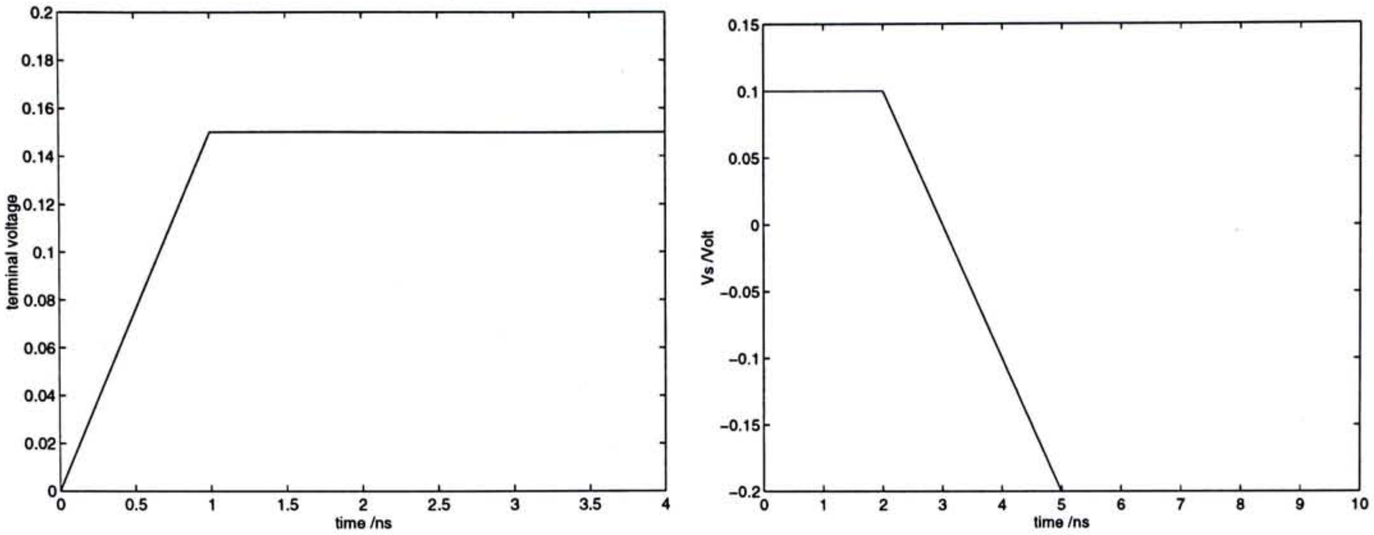
collocation points. We see that a huge amount of collocation points (about 80%) are deleted without any reduction of accuracy.

Hence reduced transform matrices $\mathcal{D}^0, \mathcal{D}^1, \mathcal{D}^2$ are with the dimension 105×107, while full transform matrices are with dimension 513×515. In fact, the redundancy of these transform matrices are removed by the procedure describe previously. Such collocation points/wavelet basis reduction scheme will lead to very efficient computation in seeking the transient solution. We show this in next subsection. Besides, the reduced transform matrices are used for dc solution of arbitrary biased diode.

5.1.2 Transient solution

Once we get the reduced transform matrices $\mathcal{D}^0, \mathcal{D}^1, \mathcal{D}^2$, we can use it in solving diode transient equations (5.5)(5.6)(5.7) for all time steps. Having a systematic generation scheme of these reduced transform matrices is a major advantage of our method.

First, we try to investigate the forward transient response that the terminal voltage of diode increases from zero to a certain positive DC level. The terminal diode voltage is shown in Figure 5.12(a). Apply the cubic spline wavelet collocation method, we obtain solutions $\psi(x), \phi_n(x), \phi_p(x)$ in each time step, which are shown in Figure 5.13(a).



(a) (b)

Figure 5.12: Diode terminal voltage waveforms: (a) from zero to positive DC; (b) from positive DC to negative DC.

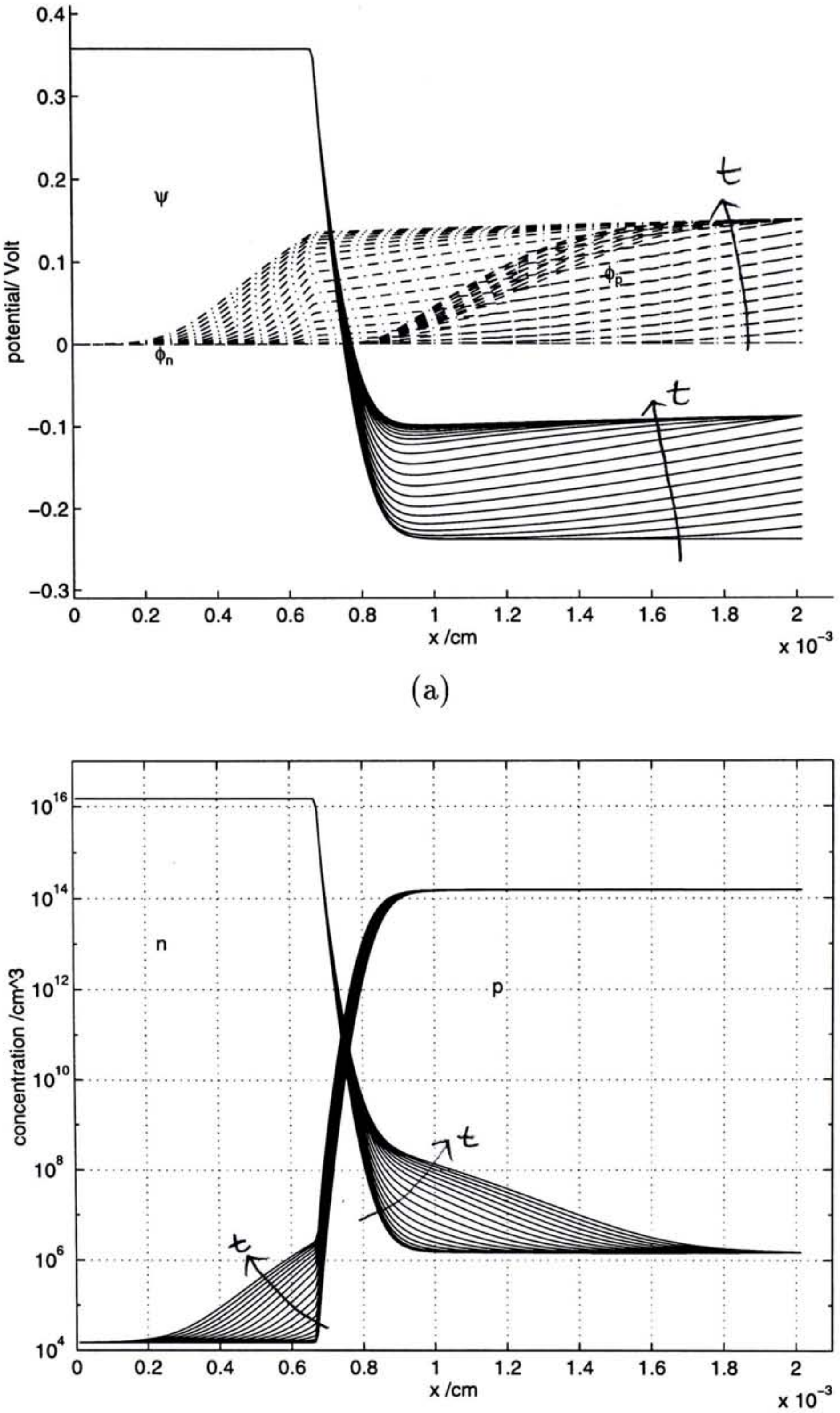
We calculated the solutions of carrier concentrations $n(x)$ and $p(x)$ by

$$n = n_i \exp(\frac{\psi - \phi_n}{Vt}), \qquad p = n_i \exp(\frac{\phi_p - \psi}{Vt}),$$

and solutions in all time steps are shown in Figure 5.13(b).

From Figure 5.13(a), we observe that (1) The diode built-in voltage is decreasing when terminal voltage is increasing; (2) The relative potentials between ψ and ϕ_n, and between ψ and ϕ_p in two boundaries do not change in any time step, and this is inferred from the assumption that majority concentrations in two boundary always equal to doping concentrations; (3) The quasi-Fermi levels ϕ_n, ϕ_p of electrons and holes respectively are split in the region around the junction, which is called depletion region in the literature. The splitting of ϕ_n and ϕ_p indicates that this region is not in equilibrium, i.e, the product of carrier np do not equal to n_i^2. Hole-electron recombination occurs in this region. The bulk region far away from the junction is in the equilibrium state because two quasi-Fermi levels are equal.

From Figure 5.13(b), we observe that (1) The majority concentrations almost keep constant; (2) extra minorities are being injected to opposite side when the terminal voltage is increasing. These injected extra minorities are stored in the opposite region,



(a)

(b)

Figure 5.13: Diode forward transient response under terminal voltage of Figure 5.12(a) (a) ψ, ϕ_n and ϕ_p; (b) carrier concentrations n and p.

and will lead to the charge storage effect when terminal voltage decreases. (3) When the diode is biased in dc steady state, the extra minorities are diffused into the bulk region. In addition, the concentrations of extra minorities in the edges of so called depletion region are more than 10 times of that in equilibrium state.

The distributions of ψ, ϕ_n and ϕ_p of reverse transient response are shown in Figure 5.14(a). The excitation is shown in Figure 5.12(b). The corresponding electron concentration n and hole concentration p at each time step are shown in Figure 5.14(b).

From Figure 5.14(a), we observe that (1)When the final state (negative DC biased) is reached, the quasi-Fermi levels ϕ_n and ϕ_p are also split in the region around the junction. Different from forward transient, the ϕ_p is above ϕ_n now. This indicates that the recombination occurred here are negative, i.e., hole-electron pairs are generated in this region. (2) Built-in voltage is increasing when terminal voltage is decreasing.

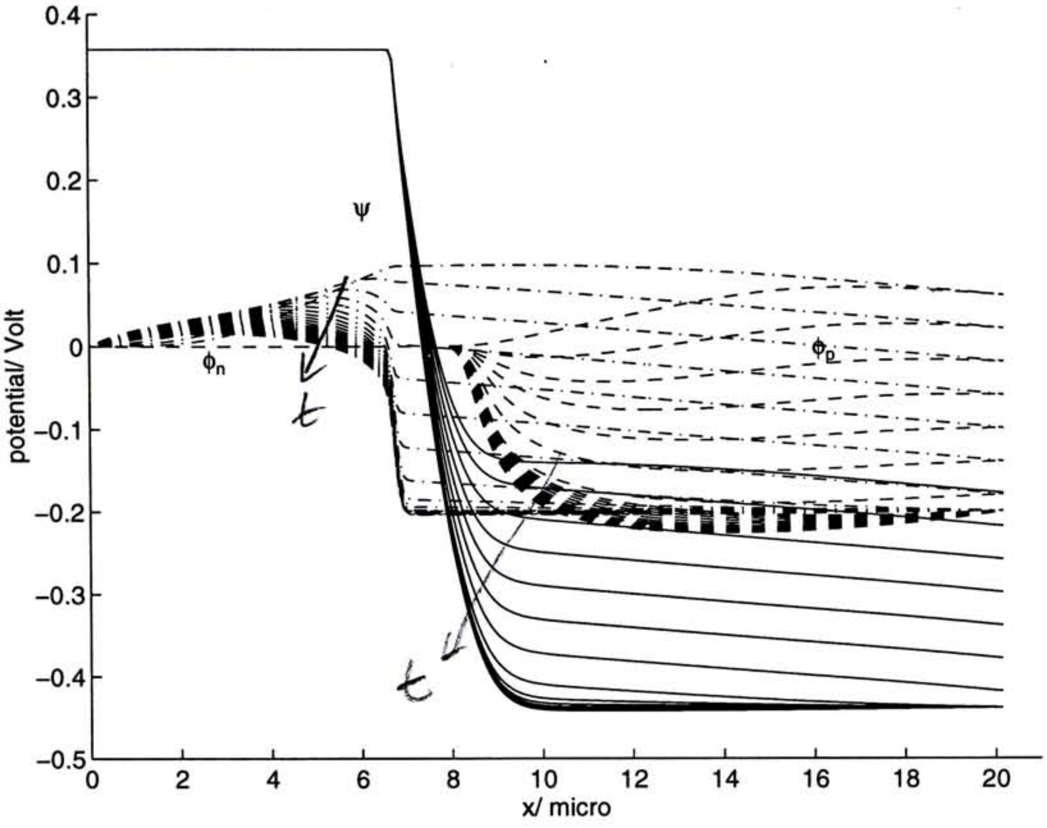
From Figure 5.14(b) we observe that (1) The extra minority in N and P regions are not removed immediately after application of reverse terminal voltage, but been removed gradually. This phenomenon is called charge storage effect. (2) In the final negative dc biased state, the minorities in the edges of so called depletion region are below its equilibrium level.

5.1.3 Convergence

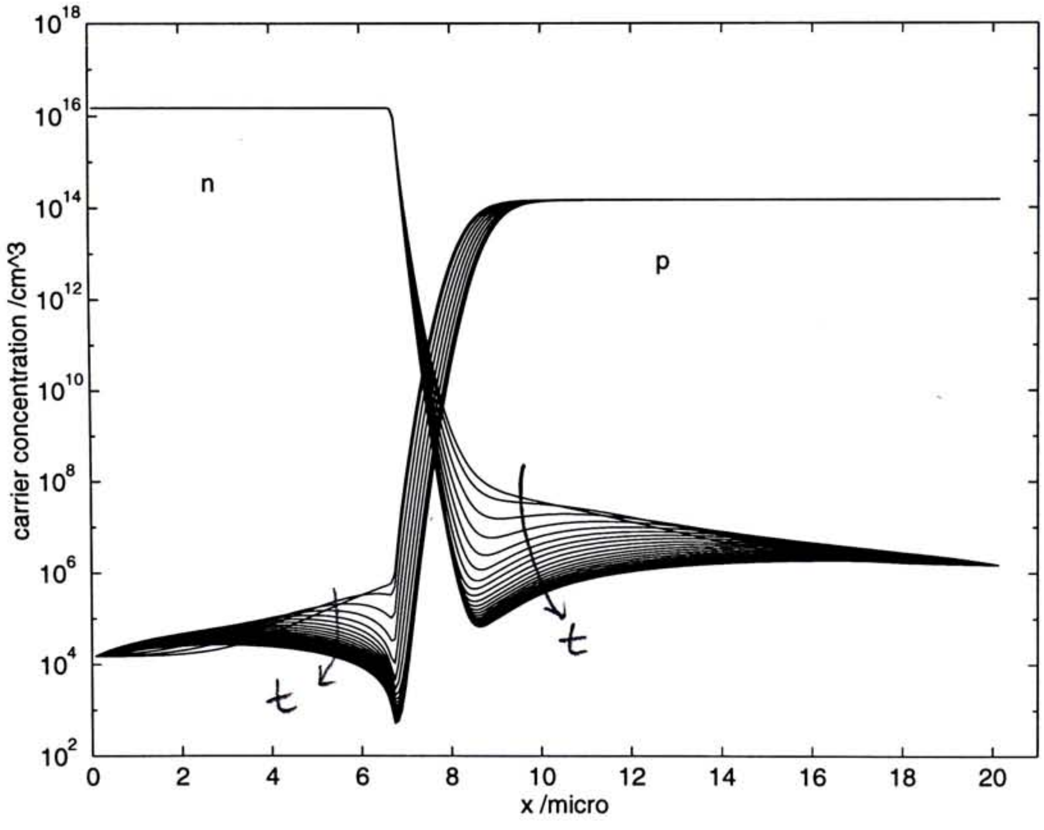
In this subsection, we discuss the convergence of wavelet collocation method in device simulation. The convergence we discuss here is of the Gummel's relaxation process, which is the only one iteration process involved in our method.

The convergence of any iteration process is much dependent on the initial guess. Gummel's method is not the exception.

In solving steady state solution, the program converges only after 2 iterations with proper initial guess. Experiments show that when one more iteration is completed, the error ξ, where $\xi = max\left(|max(\psi^{k+1} - \psi^k)|, |max(\phi_n^{k+1} - \phi_n^k)|, |max(\phi_p^{k+1} - \phi_p^k)|\right)$, is reduced by 10 times, i.e, we gain one decimal accuracy each iterations.



(a)



(b)

Figure 5.14: Diode reverse transient response under terminal voltage of Figure 5.12(b) (a) ψ, ϕ_n and ϕ_p; (b) carrier concentrations n and p.

In solving transient solution, the program converges within 20 iterations for all time steps.

5.2 Bipolar Transistor

The device equations of bipolar transistor are the same with that of p-n junction diode. The only difference is that the bipolar transistor is a three terminal device while p-n junction diode is simply a two terminal device. Therefore, they have different boundary conditions. In the beginning of this section, we will give the boundary model of bipolar transistor. Then, the steady state and transient solutions of a n-p-n transistor will be shown.

The parameters of a transistor which will be discussed in this section are list in Table 5.2.

Table 5.2: Physical parameters of a bipolar transistor.

Material	Silicon (relative permittivity = 11.8)
Temperature	$300^0 K$
Intrinsic Impurity Concentration(n_i)	$1.5 \times 10^{10}/cm^3$
Doping	Emitter: $N_E = 10^5(or 1.5 \times 10^{15} cm^{-3})$ Base, $N_B = 10^4(or 1.5 \times 10^{14} cm^{-3})$ Collector, $N_C = 10^3(or 1.5 \times 10^{13} cm^{-3})$
Intrinsic Debye Length($\sqrt{\frac{V_t \epsilon}{q n_i}}$)	$0.00335618cm$ or $33.5618\mu m$
Length	Emitter length= 0.2 (or $6.7\mu m$) Base length=0.05 (or $1.68\mu m$) Collector length=0.5 (or $17\mu m$)
Carrier mobilities	electron, $1500cm^2/V - sec$ hole, $450cm^2/V - sec$
Minority life time	electron $\tau_n = 10^{-4}sec$ hole $\tau_p = 10^{-5}sec$

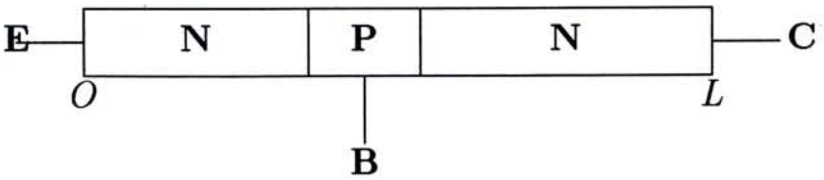


Figure 5.15: Diagram of bipolar transistor

5.2.1 Boundary Model

The boundary conditions at the emitter and collector terminals are very simple. Let's choose the electric potential at base contact as reference, and assume the carrier concentration at these boundary equals to the ionized doping concentration, we obtain:

$$\begin{aligned} \psi(0,t) &= \ln N_E - v_{be}; \quad \phi_n(0,t) = \phi_p(0,t) = -v_{be}; \\ \psi(L,t) &= \ln N_C - v_{bc}(t); \quad \phi_n(L,t) = \phi_p(0,t) = -v_{bc}(t); \end{aligned}$$

where v_{be} is the applied base-emitter voltage, v_{bc} is base-collector voltage, and N_E, N_B, N_C are emitter doping, base doping and collector doping respectively.

Let's denote the collocation point index at base contact as B. In this point, we do NOT evaluate the Poisson's equation, continuity equation for electrons and holes. The following equation

$$\psi(B,t) = \ln N_B, \tag{5.11}$$

is used instead of Poisson's equation. This equation is the boundary condition for ψ. On the other hand, the continuity equation at this point is no longer satisfied because of the charge injected from base terminal. Hence, we should use the following two equations

$$j_n(B-1) = j_n(B+1) - j_{Bn}, \tag{5.12}$$

$$j_p(B-1) = j_p(B+1) - j_{Bp}, \tag{5.13}$$

to substitute electron and hole transport equations respectively, where j_{Bn}, j_{Bp} are the normalized hole and electron current density, respectively, flowing into the base region of transistor. The current density normalization factor is $\frac{qn_iV_t}{L_D}$. For NPN transistors,

$j_{Bn} = 0$ and j_{Bp} equals to the base current. For PNP transistors, $j_{Bp} = 0$ and j_{Bn} equals to the base current. The above two equations also describe the charge flow continuity phenomenon.

In wavelet collocation method, the above boundary model can be represented by:

$$\hat{\psi}^t(1) = \ln N_E - v_{be}(t), \tag{5.14}$$

$$\hat{\phi}^t_n(1) = -v_{be}, \tag{5.15}$$

$$\hat{\phi}^t_p(1) = -v_{be}, \tag{5.16}$$

for the emitter terminal, and

$$\hat{\psi}(N) = \ln N_C - v_{bc}(t), \tag{5.17}$$

$$\hat{\phi}^t_n(N) = -v_{bc}(t), \tag{5.18}$$

$$\hat{\phi}^t_p(N) = -v_{bc}(t), \tag{5.19}$$

(N is the index corresponding to basis function $\eta_1(L - x)$)

for the collector terminal, and

$$\mathbf{D}^0\hat{\psi}^t = \ln N_B, \tag{5.20}$$

$$\mu_n n(B-1)\mathbf{D}^1(B-1,:)\hat{\phi_\mathbf{n}} = \mu_n n(B+1)\mathbf{D}^1(B+1,:)\hat{\phi_\mathbf{n}} + j_{Bn}, \tag{5.21}$$

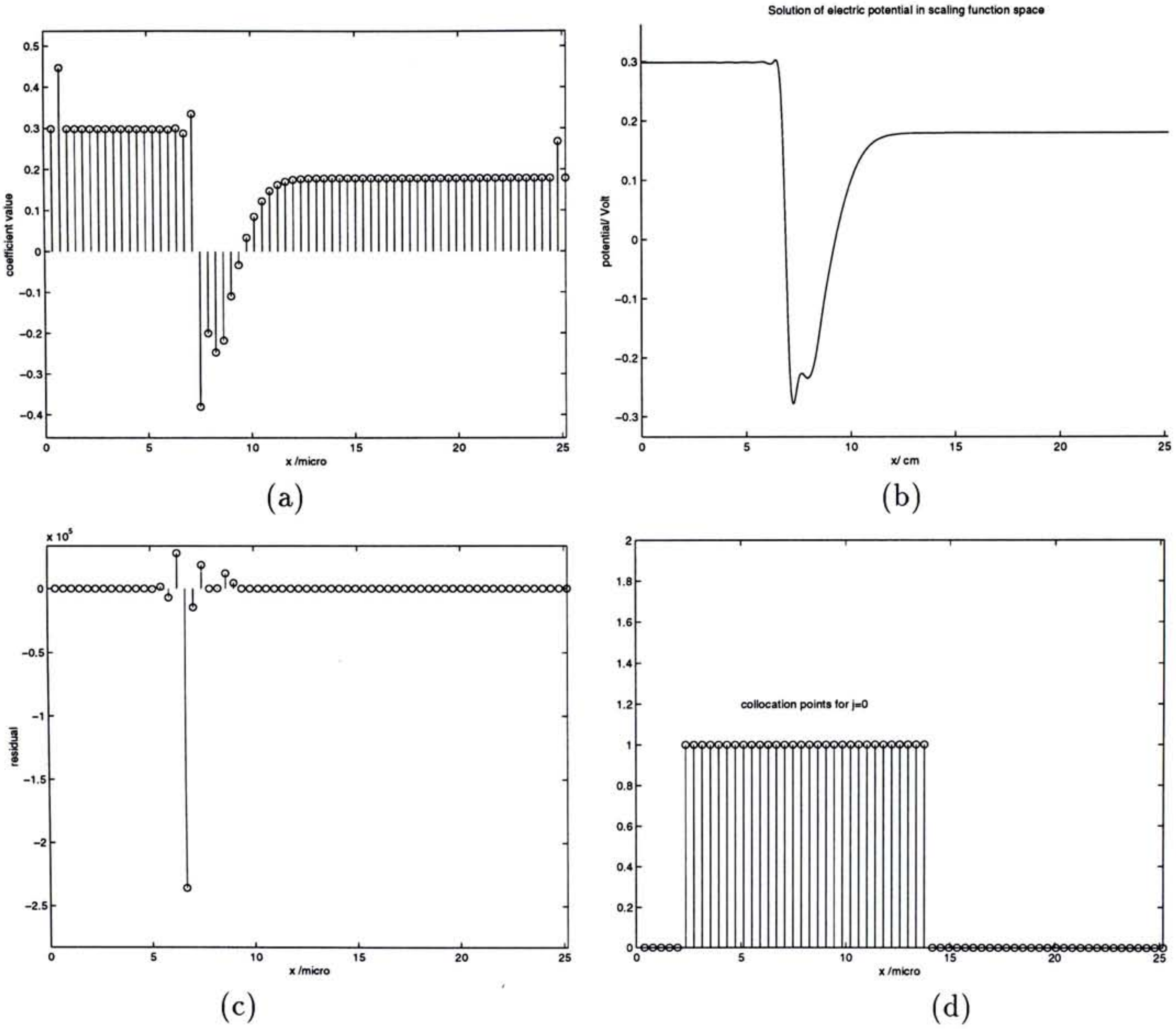
$$\mu_p p(B-1)\mathbf{D}^1(B-1,:)\hat{\phi_\mathbf{p}} = \mu_p p(B+1)\mathbf{D}^1(B+1,:)\hat{\phi_\mathbf{p}} + j_{Bp}, \tag{5.22}$$

for the base terminal, where $n(B-1) = \exp\{\mathbf{D}^0(B-1,:)(\hat{\psi} - \hat{\phi}_n)\}$ is the electron concentration at point (B-1), and so on.

5.2.2 DC Solution

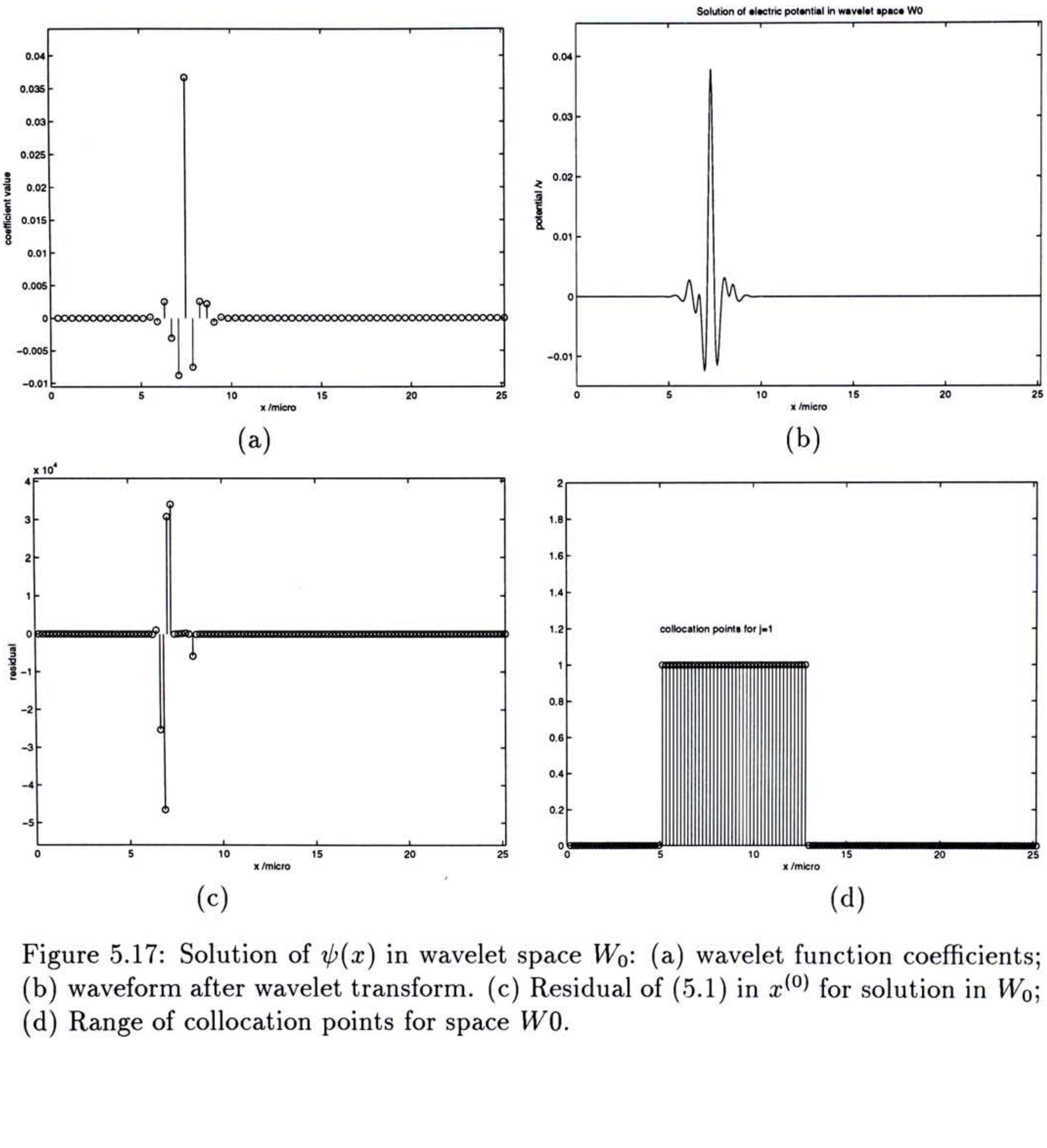
In this subsection, the steady state solution of unbiased transistor is presented. We also show the grid generation process here. The transient response will be presented in next subsection.

First, the solution in scaling function space is shown in Figure 5.16(a)(b), where Figure 5.16(a) shows the scaling function coefficients, and Figure 5.16(b) shows the



(a) (b) (c) (d)

Figure 5.16: Solution of $\psi(x)$ in scaling function space, (a) scaling function coefficients; (b) waveform of ψ after wavelet transform. (c) Residual of (5.1) in $x^{(0)}$ for solution in V_0; (d) Range of collocation points for space $W0$.



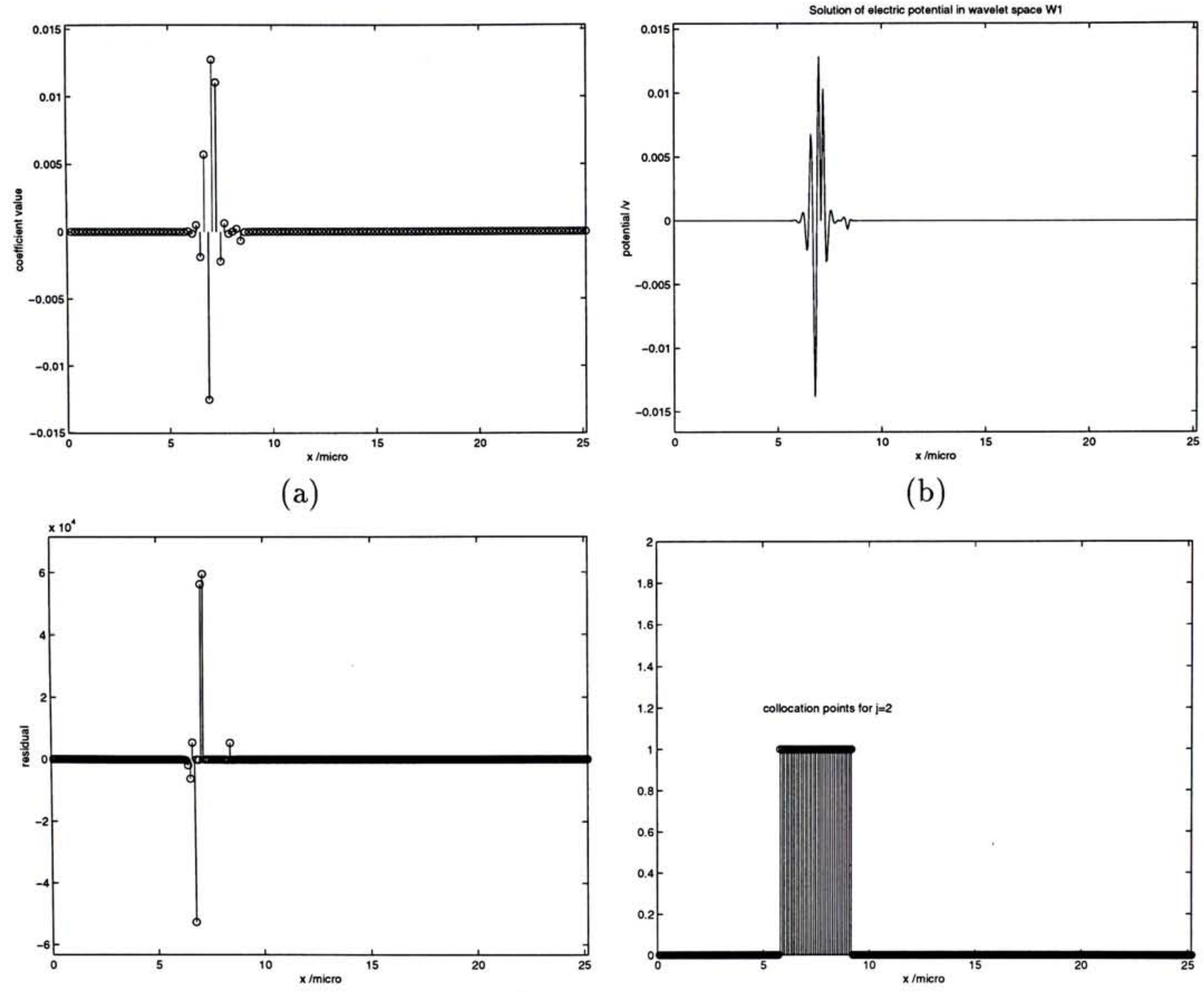
(a)

(b)

(c)

(d)

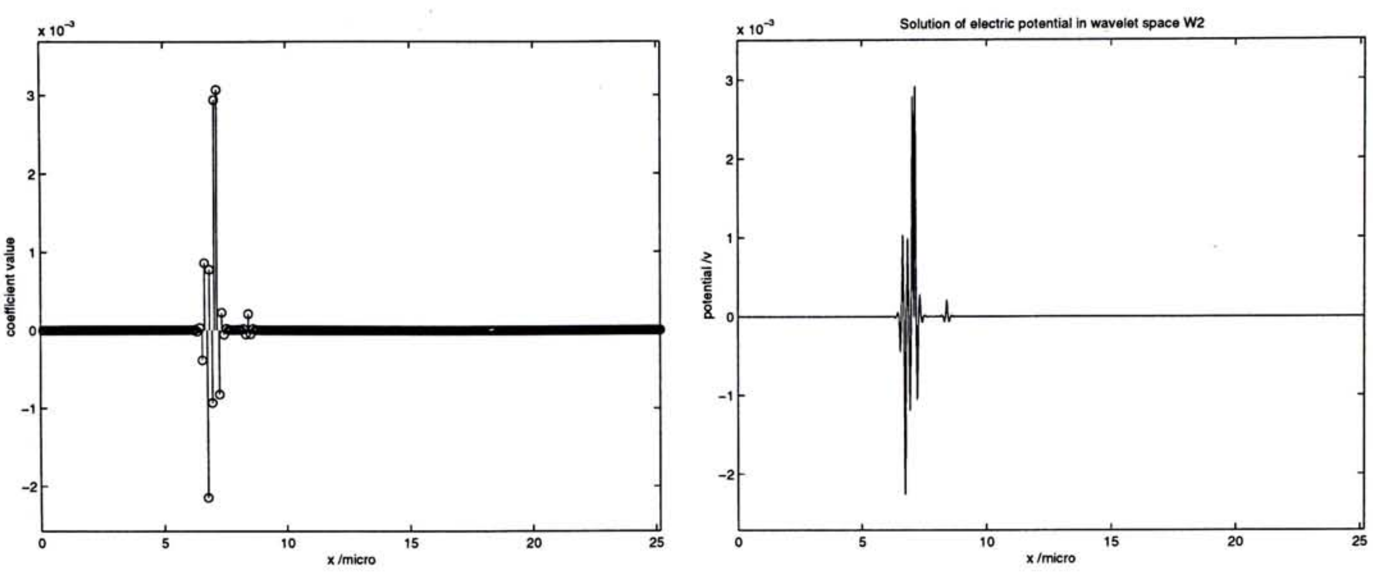
Figure 5.17: Solution of $\psi(x)$ in wavelet space W_0: (a) wavelet function coefficients; (b) waveform after wavelet transform. (c) Residual of (5.1) in $x^{(0)}$ for solution in W_0; (d) Range of collocation points for space $W0$.



(a) (b)

(c) (d)

Figure 5.18: Solution of $\psi(x)$ in wavelet space W_1: (a) wavelet function coefficients; (b) waveform after wavelet transform. (c) Residual of (5.1) in $x^{(1)}$ for solution in W_1; (d) Range of collocation points for space $W0$.



(a) (b)

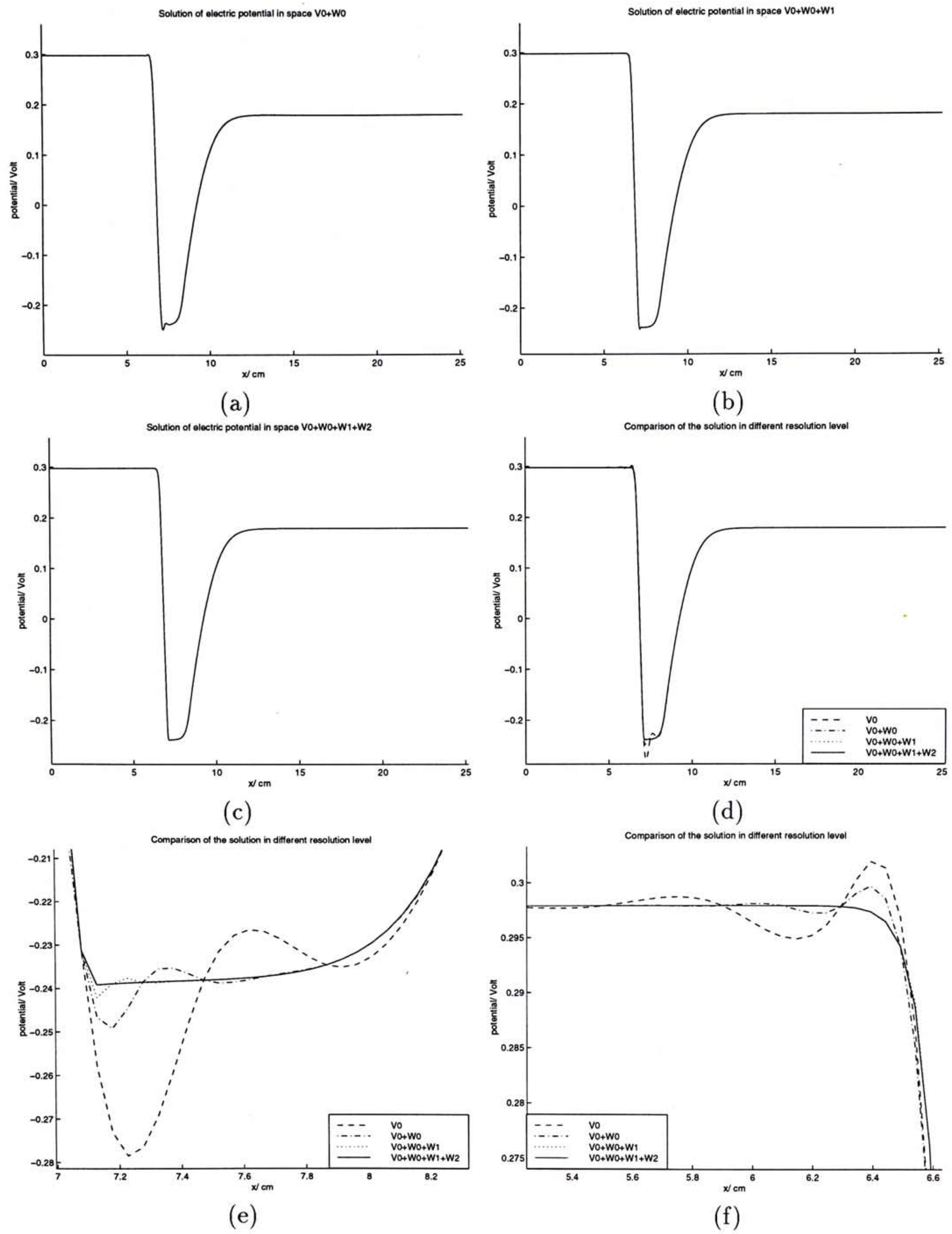
Figure 5.19: Solution of $\psi(x)$ in wavelet space W_1: (a) wavelet function coefficients; (b) waveform after wavelet transform.

waveform of ψ. The ϕ_n and ϕ_p are zero for this equilibrium case (unbiased transistor). The residual of Poisson's equation at collocation points x^0 for this solution is shown in Figure 5.16(c). Based on this residual, we determine the collocation range for wavelet space W_0 as shown in Figure 5.16(d).

We repeat such procedure for higher resolution levels, and obtain simulation results as shown in Figure 5.17, Figure 5.18 and Figure 5.19 for space W_0, W_1 and W_2 respectively. The amplitude of solution in wavelet space W_2 are small enough (as shown in Figure 5.19(b)). This means that the solution converged, so we stop this process.

We list the full scale, reduced and remaining collocation point number at different resolution levels in Table 5.3. From this table, we observe that there are 343 collocation points are deleted out of 513. This is a huge amount, and will lead very efficient computation in subsequent transient simulation.

The solutions in space $V0 \oplus W0$, $V0 \oplus W0 \oplus W1$ and $V0 \oplus W0 \oplus W1 \oplus W2$ are shown in Figure 5.20(a),(b) and (c) respectively. And they are put together in Figure 5.20(d) for comparison. To see more clearly, we zoom into two different regions



(a) (b) (c) (d) (e) (f)

Figure 5.20: Solution of $\psi(x)$ in spaces (a) $V0 \oplus W0$, (b) $V0 \oplus W0 \oplus W1$, (c) $V0 \oplus W0 \oplus W1 \oplus W2$; (d) Comparison of solution in all resolution levels; (e) Zoom in of (d) in base region; (f) Zoom in of (d) in another region.

Table 5.3: The full scale, reduced and remaining collocation point number at different resolution levels.

resolution level	full scale collocation point number	deleted collocation point number	remaining collocation point number
V_0	65	0	65
W_0	64	34	30
W_1	128	88	40
W_2	256	221	35
Total	513	343	170

of Figure 5.20(d) as shown in Figure 5.20(e) and (f). From these graph, we see that the solution is getting more and more accurate when the resolution level is increasing, and solution converges at space W_2.

5.2.3 Transient Solution

In this section, we investigate the transistor's transient response.

We will show the case when base-emitter and base-collector voltages are as shown in Figure 5.21(a) and (b) respectively. Here the base-emitter junction is forward biased, and the base-collector junction is reverse biased. Besides, let the base current be zero. Note that the transistor are slightly biased, because the transistor is slightly doped.

With such excitation, we get the transient solutions of ψ, ϕ_n and ϕ_p as shown in Figure 5.22(a) and (b). The corresponding electron and hole concentrations' distribution at different time step are shown in Figure 5.23.

In transient simulation, we use the reduced transform matrices with dimension 170 only. So we achieved very efficient computation.

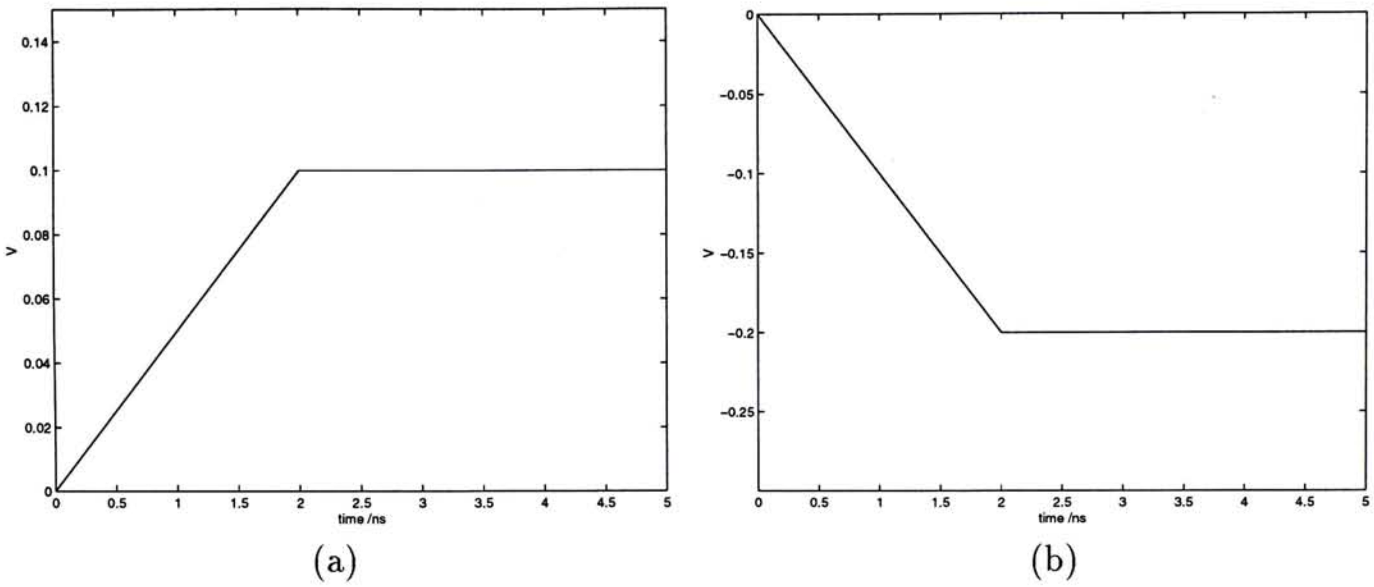


Figure 5.21: (a) Base-Emitter voltage waveform; (b) Base-Collector voltage waveform.

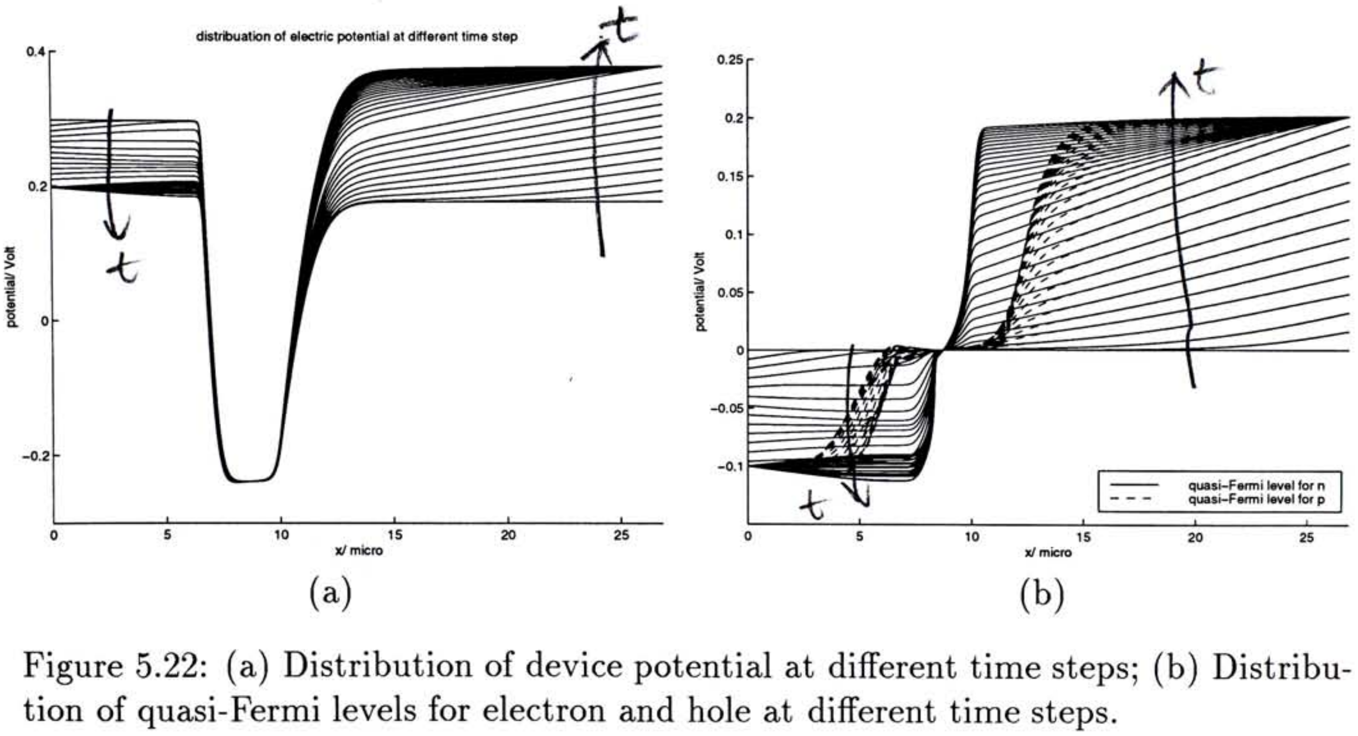


Figure 5.22: (a) Distribution of device potential at different time steps; (b) Distribution of quasi-Fermi levels for electron and hole at different time steps.

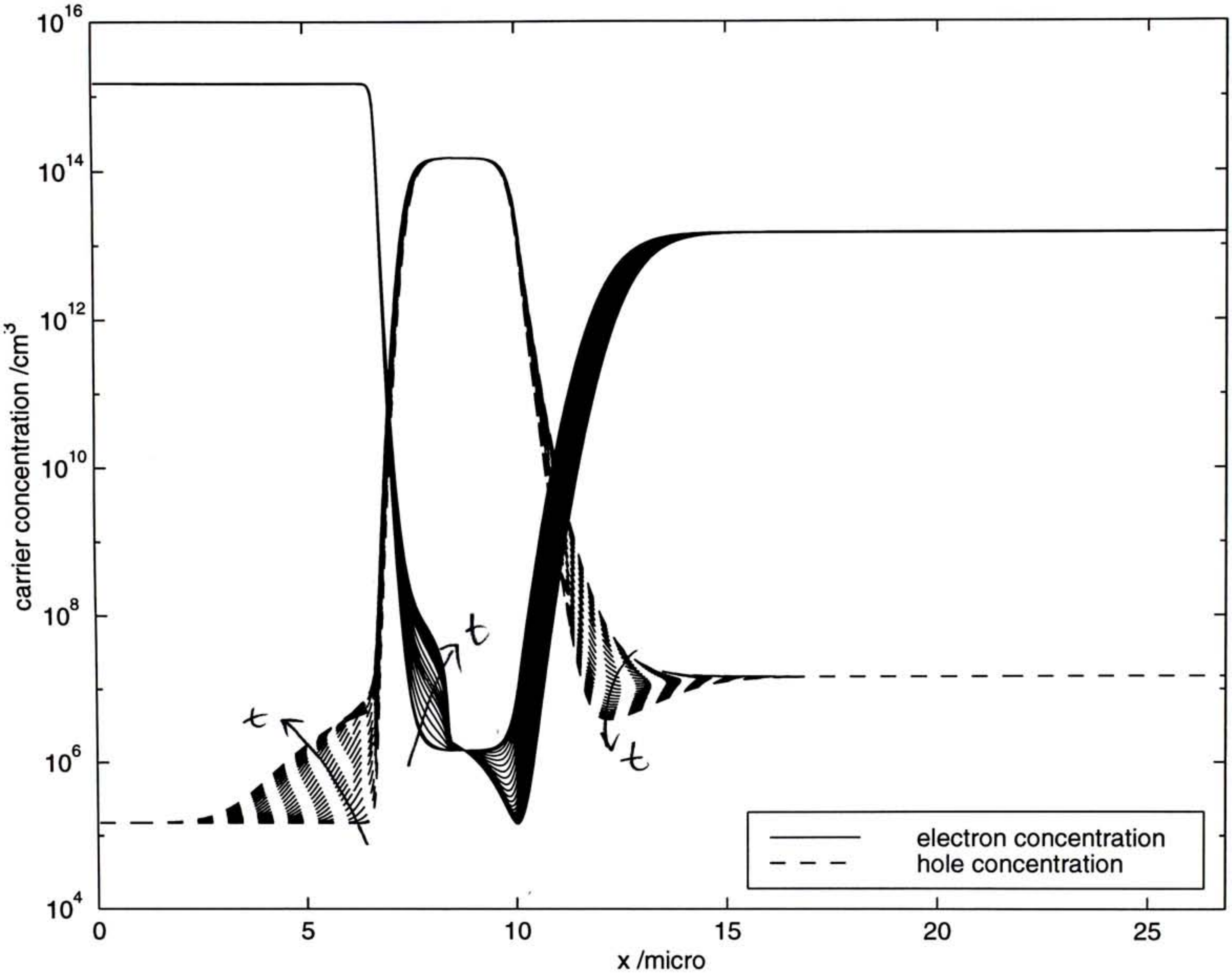


Figure 5.23: Distribution of carriers concentration at different time step.

Chapter 6

Conclusions

In this thesis, we have developed a wavelet-based method for one-dimensional semiconductor device simulation. Wavelet alternative is provided for a device simulator on two parts: grid generator and numerical solver.

The physical device model we used is the classical drift-diffusion model. This model includes five fundamental equations: Poisson's equation, continuity equations for electron and hole, transport equations for electron and hole. After simplification and normalization, three partial differential equations (PDEs) are obtained. The device variables are electric potential, quasi-Fermi potentials for electron and hole. To solve these PDEs, we use *cubic spline wavelet collocation method* for spatial discretization and finite difference method for time discretization. The set of basis functions we used here is a wavelet family developed by Wei Cai [1].

An adaptive method has been developed to delete the redundant collocation points and the corresponding basis functions, based on the localization property of wavelet functions. This collocation point generation process is equivalent to the grid generation process of other device simulators. However, our method is much attractive because the algorithm is very simple and efficient. Once the set of collocation points (grids) is determined, we use it for seeking the steady state and transient responses of devices.

We summarize the advantages of wavelet method in device simulation as following: (1) A large amount of wavelet basis/collocation points which are not necessary can be removed due to the localization property of wavelet functions. (2) Such collocation point (grid) reduction scheme leads to very efficient computation. (3) Full scale transform matrices can be pre-calculated, and reduced transform matrices can be subtracted from it. (4) Easy programming. This method is specially good for devices with local rapid changing variables, which are very common.

Two types of semiconductor devices: p-n junction diode and bipolar transistor, as examples, have been successfully simulated by this method.

Our work is the first step towards the goal of applying wavelet method in commercial device simulator. The future direction of this research should be focused on two dimensional device, and on more complicate device models other than drift-diffusion model, to apply such method on more complicate devices. It is because the most real devices are with two dimensional structure. Besides, the drift-diffusion model is not adequate for many devices, for example, devices with hot electron effect. We believe that wavelet methods will get broad applications in this field in the near future.

Bibliography

[1] Wei Cai and Jianzhong Wang, "Adaptive multiresolution collocation methods for initial boundary value problems of nonlinear pdes," *SIAM Journal on Numerical Analysis*, vol. 33, pp. 936–970, June 1996.

[2] Fung Yuel Chang and Kong Pang PUN, "Discrete b-spline wavelet method for semiconductor device simulation," *Proceedings of the International Symposium on Circuits and Systems, Hong Kong, 1997*, vol. 1, pp. 193–196, 1997.

[3] Jin-chao Xu and Wei-chang Shann, "Galerkin-wavelet methods for two-point boundary value problems," *Numerische Mathematik*, vol. 63, pp. 123–144, 1992.

[4] S. Jaffard, "Wavelet methods for fast resolution of elliptic problems," *SIAM Journal on Numerical Analysis*, vol. 29, pp. 965–986, 1992.

[5] M. Ravachol R. Glowinski, W.M. Lawton and E. Tenenbaum, "Wavelets solution of linear and nonlinear elliptic, parabolic and byperbolic problems in one space dimension," *Computing Methods in Applied Sciences and Engineering, R. Glowinski, A, Lichnewsky eds., SIAM Philadelphia*, pp. 55–120, 1990.

[6] R.D. Middlebrook, *An introduction to junction transistor theory,* New York: Wiley, 1957.

[7] Shyh Wang, *Fundamentals of Semiconductor Theory and Dvice Physics*, Prentice Hall, 1989.

[8] J.R. Hauser N.D. Arora and D.J. Roulston, "Electron and hole mobilities in silicon as a function of concentration and temperature," *IEEE Electron Devices*, vol. ED-29, pp. 292, 1982.

[9] K.K. Thornber, "Relation of drift velocity to low-field mobility and high-field saturation velocity," *Journal of Applied Physics*, vol. 51, pp. 2127–2136, 1980.

[10] R.N. Hall, "Electron-hole recombination in germanium," *Physics review*, vol. 87, pp. 387, 1952.

[11] W. Shockley and W.T. Read, "Statistics of the recombination of holes and electrons," *Physics Review*, vol. 87, pp. 835–842, Sept 1952.

[12] W. Bergner R. Kircher, *Three-Dimensional Simulation of Semiconductor Devices*, Birkhäuser Verlag, 1991.

[13] A. De Mari, "An accurate numerical steady-state one-dimensional solution of the p-n junction," *Solid-State Electronics*, vol. 11, pp. 33–58, 1968.

[14] Kendall E. Atkinson, *An Introduction to Numerical Analysis*, Wiley, 1989.

[15] Stephen D. Mobbs, "Numerical techniques - the finite element method," in *Semiconductor Device Modelling*, Christopher M. Snowden, Ed. Springer-Verlag, 1989.

[16] B.V. Gokhale, "Numerical solutions for a one-dimensional silicon n-p-n transistor," *IEEE Transactions on Electron Devices*, vol. ED-17, pp. 594–602, Aug. 1970.

[17] H.K. Gummel, "A self-consistent iterative scheme for one-dimensional steady state transistor calculations," *IEEE Transactions on Electron Devices*, vol. ED-11, pp. 455–465, Oct. 1964.

[18] Giovanni Naldi Silvia Bertoluzza and Jean Christophe Ravel, "Wavelet methods for the numerical solution of boundary value on the interval," pp. 425–448, 1994.

[19] C.H. Hsiao C.F. Chan, "Haar wavelet method for solving lumped and distributed-parameter systems," *IEE Proceeding -Control Theory Application*, pp. 87–94, January 1997.

[20] N.Chen D.Zhou and W.Cai, "A fast wavelet collocation method for high-speed vlsi circuit simulation," *ICCAD 95, San Jose, California*, pp. 115–122, 1995.