Efficient Techniques for Wideband Impedance Extraction of Complex 3-D Geometries

by

Zhenhai Zhu

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

August 2002

Author . . . .
Department of Electrical Engineering and Computer Science
August 18, 2002

Certified by . . . .
Jacob K. White
Professor
Thesis Supervisor

Accepted by . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Efficient Techniques for Wideband Impedance Extraction of Complex 3-D Geometries

by

Zhenhai Zhu

Submitted to the Department of Electrical Engineering and Computer Science
on August 18, 2002, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

This thesis describes the efficient computation of frequency-dependent impedances for complex three-dimensional geometires of conductors from zero frequency to microwave frequencies. Previous state-of-the-art accelarated fast solver (FastHenry) uses a formulation based upon magneto-quasi-static (MQS) assumption and hence could not consider capacitive effect. In addition, the frequency-dependent volume filaments used in FastHenry renders the computational cost prohibitive at high frequencies due to skin-effect. In this thesis, a surface integral formulation combined with a pre-corrected FFT algorithm is used to compute the impedance matrix in nearly order n time and memory, where n is the number of surface panels. Computational results are given to demonstrate that the new algorithm can perform MQS, electro-magneto-quasi-static and fullwave analysis of realistic integrated circuit interconnect and packaging problems using a fixed set of surface panels across wide frequency range.

Thesis Supervisor: Jacob K. White
Title: Professor

Acknowledgments

First and primary thanks must go to professor Jacob White, who has given me many advices and strong support.

I would like to acknowledge my collaborators Dr. Jingfang Huang and Dr. Ben Song, who have made numerous contributions to this work.

I also want to thank my office mate, Luca Daniel. We have had many fruitful technical discussions.

Thanks should also go to our student system administrators Michal Rewienski and Tom Klemas.

And last but certainly not the least, I want to thank my wife, Jing Wang, for her love, patience and constant support.

The work needed to produce the thesis was sponsored by the Semiconductor Research Corporation, the NSF program in computer-aided design, and grants from Synopsys, Compaq and Intel.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 motivation

The layout parasitics in critical nets in high frequency analog and high speed digital integrated circuits must be analyzed using methods that take into account distributed resistive, capacitive and inductive effects, and may even require a careful treatment of radiation. To extract such impedances requires detailed electromagnetic analysis over a wide frequency range, usually from zero to hundreds of giga hertz [15].

It is widely agreed that the only approaches that have proven to be capable of detailed electromagnetic analysis of complicated integrated circuit interconnects are the accelerated integral equation methods like those used in FastCap [29] and FastHenry [19]. Even though the integral equation method is a well studied subject [11, 47, 10], there does not exist a fast integral equation solver that solves Maxwell's equations in general 3D structures with lossy conductors which is accurate from zero frequency to microwave frequencies.

1.2 Integral Formulations

Many integral formulations have been developed and can be generally categorized into four kinds according to the state variables used in these formulations. 1) Formulations using the field variables E and H have been used for decades to solve the radiation and scattering problems [11, 47] as well as eddy current problems [23, 38]. The well-known formulations include the electric field integral equation (EFIE) and magnetic field integral equation (MFIE) [4, 47], which are also known as Stratton-Chu's formulation [42, 22]. 2) Formulations using the current and charge as state variable include the mixed potential integral equation (MPIE) formulation [11]. 3) Formulations using vector and scalar potentials as state variable have various forms and are very commonly used for solving eddy current problems [30]. 4) Formulations using virtual sources, such as virtual current or charge, are also commonly used for solving eddy current problems [26, 17].

It is well-known that EFIE and MFIE formulations are not guaranteed to produce a unique solution at interior resonant frequencies for closed structures [47, 5]. Many remedies have been proposed [31]. But there still remain many unsolved problems. So far, no wideband fullwave analysis program has been developed based upon these formulations.

The MPIE formulation has been extensively used for the analysis of microstrip structures [28, 3, 2, 24] and the arbitrary shaped conductors with only surface current [35]. It was recognized in [27] that MPIE has accuracy problem at low frequencies. The so-called loop/star and loop/tree basis fucntions were used to overcome this low-frequency problem [27, 50]. The MPIE formulation has also been used for the analysis of interconnects in VLSI or analog circuits. In this case, it is also known as the PEEC model [13]. Interestingly, simply becaue the PEEC model uses a different excitation term than the one used in MPIE for scattering problems, the cause of the low-frequency problem identified in [27] is eliminated. Results of the MQS analysis in [19] and EMQS analysis in [18] have clearly demonstrated that the PEEC model could produce accurate results across wide frequency range, from zero to hundreds of giga hertz. However, unlike the microstrip structures, which are usually approximated by zero-thickness perfect or lossy conductors [28, 3, 2, 24], typical interconnect structures are lossy conductors with finite thickness. Because of the skin effect, analyzing them involves a frequency-dependent discretization of the interior of conductors and the substrate ground. At high frequencies, this kind of discretization usually renders the number of piecewise costant basis functions (also called filaments) to be prohibitively large. Recently, an entire-domain basis scheme has shown some promise to remedy the situation [25], but we have yet to see that it will eventually lead to a wideband fast MaxWell's equation solver for general 3D structures.

The motivation behind this thesis is to find a numerically stable surface integral formulation, as such formulations avoid a frequency-dependent discretization of the interior of conductors and the substrate. The formulation should be capable of wideband analysis and it should also be easily accelarated by the well-established techniques, such as fast multiple method [8, 7] or the pre-corrected FFT algorithm [32].

One recently developed surface integral formulation has shown promise [46, 48], but was plagued with numerical difficulties of poorly understood origin. It was shown in [52] that one of that formulation's difficulties was related to inaccuracy in the approach to evaluate integrals over discretization panels, and a more accurate approach based on an adapted piecewise quadrature scheme was proposed. Numerical examples in [52] have demonstrated that the formulation is indeed valid across wide frequency range, from zero to at least hundreds of giga hertz. It is also shown in [52] that the condition number of the original system of integral equations can be reduced by differentiating one of the integral equations. With these issues being resolved, the formulation is accelaration-ready.

1.3 Fast Integral Equation Solvers

Fast Multiple Method (FMM) [8, 7] has seen its success in many applications, such as electrostatic analysis in FastCap [29], magneto-quasi-static analysis in FastHenry [19], and fullwave analysis in the Fast Illinois Solver Code [41]. But it is kernel-dependent by nature. On the other hand, the pre-corrected FFT (pFFT) algorithm [33], which has been successfully used in many applications [32, 49], is nearly kernel-independent. Since our surface integral formulation has a number of different kernels, even hyper-singular ones, the pFFT algorithm seems to suit our formulation better. In addition, as a by-product of our work, we also want to develop a flexible and stand-alone fast integral equation solver that can handle various kinds of integral operators, at least the ones that are most commonly used in the boundary element method framework [10]. To the best of our knowledge, this kind of solver has not yet been developed and made available to the public. Using this fast solver as a powerful engine and based on the improved surface integral formulation in [52], we have developed a fast impedance extraction program, fastImp. Experiments using several large examples show that fastImp could perform MQS, EMQS and fullwave analysis of interconnect structures with hundreds thousands of unknowns from zero frequency all the way to hundreds of giga hertz.

1.4 Thesis Outline

In chapter 2, we will derive the surface integral formulation and show its connection to the EFIE and the MPIE and why it is widebanded. In chapter 3, we will show how the piecewise quadrature scheme improves the acuracy of panel integration and that it solves the low frequency problem in [48]. In chapter 4, we will explain how to accelarate the complicated integral operators in our surface formulation with the pFFT algorithm. Several large examples in chapter 5 are used to demonstrate the speed and the accuracy of fastImp. And finally chapter 6 concludes the thesis.

Chapter 2

Derivation of the Surface Integral Formulation

Figure 2-1 is a general description of the 3D interconnect structures embedded in an isotropic and homogeneous medium. We assume that each conductor $V_i, i = 1, 2, ..., n$, is piecewise homogeneous and the homogeneous medium region is always denoted by V_0.

We will derive the surface integral formulation from a different viewpoint than the one used in [48]. This way, it is very easy to see its connections to the MPIE formulation and the EFIE formulation.

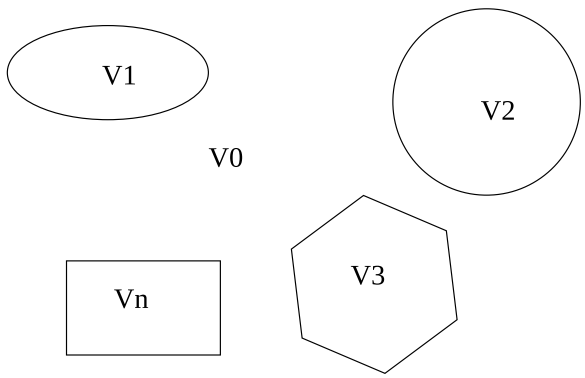

Figure 2-1: A general description of the 3D interconnect structures embedded in homogeneous medium

2.1 Governing equations

In time-harmonic form, the independent and definite forms of MaxWell's equations are [44]

$$\nabla\times\vec{E} = -j\omega\mu\vec{H} \tag{2.1}$$

$$\nabla\times\vec{H} = \vec{J} + j\omega\varepsilon\vec{E} \tag{2.2}$$

$$\nabla\cdot\vec{J} = -j\omega\rho \tag{2.3}$$

$$\vec{J} = \sigma\vec{E} \tag{2.4}$$

Equations (2.1) and (2.2) imply

$$\nabla\times\nabla\times\vec{E} - \omega^2\varepsilon\mu\vec{E} = -j\omega\mu\vec{J}. \tag{2.5}$$

It is obvious that equations (2.1)-(2.4) are equivalent to equations (2.1) and (2.3)-(2.5). Since the charge inside a good conductor is zero [34] and each conductor is homogeneous, substitution (2.4) into (2.3) and setting to zero right side of (2.3) yields

$$\nabla\cdot\vec{E}(\vec{r}) = 0, \;\; \vec{r}\in V_i. \tag{2.6}$$

Hence equation (2.5) can be reduced to

$$(\nabla^2 + \omega^2\varepsilon\mu)\vec{E}(\vec{r}) = j\omega\mu\vec{J}(\vec{r}), \;\; \vec{r}\in V_i. \tag{2.7}$$

It should be note that the combination of equation (2.6) and (2.7), not just equation (2.7) alone, is equivalent to equation (2.5).

Equations (2.1), (2.4), (2.6) and (2.7) are the governing equations inside each conductor V_i, and equations (2.1)-(2.4) are the governing equations in the homogeneous medium, as well as inside the conductors whenever it is necessary.

2.2 Boundary conditions

The surface of each conductor has two parts: contact surface and non-contact surface, as shown in figure 2-2. Contact is the artificially exposed surface. It is created primarily because we want to use the divide-and-conquer strategy to seperate a block of 3D interconnect from other parts within a large chip. Since contacts are acctually in the interior of a conductor, it is reasonable to assume that the charge on the contacts is zero. So equation (2.6) also holds true on the contacts.

Becasue of the nature of commonly used strategy to decompose a large chip into many smaller blocks, the conductors connected to these contacts are usually long and thin signal lines. Hence it is reasonable to assume that the current goes into these contacts does not have the transversal components, i.e., $\vec{t} \cdot \vec{J} = 0$, where $\vec{t}$ is the unit tangential vector on the contacts. Substituting (2.4) into it yields

$$\vec{t}(\vec{r}) \cdot \vec{E}(\vec{r}) = 0, \text{ if } \vec{r} \text{ is on a contact.} \tag{2.8}$$

Equations (2.6) and (2.8) imply

$$\frac{\partial E_n(\vec{r})}{\partial n(\vec{r})} = 0, \text{ if } \vec{r} \text{ is on a contact.} \tag{2.9}$$

On the other hand, since charge on a non-contact surface is not necessarily zero, in view of (2.3), the boundary condition becomes [44]

$$E_n(\vec{r}) = \frac{j\omega\rho(\vec{r})}{\sigma}, \text{ if } \vec{r} \text{ is on non-contact surface.} \tag{2.10}$$

It shoule be noted that $\vec{E}$ and $\frac{\partial \vec{E}}{\partial n}$ in this section are defined only on the inner side of conductor surface. In fact, in this paper we try to avoid using the matching boundary conditions commonly used in solving scattering and radiation problems. The reason will be made clear in section 2.4.

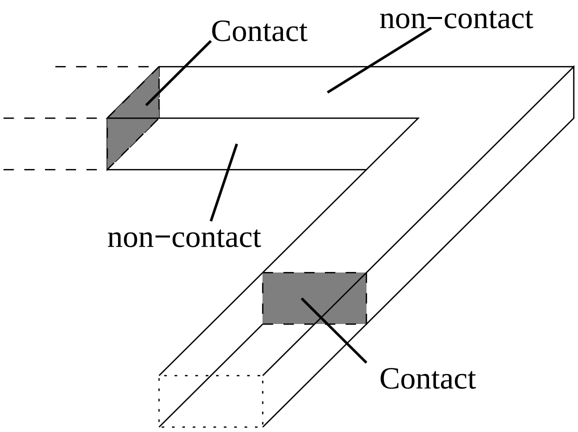

Figure 2-2: The surface of a 3D interconnect conductor

2.3 Surface integral representation

Thanks to Green's second identity, the surface integral representation of the solution to equation (2.7) inside conductor V_i is [4]

$$T\vec{E}(\vec{r}) = \int_{S_i} dS'\left(G_0(\vec{r},\vec{r'})\frac{\partial \vec{E}(\vec{r'})}{\partial n(\vec{r'})} - \frac{\partial G_0(\vec{r},\vec{r'})}{\partial n(\vec{r'})}\vec{E}(\vec{r'})\right) + j\omega\mu\int_{V_i} dV' G_0(\vec{r},\vec{r'})\vec{J}(\vec{r'}) \tag{2.11}$$

where

$$G_0(\vec{r},\vec{r'}) = \frac{e^{jk_0|\vec{r}-\vec{r'}|}}{4\pi|\vec{r}-\vec{r'}|},\ k_0 = \omega\sqrt{\varepsilon\mu}\ , \tag{2.12}$$

$$T = \begin{cases} 1 & \text{if } \vec{r} \in V_i \\ 1/2 & \text{if } \vec{r} \in S_i \\ 0 & \text{otherwise} \end{cases} \tag{2.13}$$

and S_i is the surface of conductor V_i. When $\vec{r} \in S_i$ the surface integral in (2.11) should be the principal value integral. If we write equation (2.11) for each conductor separately but let $\vec{r}$ be fixed on the surface of a particular conductor V_k, and then sum up these equations, we obtain

$$\frac{1}{2}\vec{E}(\vec{r}) = \int_{S} dS'\left(G_0(\vec{r},\vec{r'})\frac{\partial \vec{E}(\vec{r'})}{\partial n(\vec{r'})} - \frac{\partial G_0(\vec{r},\vec{r'})}{\partial n(\vec{r'})}\vec{E}(\vec{r'})\right) + j\omega\mu\int_{V} dV' G_0(\vec{r},\vec{r'})\vec{J}(\vec{r'}) \tag{2.14}$$

where S is the union of all conductor surfaces and V is the union of all conductor regions, and $\vec{r} \in S_k, k = 1, 2, ..., n$.

Substituting (2.4) into (2.7) yields,

$$\nabla^2 \vec{E}(\vec{r}) + (\omega^2 \varepsilon \mu - j\omega\mu\sigma_i)\vec{E}(\vec{r}) = 0, \ \ \vec{r} \in V_i \tag{2.15}$$

where σ_i is the conductivity of the conductor V_i. Again, thanks to Green's second identity, the surface integral representation of the solution to equation (2.15) is

$$\frac{1}{2}\vec{E}(\vec{r}) = \int_{S_i} dS' (G_1(\vec{r}, \vec{r'}) \frac{\partial \vec{E}(\vec{r'})}{\partial n(\vec{r'})} - \frac{\partial G_1(\vec{r}, \vec{r'})}{\partial n(\vec{r'})} \vec{E}(\vec{r'})), \quad \vec{r} \in S_i \tag{2.16}$$

where

$$G_1(\vec{r}, \vec{r'}) = \frac{e^{jk_1|\vec{r}-\vec{r'}|}}{4\pi|\vec{r} - \vec{r'}|}, \ k_1 = -\sqrt{\omega^2 \varepsilon \mu - j\omega\mu\sigma_i} \ . \tag{2.17}$$

Since (2.14) and (2.16) are the formal solutions to the same equation in slightly different forms, they are obviously equivalent. We use both of them instead of just one merely for the derivation purpose.

So far, only the formal solutions to equation (2.7) inside each conductor has been found. To find the formal solution to the governing equations in region V_0, the homogeneous medium, we turn to the MPIE. The reason we still want to use the MPIE will be made clear shortly. Now each conductor is treated as a volume current source. Same as the standard MPIE formulation [11], the electric field everywhere, including the interior of every conductor, is

$$\vec{E}(\vec{r}) = -j\omega\vec{A} - \nabla\phi(\vec{r}) = j\omega\mu \int_V dV' G_0(\vec{r}, \vec{r'}) \vec{J}(\vec{r'}) - \nabla\phi(\vec{r}) \tag{2.18}$$

where

$$\phi(\vec{r}) = \int_S dS' \frac{\rho(\vec{r'})}{\varepsilon} G_0(\vec{r}, \vec{r'}). \tag{2.19}$$

Unlike standard MPIE, the lorentz gauge $\nabla \cdot \vec{A} + j\omega\varepsilon\mu\phi = 0$ is not explicitly enforced because it is equivalent to equation (2.6), which is explicitly enforced in our formulation. Now it is clear that had equation (2.5) instead of equations (2.6) and (2.7) been used as the governing equations, we would have to enforce lorentz gauge, which would introduce the vector potential $\vec{A}$ or ultimately the volume integral term into our formulation.

Let $\vec{r} \in S_k$ in equation (2.18) and subtract it from equation (2.14), we obtain

$$-\frac{1}{2}\vec{E}(\vec{r}) = \int_S dS'(G_0(\vec{r},\vec{r'})\frac{\partial \vec{E}(\vec{r'})}{\partial n(\vec{r'})} - \frac{\partial G_0(\vec{r},\vec{r'})}{\partial n(\vec{r'})}\vec{E}(\vec{r'})) + \nabla\phi(\vec{r}),\ \vec{r} \in S_k \tag{2.20}$$

where $k = 1, 2, ..., n$. It should be noted that the integral representation (2.20) is no longer the formal solution to equation (2.7), hence it is not equivalent to (2.16) any more. Now we have found the surface integral representation of the formal solutions to the governing equations inside conductors and homogeneous medium. It should be noted that the surface integrals in (2.14), (2.16) and (2.20) are all principal value ones.

2.4 Connections to EFIE and MPIE

There are two somewhat unconventional ingredients in our formulation: 1) there is no matching boundary conditions; 2) the mixture of EFIE and MPIE is used. Each ingredient has its own ramifications.

Because a contact is the virtual boundary between two pieces of conductors, as shown in figure 2-2, and we do not have any infomation about the one that is not included in the 3D interconnect structure, we want to avoid matching bounbary conditions on the contact. To be consistent, we have to avoid matching bounbary conditions on all surface.

The MPIE has the volume integral term but does not need matching boundary conditions. On the other hand, the EFIE needs matching boundary conditons but does not have the volume integral term if currents inside conductors are not treated as sources. Incidentally, the volume integral term in equation (2.18) is the same as the one in equation (2.14). And we have used this fact to cancel out this undesirable volume integral term. So EFIE and MPIE complement each other well and their combination results in a true surface integral representation.

In addition, we want to use PEEC idea to compute the impedance since it natually fits in the circuit simulation environment. So we impose voltage on the contacts and compute the current goes into the contacts and then use the voltage-current definition $Z = V/I$ to compute impedance. This is another reason we want to use the MPIE because one of its state variable ϕ is natually related to voltage. From our experience it is rather cubersome to introduce excitation by merely using electric and/or magnetic fields as state variables.

It is worth noting that the EFIE formulation in this paper is slightly different from the standard one. There are a few equivalent forms of EFIE, the one closest to equation (2.16) is [47]

$$\frac{1}{2}\vec{E}(\vec{r}) = \int_{S_i} dS'[G_1(\vec{r},\vec{r'})\frac{\partial \vec{E}(\vec{r'})}{\partial n(\vec{r'})} - \frac{\partial G_1(\vec{r},\vec{r'})}{\partial n(\vec{r'})}\vec{E}(\vec{r'}) + \hat{n}(\vec{r'})G_1(\vec{r},\vec{r'})(\nabla' \cdot \vec{E}(\vec{r'}))], \quad \vec{r} \in S_i. \tag{2.21}$$

And the one closest to equation (2.11) is equation (2.21) with the addition of a volume integral term exactly same as the one in equation (2.11). The starndard EFIE is derived from vector Helmholtz equation (2.5) using Green's second identity in vector form. And equation (2.6) is not explicitly enforced. However, as discussed before, equation (2.6) must be enforced in our formulation. This is why we choose equation (2.16) rather than equation (2.21) in standard EFIE.

2.5 Surface formulation

In light of the observation made in section 2.4, we introduce one last equation, equation (2.29), into our formulation. We follow the convention in the PEEC model, using the difference between ϕ on two contacts of the same conductor as the voltage excitation term [19, 18].

In summary, the formulation for fullwave analysis consists of the following equations

$$\frac{1}{2}\vec{E}(\vec{r}) = \int_{S_i} dS'(G_1(\vec{r},\vec{r'})\frac{\partial \vec{E}(\vec{r'})}{\partial n(\vec{r'})} - \frac{\partial G_1(\vec{r},\vec{r'})}{\partial n(\vec{r'})}\vec{E}(\vec{r'})), \quad \vec{r} \in S_i, \tag{2.22}$$

$$-\hat{t}(\vec{r}) \cdot \frac{1}{2}\vec{E}(\vec{r}) = \hat{t}(\vec{r}) \cdot \int_{S} dS'(G_0(\vec{r},\vec{r'})\frac{\partial \vec{E}(\vec{r'})}{\partial n(\vec{r'})} - \frac{\partial G_0(\vec{r},\vec{r'})}{\partial n(\vec{r'})}\vec{E}(\vec{r'})) + \hat{t}(\vec{r}) \cdot \nabla\phi(\vec{r}), \quad \vec{r} \in S_{nc}. \tag{2.23}$$

$$\phi(\vec{r}) = \int_{S} dS' \frac{\rho(\vec{r'})}{\varepsilon} G_0(\vec{r},\vec{r'}), \quad \vec{r} \in S. \tag{2.24}$$

$$E_n(\vec{r}) = \frac{j\omega\rho(\vec{r})}{\sigma}, \quad \vec{r} \in S_{nc} \tag{2.25}$$

$$\hat{t}(\vec{r}) \cdot \vec{E}(\vec{r}) = 0, \quad \vec{r} \in S_c \tag{2.26}$$

$$\frac{\partial E_n(\vec{r})}{\partial n(\vec{r})} = 0, \quad \vec{r} \in S_c \tag{2.27}$$

$$\nabla \cdot \vec{E}(\vec{r}) = 0, \quad \vec{r} \in S_{nc} \tag{2.28}$$

$$\phi(\vec{r}) = constant, \quad \vec{r} \in S_c \tag{2.29}$$

where S_{nc} and S_c are the non-contact part and contact part of the conductor surface S, respectively.

The formulation has eight scalar state variables, E_x, E_y, E_z, $\frac{\partial Ex}{\partial n}$, $\frac{\partial Ey}{\partial n}$, $\frac{\partial Ez}{\partial n}$, ϕ and ρ. Since there is no matching boundary condition, all components of $\vec{E}$ and $\frac{\partial \vec{E}}{\partial n}$ are on the inner side of conductor surface. Because equation (2.20) along the normal direction is not enforced, the total number of scalar equations is also eight.

For EMQS analysis, k_0 in equation (2.12) becomes zero and the term $\omega^2\varepsilon\mu$ in equation (2.17) should be dropped [12]. But the number of state variables is unchanged. For MQS analysis, on top of above simplification, the charge ρ in equation (2.25) becomes zero [12]. Hence it becomes redundant and is not used as a state variable and equation (2.24) is not used either. Hence the total number of scalar unknowns and equations becomes seven.

2.6 Why this surface formulation is widebanded

Numerical results in [52] have clearly shown that our formulation is valid from zero frequency to microwave frequencies. Since we have established in section 2.4 that our formulation is a combination of EFIE and MPIE, we are ready to explain why it is widebanded. The reason turns out to be rather simple: both EFIE and MPIE are widebanded themselves for the analysis of interconects. The following is our reasoning.

When the MPIE is used to solve scattering and radiation problems, the known is E^i, the incidence field for scattering problems or the excitation field for antenna problems. The governing equations are [28, 3, 2]

$$-\vec{E}^s = j\omega\vec{A} + \nabla\phi = \vec{E}^i \tag{2.30}$$

$$\nabla_s \cdot \vec{J} = -j\omega\rho_s \tag{2.31}$$

$$\phi(\vec{r}) = \int_S dS' G_0(\vec{r},\vec{r'})\rho_s(\vec{r'}) \tag{2.32}$$

$$\vec{A}(\vec{r}) = \int_V dV' G_0(\vec{r},\vec{r'})\vec{J}(\vec{r'}) \tag{2.33}$$

where ∇_s is the surface divergence. It was pointed out in [27] that when the frequency is sufficiently low, the vector potential contributions to the elements of system matrix are insignificant compared with the scalar potential contributions. As a result, the vector potential contributions are lost. The remaining scalar potential contributions depend only on $\nabla_s \cdot \vec{J}$. Knowledge of $\nabla_s \cdot \vec{J}$ is not sufficient to determine $\vec{J}$. Therefore, the solutions are inaccurate at low frequencies. A loop/tree or loop/star basis function pair can be used to separate the contributions from vector potential A and scalar potential ϕ to the system matrix element, and hence solves the low-frequency problem [50].

However, when the same MPIE or the PEEC model is used for the analysis of interconnects, the first governing equation is slightly modefied. It becomes

$$\vec{E} = -j\omega\vec{A} - \nabla\phi = \frac{\vec{J}}{\sigma} \tag{2.34}$$

and the excitation is the user-specified scalar potential or voltage on the contacts. When the frequency is identically zero, equation (2.34) becomes

$$-\nabla\phi = \frac{\vec{J}}{\sigma} \tag{2.35}$$

which is the equivalence of $V = RI$ in circuit [34]. This is exactly the kind of low-frequency behavior we expect! Hence we do not see any low-frequency problem at all. We could use similar reasoning to verify that the MPIE does not have high-frequency prolems either. So it is a wideband formulation by itself.

As for EFIE, it only has nonuniqueness problem at resonance frequencies of closed perfect conductors. Since interconnects are usually lossy open structures, we should not have this problem. Hence the EFIE is also a wideband formualtion by itself when it is used for the analysis of interconnects.

Since our formulation is a combination of these two widebanded formulations, it should not be a surprise that our formulation turns out to be widebanded too.

For the MPIE, independent of the accuracy of the system matrix element at low requencies, there is a condition number issue. In [51], on top of the loop/star and loop/tree basis functions, a preconditioner was proposed to reduce the number of iterations of an iterative matrix solver for the analysis of the scattering and radiation problems. For the analysis of interconnects, a mesh current idea, which enforces $\nabla \cdot \vec{J} = 0$ implicitly, was used to make the system matrix better conditioned in [19, 18]. In our formulation, we use a sparse pre-conditioner matrix to reduce the iterations of the matrix solver GMRES [40]. It is constructed by ignoring the interaction between panels in integral equations (2.22), (2.23) and (2.24), and using equations (2.25), (2.26), (2.27), (2.28) and (2.29) directly. It is shown in [52] that the condition number of the system can be further improved by replacing equation (2.22) with its normal derivative, i.e.,

$$\frac{1}{2}\frac{\partial}{\partial n(\vec{r})}\vec{E}(\vec{r}) = \frac{\partial}{\partial n(\vec{r})}[\int_{S_i} dS'(G_1(\vec{r},\vec{r'})\frac{\partial \vec{E}(\vec{r'})}{\partial n(\vec{r'})} - \frac{\partial G_1(\vec{r},\vec{r'})}{\partial n(\vec{r'})}\vec{E}(\vec{r'}))]. \tag{2.36}$$

2.7 Discretization of the formulation

Applying the integral form of equation (2.28)

$$\oint \vec{E}(\vec{r})dS = 0 \tag{2.37}$$

to the surface of an infinitely thin small rectangular box beneath the conductor surface, we obtain

$$\int_{\Gamma} dx\vec{E}_t(x) \cdot (\hat{n}(x) \times \hat{l}(x)) - \int_{\Omega} dS(\vec{r})\frac{\partial E_n(\vec{r})}{\partial n(\vec{r})} = 0 \tag{2.38}$$

where Ω is the top of the box, Γ is the periphery of Ω. It is easy to see that equations (2.26) and (2.27) are sufficient conditions for equation (2.38) to hold true. Since these two

Figure 2-3: An infintely thin small rectangular box benneath the conductor surface

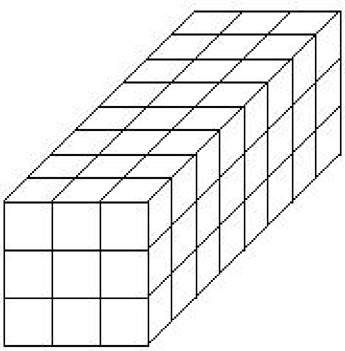

Figure 2-4: Panel discretization

equations are much simpler, we use them for contact surface and use equation (2.38) only for non-contact surface.

In order to discretize the integral equations (2.22), (2.23) and (2.24), a piecewise constant centroid collocation scheme is used in this paper. The conductor surface is discretized into N flat quadrilateral panels as shown in figure 2-4. Seven unknowns are associated with each panel: E_x, E_y, E_z, $\frac{\partial Ex}{\partial n}$, $\frac{\partial Ey}{\partial n}$, $\frac{\partial Ez}{\partial n}$ and ρ. The scalar potential ϕ is associated with the panel vertices. With this setting, equations (2.25), (2.26), (2.27), (2.38) and (2.29) become simple algebraic equations. Please refer to [48] for more details on discretization.

Chapter 3

Improving the accuracy of panel integration

3.1 Definition

After discretization, the integrals over conductor surface S or S_i are replaced by the summation of integrals over panels. These integrals are

$$I_1(\vec{r}) = \int_{P_i} dS' G(\vec{r}, \vec{r'}) \tag{3.1}$$

$$I_2(\vec{r}) = \int_{P_i} dS' \frac{\partial G(\vec{r}, \vec{r'})}{\partial n(\vec{r'})} = \hat{n}(P_i) \cdot \int_{P_i} dS' \nabla_{\vec{r'}} G(\vec{r}, \vec{r'}) \tag{3.2}$$

$$I_3(\vec{r}) = \frac{\partial}{\partial n(\vec{r})} \int_{P_i} dS' G(\vec{r}, \vec{r'}) = \hat{n}(\vec{r}) \cdot \nabla_{\vec{r}} \int_{P_i} dS' G(\vec{r}, \vec{r'}) \tag{3.3}$$

where P_i is the i-th panel, $\hat{n}(P_i)$ is the unit normal vector on the flat panel P_i, and $G(\vec{r}, \vec{r'})$ is either $G_0(\vec{r}, \vec{r'})$ or $G_1(\vec{r}, \vec{r'})$ defined in (2.12) and (2.17). From the symmetry property of the Green's function, it follows that

$$\int_{P_i} dS' \nabla_{\vec{r'}} G(\vec{r}, \vec{r'}) = -\nabla_{\vec{r}} \int_{P_i} dS' G(\vec{r}, \vec{r'}) = -\nabla_{\vec{r}} I_1(\vec{r}). \tag{3.4}$$

Therefore, to compute the integrals in equation (3.1) (4.43) and (3.3), all we need is to compute $I_1(\vec{r})$ and $\frac{\partial I_1(\vec{r})}{\partial D}$, where D stands for x, y or z.

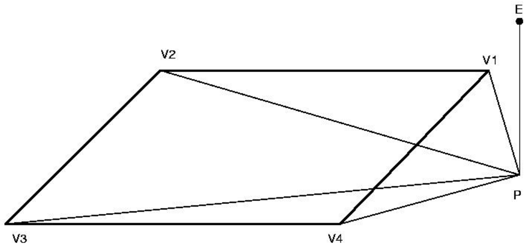

Figure 3-1: Decomposition of an integration over a polygon into several integrations over traingels

3.2 Decomposition

It is shown in [14] that any integration over a polygon is equal to the signed summation of the integration over a chosen set of triangles. The vertices of these triangles are those of the polygon and the projection of the evaluation point onto the plane where the polygon lies, as shown in figure 3-1. To be more precise, let $f(\vec{r})$ be a general integrand, its integration over a polygon in figure 3-1 could be written as

$$\int_S d\vec{r} f(\vec{r}) = \sum_{i=1}^{N} s_i \int_{PV_iV_{i+1}} d\vec{r} f(\vec{r}) \tag{3.5}$$

where N is the number of vertices, $V_{N+1} = V_1$, and $s_i = -1$ if V_iV_{i+1} is clockwise looking from the evaluation point E and $s_i = 1$ if otherwise. This idea was used in [48] to compute the integrals $I_1(\vec{r})$ and $\frac{\partial I_1(\vec{r})}{\partial D}$.

3.3 Desingularization and Reduction to 1-D integration

In a polar coordinate system, a triangle after the decomposition is shown in figure 3-2. Using the relation $R = \sqrt{r^2 + h^2}$ and $RdR = rdr$, the integrals I_1 and $\frac{\partial I_1}{\partial D}$ over this triangle could be rewritten in polar coordinates as

$$\begin{aligned} I_1 &= \int_{\theta_A}^{\theta_B} d\theta \int_0^{r_1(\theta)} rdr \frac{e^{ikR}}{4\pi R} \\ &= \int_{\theta_A}^{\theta_B} d\theta \int_h^{R_1(\theta)} dR \frac{e^{ikR}}{4\pi} \end{aligned}$$

$$= \int_{\theta_A}^{\theta_B} d\theta \frac{e^{ikR_1(\theta)} - e^{ikh}}{4\pi ik} \quad k \neq 0 \tag{3.6}$$

$$or \quad = \int_{\theta_A}^{\theta_B} d\theta \frac{R_1(\theta) - h}{4\pi} \quad k = 0 \tag{3.7}$$

$$\frac{\partial I_1}{\partial D} = \int_{\theta_A}^{\theta_B} d\theta (\frac{e^{ikR_1(\theta)}}{4\pi} \frac{\partial R_1(\theta)}{\partial D} - \frac{e^{ikh}}{4\pi} \frac{\partial h}{\partial D}) \tag{3.8}$$

Now the singularity of the original kernels in I_1 and $\frac{\partial I_1}{\partial D}$ has been eliminated and the 2-D integrations have been reduced to 1-D integrations. The quadrature rule is used to compute the two 1-D integrations in equation (3.6) and (3.8). The shared rapid changing kernel in these two integrals is $f(\theta) = e^{ikR_1(\theta)}$, where $R_1(\theta) = \sqrt{d^2 sec^2(\theta) + h^2}$. When $d << AB$, $\theta_A \approx \frac{-\pi}{2}$ and $\theta_B \approx \frac{\pi}{2}$, and $f(\theta)$ changes rapidly over the interval. Many quadrature points must be used to achieve reasonable accuracy.

3.4 piecewise Quadrature Scheme

A simple variable transformation and a piecewise quadrature scheme can be used to solve the above-mentioned problem. Let $x = dtan(\theta)$, it easily follows that $\frac{d\theta}{dx} = \frac{d}{r^2}$, where $r^2 = d^2 + x^2$. The rapidly changing part of I_1 and $\frac{\partial I_1}{\partial D}$ could be rewritten as

$$\int_{\theta_A}^{\theta_B} d\theta e^{ikR} = \int_{x_A}^{x_B} dx g(x), where\, g(x) = \frac{d}{r^2} e^{ik\sqrt{h^2+r^2}} \tag{3.9}$$

The distribution of the integrand $g(x)$ is shown in the top figure of the figure 3-3. Many quadrature points must still be used to get accurate evaluation because of the rapid variation about $x = 0$. However if we divide the integration domain into two sub-domains, as shown in the middle and the bottom figure of the figure 3-3, and use a piecewise integration scheme, the number of quadrature points needed will be dramatically reduced. The convergence behavior of the integration over the whole domain and over the two sub-domains is shown in figure 3-4. It is clear that the piecewise scheme uses fewer quadrature points, or has higher accuracy if only a small number of quadrature points are used. Unfortunately, this is not appreciated in [48] and a small number (24) of quadrature points are used for the integration over the whole domain. Since the lower the frequency, the smaller the damp-

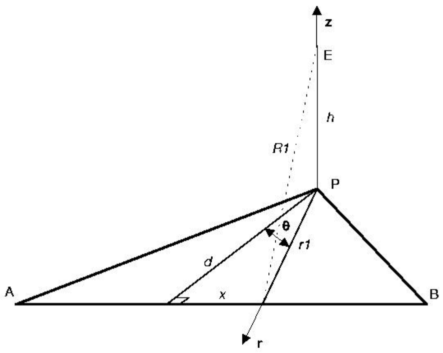

Figure 3-2: Triangle in polar coordinate system, d is the distance between point P and edge AB

ing factor in complex wave number k, hence the higher the peak of the integrand $g(x)$, the formulation in [48] has a low frequency problem.

3.5 Testing examples

We will use two simple examples to validate the propsed piecewise quadrature scheme. The first example is a simple ring structure since the analytical formulas exist for the low-frequency inductance of a ring [9]. The second example is a spiral structure. We compare our results to those of the public domain program FastHenry [19]. In order to compare with the magnetoquasistatic analysis program FastHenry, these two examples were analyzed magnetoquasistaticly.

3.5.1 Ring

The ring is $10mm$ in radius, with a square cross section of the size $0.5mm$ by $0.5mm$. The conductivity is that of the copper, which is 5.8e7. The low frequency inductance calculated using the formula in [9] is 48.89 nH. The results obtained by using FastHenry and the formulation derived in section **??** enhanced with the piecewise quadrature scheme proposed in section **??** are shown in figure 3-5 and 3-6. The two results agree well. The number of filaments used by FastHenry is 960, 3840 and 15360, respectively. The surface formualtion only uses 992 panels across the entire frequency range. It should be noted

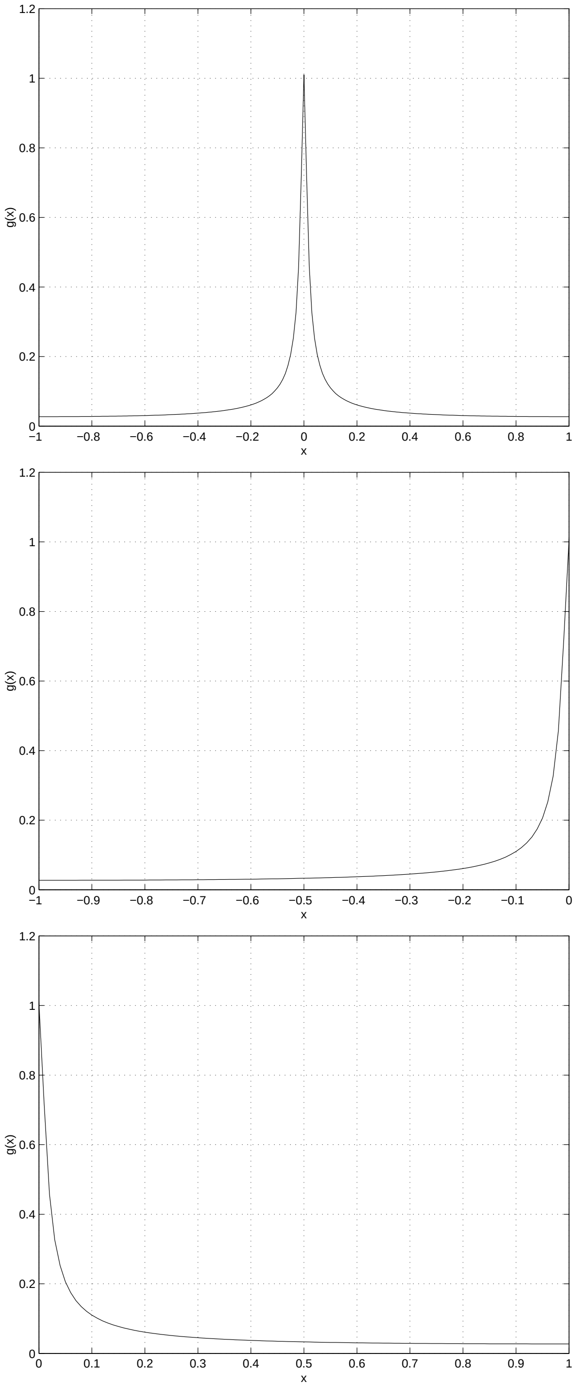

Figure 3-3: Distribution of the integrand, the top figure is the distribution of the original integrand, the middle and the bottom figure are the left and right part of the top figure

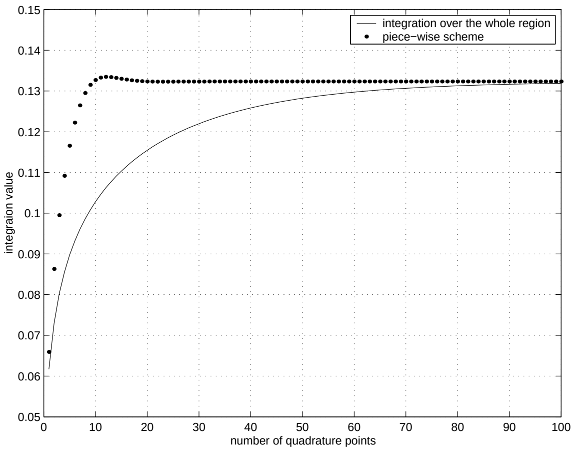

Figure 3-4: convergence behavior of different schemes

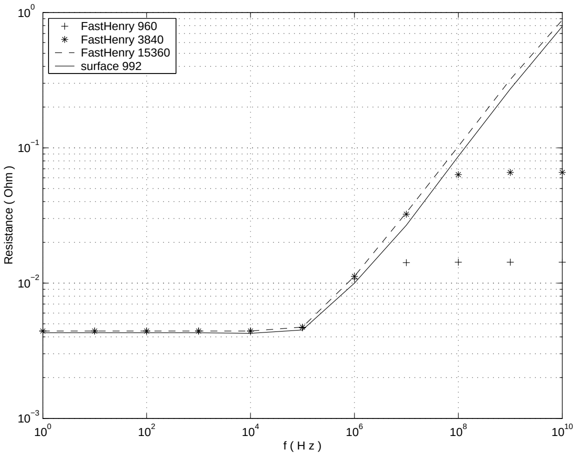

Figure 3-5: Resistance of a ring

that the inductance obtained with the surface formulation is very close to 48.89nH in the low frequency range. This suggests that the low frequency problem reported in [48] has been eliminated without using the linearization technique proposed therein. Also, at high frequency, the resistance scales to the square root of frequency and the inductance drops a little. This suggests that the skin-effect has been well captured. So this ring example does validate our panel integration scheme.

3.5.2 Spiral inductor

The inner radius of the spiral is 10mm. Its cross section is a square of the size 0.5mm by 0.5mm, and the spacing between two succesive revolutions is 0.5mm. The spiral has two

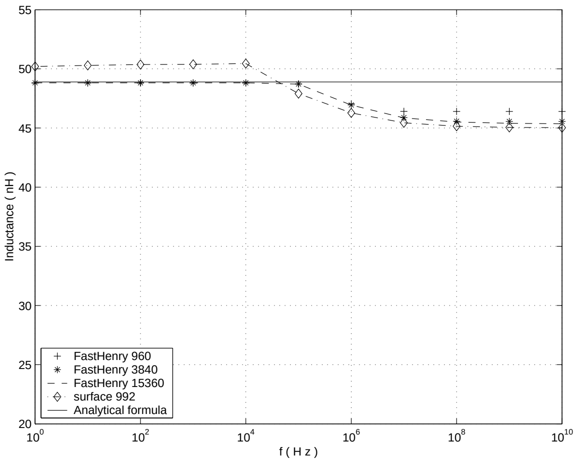

Figure 3-6: Inductance of a ring

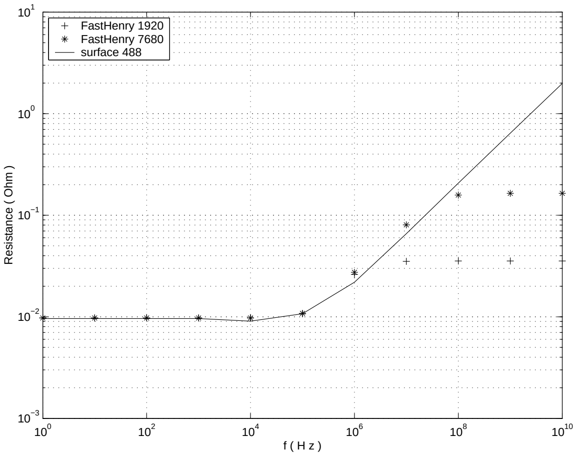

Figure 3-7: Resistance of a spiral

revolutions. At low frequencies, the computed resistance and inductance agree well with those obtained with FastHenry, as shown in figure 3-7 and 3-8. This again validates our panel integration scheme. It is worth mentioning that FastHenry does not capture the skin-effect at high frequencies due to the fixed number of filaments. On the other hand, with a fixed number of panels, the surface integral formualtion has well captured the skin-effect. This example clearly demonstrates the advantage of the surface integral formulation over the volume integral formulation.

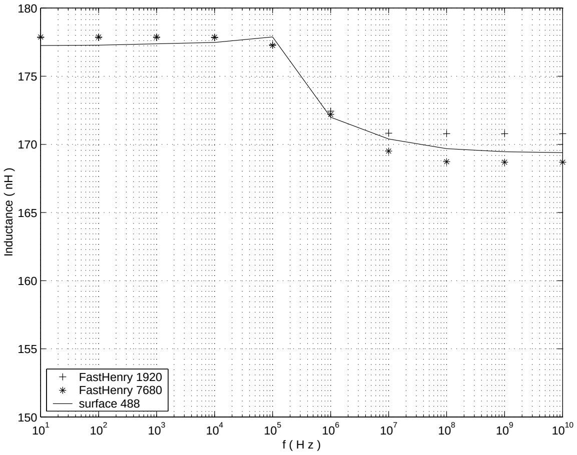

Figure 3-8: Inductance of a spiral

Chapter 4

Pre-corrected FFT algorithm

After discretization, the algebraic equations (2.25), (2.26), (2.27), (2.38) and (2.29) become sparse matrix equations. But integral equations (2.22), (2.23), (2.24) and (2.36) become dense matrix equations. So solving the whole system matrix using iterative methods still takes $O(N^2)$ operations, where N is the number of unknowns. In this thesis, we use the pre-corrected FFT algorithm to accelarate the dense matrix vector product corrresponding to the operation of those integral operators in (2.22), (2.23), (2.24) and (2.36).

Even though numerous fast algorithms already exist for efficiently solving the integral equations, such as Fast Multipole Method (FMM) [8, 7, 36, 37], hirarchical SVD [20], panel clustering method [10] and the pre-corrected FFT (pFFT) algorithm [32], the practical implementation of such methods may still seem daunting to researchers and engineers, who are most oftern not specialists in fast integral equation solvers. As a result many existing codes still use the traditional dense matrix approaches, which need $O(N^2)$ memory and at least $O(N^2)$ CPU time. One of the objects of this work is to provide a flexible and extensible code to the public domain so that the researchers can easily accelarate their codes. Hence we want to use an algorithm that is flexible enough to handle the integral kernels commonly used in the above mentioned engineering applications.

Though not as good as FMM's more than ten digit accuracy, pFFT's four to five digit accuracy is good enough for most engineering applications, where the accuracy requirement is usually modest. More importantly, the pFFT method is almost kernel-independent. For example, it could easily handle both Helmholtz kernel and Laplace kernel and their close relatives in a unified framework. This makes it a particularly good algorithm for our fast solver.

On the other hand, FMM could not handle Helmholtz kernel and Laplace kernel in a unified framework. And it has numerical difficulty in dealing with large variation of the wave number in Helmholtz kernel [36, 37].

4.1 Mathematical Preliminaries

An abstract form of the kernels in (2.22), (2.23), (2.24) and (2.36) is

$$K(\vec{r'},\vec{r}) = \mathcal{F}_1(\mathcal{F}_2(G(\vec{r'},\vec{r}))) \tag{4.1}$$

where $G(\vec{r'},\vec{r})$ is the Green's function, and the possible options for operator $\mathcal{F}_1(\cdot)$ and $\mathcal{F}_2(\cdot)$ are

$$\mathcal{F}_1(\cdot) = U(\cdot), \frac{d(\cdot)}{dx(\vec{r})}, \frac{d(\cdot)}{dy(\vec{r})}, \frac{d(\cdot)}{dz(\vec{r})}, \frac{d(\cdot)}{dn(\vec{r})}, \tag{4.2}$$

and

$$\mathcal{F}_2(\cdot) = U(\cdot), \frac{d(\cdot)}{dx(\vec{r'})}, \frac{d(\cdot)}{dy(\vec{r'})}, \frac{d(\cdot)}{dz(\vec{r'})}, \frac{d(\cdot)}{dn(\vec{r'})}, \tag{4.3}$$

and $U(\cdot)$ is the identity operator.

For the sake of clarity, we use a simple single-kernel integral equation

$$\int_S dS' K(\vec{r'},\vec{r})\rho(\vec{r'}) = f(\vec{r}), \ \ \vec{r} \in S \tag{4.4}$$

to illustrate how the pFFT algorithm can be used to accelarate the operation of an integral operator. Function $f(\vec{r})$ is the known right hand side term. The procedure extends easily to the integral equations with multiple kernels, such as (2.22), (2.23), and (2.36).

The standard procedure to solve equation (4.4) numerically is to discretize it by means of projection [10] and solve the resultant linear system with an iterative method [39, 45], such as GMRES [40]. Let X be the infinite-dimensional functional space in which the exact solution of equation (4.4) lies, and assume that $B_n \subset X$ and $T_n \subset X$ are its subspaces with spans $\{b_j(\vec{r}), j = 1,2,...,n\}$ and $\{t_i(\vec{r}), i = 1,2,...,n\}$, where n is the dimension of both subspaces. In general, the solution of the equation (4.4) is not in subspace B_n. Therefore, the approximate solution

$$\rho_n(\vec{r}) = \sum_{j=1}^{n} \alpha_j b_j(\vec{r}) \in B_n \tag{4.5}$$

generates an error

$$e_n(\vec{r}) = \int_S dS' K(\vec{r'}, \vec{r}) \rho_n(\vec{r'}) - f(\vec{r}) = \phi(\vec{r}) - f(\vec{r}), \ \ \vec{r} \in S \tag{4.6}$$

and the unknown expansion coefficients α_i could be computed by enforcing the projection of the error into T_n to vanish, i.e.,

$$< t_i(\vec{r}), e_n(\vec{r}) >=< t_i(\vec{r}), \phi(\vec{r}) > - < t_i(\vec{r}), f(\vec{r}) >= 0, \ \ i = 1, 2, ..., n \tag{4.7}$$

or

$$\sum_{j=1}^{n} \alpha_j \int_{\Delta_i^t} dS t_i(\vec{r}) \int_{\Delta_j^b} dS' K(\vec{r'}, \vec{r}) b_j(\vec{r'}) = \int_{\Delta_i^t} dS t_i(\vec{r}) f(\vec{r}), \ \ i = 1, 2, ..., n, \tag{4.8}$$

where Δ_i^t and Δ_j^b are the support of the basis fucntion $t_i(\vec{r})$ and $b_j(\vec{r})$, respectively. In matrix, equation (4.8) becomes

$$[A]\bar{\alpha} = \bar{f} \tag{4.9}$$

where

$$A_{i,j} = \int_{\Delta_i^t} dS t_i(\vec{r}) \int_{\Delta_j^b} dS' K(\vec{r'}, \vec{r}) b_j(\vec{r'}) \tag{4.10}$$

The commonly used basis functions in B_n or T_n are low-order polynomials with local support [10]. Figure 4-1 shows a piece-wise constant basis function whose support is a panel. Figure 4-2 shows a vertex-based piece-wise linear basis function whose support is the union of a cluster of panels sharing the vertex with which the basis function is associated.

When the ith testing function is $t_i(\vec{r}) = \delta(\vec{r} - \vec{r_{c,i}})$, where $\vec{r}_{c,i}$ is the collocation point, the discretization method is called the collocation method. And when $B_n = T_n$, the discretization method is called the Galerkin's method.

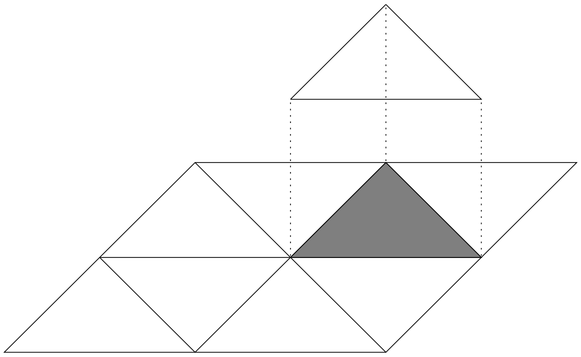

Figure 4-1: A piece-wise constant basis function, shaded area is its support

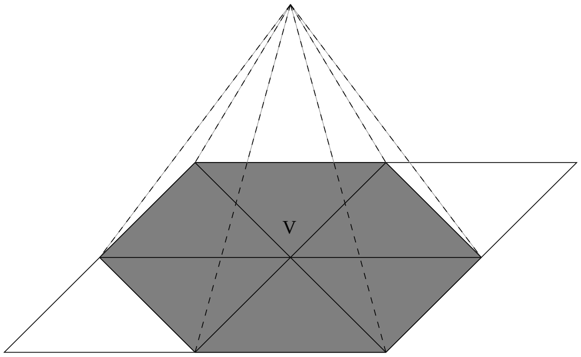

Figure 4-2: A piece-wise linear basis function associated with the vertex V, where the shaded area is its support

4.2 Philosophical Preliminaries

Since forming matrix A and computing the matrix vector product in (4.9) all require $O(N^2)$ arithmetic operations, it is obvious that using an iterative method to solve equation (4.9) needs at least $O(N^2)$ time, where N is the size of the matrix A. This could be very expensive for large N. Many fast algorithms avoid forming matrix A explicitly and compute the matrix vector product approximately, which only needs $O(N)$ or $O(Nlog(N))$ operations [7, 1, 33].

The Pre-corrected FFT (pFFT) algorithm was originally proposed in [32, 33], where the detailed steps to accelarate a single-layer integral operator were shown. The basic idea of pFFT is to separate the potential computation into far-field part and near-field part. The far-field potential is computed by using the grid charges on a uniform 3D grid to represent charges on the panels. The near-field potential is compued directly. The algorithm has four steps: Projection, Convolution, Interpolation and Nearby interaction. The effect of this algorithm is to replace the matrix vector product $A\bar{\alpha}$ in equation (4.9) with $(D+IHP)\bar{\alpha}$, where D is the direct matrix that represents the nearby interaction, I is the interpolation matrix, H is the convolution matrix, and P is the projection matrix. Matrices D, I and P are sparse, hence their memory usage is $O(N_p)$, where N_p is the number of panels, and their product with a vector needs only $O(N_p)$ work. The matrix H is a multilevel Toeplitz matrix. Hence its memory usage is $O(N_g)$ and its product with a vector could be computed by using FFT in $O(N_g log(N_g))$ operations [6], where N_g is the number of grid points. Therefore, the overall computational complexity of $(D+IHP)\bar{\alpha}$ is $O(N_p)+O(N_g log(N_g))$. For some problems, usually small or medium sized ones, N_g might be larger. Hence the computational complexity is $O(N_g log(N_g))$. For other problems, usually large-sized ones, the computational complexity is nearly $O(N_p)$.

Unlike [32, 33], we use polynomials in both interpolation and projection steps. Hence the interpolation matrix I and projection matrix P are completely independent of the Green's function $G(\vec{r},\vec{r'})$ in equation (4.1). This makes it much easier to handle the complicated kernels $K(\vec{r'},\vec{r})$ in (4.1). It also makes it straight forward to treat piecewise constant basis and high-order basis in either collocation or Galerkin's method in a unified framework. This is particularly important from implementation point of view.

4.3 Pre-corrected FFT algorithm

In this section, we will use a simple 2D example to show how to generate the four matrices, $[I]$, $[P]$, $[H]$ and $[D]$. Generalization of the procedure to the 3D cases is straight forward. The algorithm presented here is general enough such that the general integral operator in equation (4.4) discretized either by the collocation method or by the Galerkin's method using either piece-wise const element or high-order element could be handled in a unified framework.

4.3.1 Interpolation matrix

We start with the interpolation, the third and easiest step in the four-step pFFT algorithm.

Suppose the potential on the uniform grids has been computed through the first two steps, namely projection and convolution, we could use a simple polynomial interpolation scheme to compute the potential at any point within the region covered by the grids. Figure 4-3 shows a 2D 3×3 uniform grid, more points could be used to get more accurate results. The triangle inside the grid represents the local support Δ_i^t in equation (4.8). The simplest set of polynomial functions for the interpolation is $f_k(x,y) = x^i y^j, i, j = 0, 1, 2, k = 2i + j$. The potential at any point can be written as a linear combination of these polynomials,

$$\phi(x,y) = \sum_k c_k f_k(x,y) = \overline{f}^t(x,y)\overline{c} \tag{4.11}$$

where $\overline{c}$ is a column vector and t stands for transpose. Matching $\phi(x,y)$ in (4.11) with the given potential at each grid point results in a set of linear equations. In matrix form, it is

$$[F]\overline{c} = \overline{\phi}_g \tag{4.12}$$

where the j-th row of the matrix $[F]$ is the set of polynomials $\overline{f}(x,y)$ evaluated at the jth grid point (x_j, y_j), and $\phi_{g,j}$ is the given potential at point (x_j, y_j). Solving for $\overline{c}$ and substituting it back into (4.11) yields

$$\phi(\vec{r}) = \phi(x,y) = \overline{f^t}(x,y)[F]^{-1}\overline{\phi_g} = \overline{D^t_0}(\overline{r})\overline{\phi_g} \tag{4.13}$$

It should be noted that matrix $[F]$ in (4.12) is only related to the distance between points in the uniform grid and the specific set of interpolation polynomials chosen in the algorithm. So the inverse of matrix $[F]$ is done only once. And since the size of the matrix is rather small (9×9 in this simple 2D case), computing its inverse is inexpensive. It is possible that the number of polynomials is not equal to the number of points in the interpolation grid. In this case the inverse becomes psuedo inverse, which is computed using the singular value decomposition (SVD) [45].

It easily follows that the derivative of the potential at a point $\overline{r}$ with respect to α is

$$\frac{d\phi(\overline{r})}{d\alpha} = \frac{d}{d\alpha}\overline{f^t}(\overline{r})[F]^{-1}\overline{\phi_g} = \overline{D^t_\alpha}(\overline{r})\overline{\phi_g} \tag{4.14}$$

where α stands for x or y. Hence the gradient of the potential at $\overline{r}$ is

$$\nabla\phi(\overline{r}) = (\hat{x}\overline{D^t_x}(\overline{r}) + \hat{y}\overline{D^t_y}(\overline{r}))\overline{\phi_g} \tag{4.15}$$

and the normal derivative of the potential at point $\overline{r}$ is

$$\frac{d\phi(\overline{r})}{dn} = \hat{n}\cdot\nabla\phi(\overline{r}) = (n_x\frac{d\overline{f^t}(\overline{r})}{dx} + n_y\frac{d\overline{f^t}(\overline{r})}{dy})[F]^{-1}\overline{\phi_g} = \overline{D^t_n}(\overline{r})\overline{\phi_g} \tag{4.16}$$

where n_x and n_y are the projection of the unit normal vector of the function support Δ^t_i along x and y direction. Using the notation in (4.2), equations (4.13), (4.14) and (4.16) could be written as

$$\mathcal{F}_1(\phi(\vec{r})) = \overline{D^t_\beta}(\overline{r})\overline{\phi_g} \tag{4.17}$$

where $\overline{D^t_\beta}(\overline{r})$ stands for $\overline{D^t_0}(\overline{r})$, $\overline{D^t_x}(\overline{r})$, $\overline{D^t_y}(\overline{r})$ or $\overline{D^t_n}(\overline{r})$.

As described in section 4.1, we want to compute

$$\Psi_i = \int_{\Delta^t_i} dS\mathcal{F}_1(\phi(\vec{r}))t_i(\vec{r}), \;\; i = 1,2,..,N_t. \tag{4.18}$$

where N_t is the number of testing basis functions. Substituting (4.17) into (4.18) yields

$$\Psi_i = \int_{\Delta_i^t} dSt_i(\vec{r})\bar{D}_\beta^t(\vec{r})\bar{\phi}_g = (\bar{W}_\beta^{(i)})^t\bar{\phi}_g, \;\; i = 1,2,..,N_t, \tag{4.19}$$

where $\bar{W}_\beta^{(i)}$ stands for $\bar{W}_0^{(i)}$, $\bar{W}_x^{(i)}$ and $\bar{W}_y^{(i)}$. If the collocation method is used, then $\bar{W}_\beta^{(i)}$ in equation (4.19) could be simplified as

$$\bar{W}_\beta^{(i)} = \bar{D}_\beta(x_c, y_c), \;\; i = 1,2,..,N_t, \tag{4.20}$$

where (x_c, y_c) is the collocation point. When the piece-wise constant testing function is used, the support Δ_i^t is the panel associated with it, as shown in figure 4-1. When the linear testing function is used, Δ_i^t is a cluster of panels, as shown in figure 4-2. Apparently, computing elements of $\bar{W}_\beta^{(i)}$ for higher order basis functions could be more expensive because integrating over a cluster of panels needs more quadrature points than integrating over a single panel.

In matrix format, equation (4.19) becomes

$$\bar{\Psi} = [I]\bar{\phi}_g \tag{4.21}$$

where $[I]$ is an $N_t \times N_g$ matrix, and N_g is the number of grid points. To cover the local support of a basis function, only a small number of the interpolation grid points are needed, as shown in figure 4-3. Hence computing each Ψ_i through interpolation only involves potential at a few grid points. So each row of the interoplation matrix $[I]$ is rather sparse. The non-zero elements in the i-th row of the matrix $[I]$ are just the elements of the row vector $(\bar{W}_\beta^{(i)})^t$ in (4.19) or (4.20).

4.3.2 Projection matrix

Figure 4-4 shows a 2D pictorial representation of the projection step. Similar to the previous section, a triangle is used to represent the support of a basis function. A 3 × 3 projection grid is assumed here and obviously more points could be used if the accuracy requirement

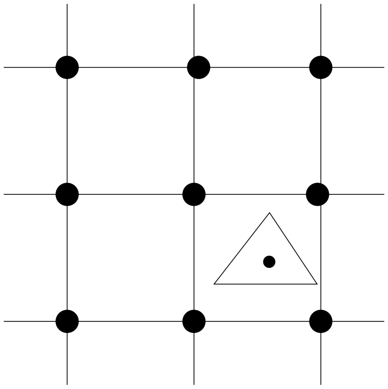

Figure 4-3: 2-D pictorial representation of the interpolation step

is higher.

We start with a point charge ρ_p at point **S** on the triangle, shown in figure 4-4. The potential at point **E** due to this point charge is

$$\phi_E^{(1)} = \rho_p G(\vec{r}_s, \vec{r}_E). \tag{4.22}$$

The purpose of the projection is to find a set of grid charges $\overline{\rho}_g$ on the projection grid points such that they generate the same potentail at point **E**, i.e.,

$$\phi_E^{(2)} = \sum_i \rho_{g,i} G(\vec{r}_i, \vec{r}_E) = (\overline{\rho}_g)^t \overline{\phi}_g = \phi_E^{(1)} \tag{4.23}$$

where $\phi_{g,i} = G(\vec{r}_i, \vec{r}_E)$. We could use the same set of polynomials in (4.11) to expand the Green's function

$$G(\vec{r}, \vec{r}_E) = \sum_k f_k(\vec{r}) c_k = \overline{f}^t(\vec{r}) \overline{c}. \tag{4.24}$$

Matching both sides at each grid point $\vec{r}_i$ yields a linear system

$$[F]\overline{c} = \overline{\phi}_g, \tag{4.25}$$

where F is same as that in (4.12). Substituting the solution $\overline{c} = F^{-1}\overline{\phi}_g$ into (4.24) and evaluating it at point **S** yields

$$G(\vec{r}_s, \vec{r}_E) = \overline{f}^t(\vec{r}_s) F^{-1} \overline{\phi}_g. \tag{4.26}$$

In light of (4.22) and (4.23) we have

$$(\overline{\rho}_g)^t = \rho_p \overline{f}^t(\vec{r}_s) F^{-1}, \tag{4.27}$$

the projection charges for a point charge. A charge distribution $b_j(\vec{r})$ on the jth basis function support could be regarded as a linear combination of an infinite number of point charges. Equation (4.27) implies that the projection charges are linearly proportional to the point charge, hence it easily follows that the projection charges for the charge distribution $b_j(\vec{r})$ is

$$(\overline{\rho}_g^{(j)})^t = [\int_{\Delta_j^b} dS b_j(\vec{r}) \overline{f}^t(\vec{r})][F]^{-1}. \tag{4.28}$$

If the piece-wise constant basis function is used, equation (4.28) becomes

$$(\overline{\rho}_g^{(j)})^t = [\int_{\Delta_j^b} dS \overline{f}^t(\vec{r})][F]^{-1}. \tag{4.29}$$

We usually have to use more than one basis function, as implied by equation (4.5). In this case, the total charge on each grid point is the accumulation of grid charge due to each basis function. Assuming there are N_b basis functions and N_g grid points, the relation between the total grid charges $\overline{Q}_g$ and the magnitude of basis functions $\overline{\alpha}$ in (4.5) is

$$\overline{Q}_g = \sum_{j=1}^{N_b} \alpha_j \overline{\rho}_g^{(j)} = [P]\overline{\alpha} \tag{4.30}$$

where $[P]$ is an $N_g \times N_b$ matrix. Due to the locality of the basis support, the projection grid for each basis function has only a small number of points. Hence each column of the projection matrix $[P]$ is rather sparse. The non-zero elements in the j-th column of matrix $[P]$ are the elements of the column vector $\overline{\rho}_g^{(j)}$ in equation (4.28) or (4.29).

If the kernel has a differential operator inside the integral, the potential at point **E** due to a point charge is

$$\phi_E^{(1)} = \frac{\partial}{\partial \beta(\vec{r}_s)}[\rho_p G(\vec{r}_s, \vec{r}_E)] = \frac{\partial}{\partial \beta(\vec{r}_s)}[\rho_p \overline{f}^t(\vec{r}_s) F^{-1} \overline{\phi}_g]. \tag{4.31}$$

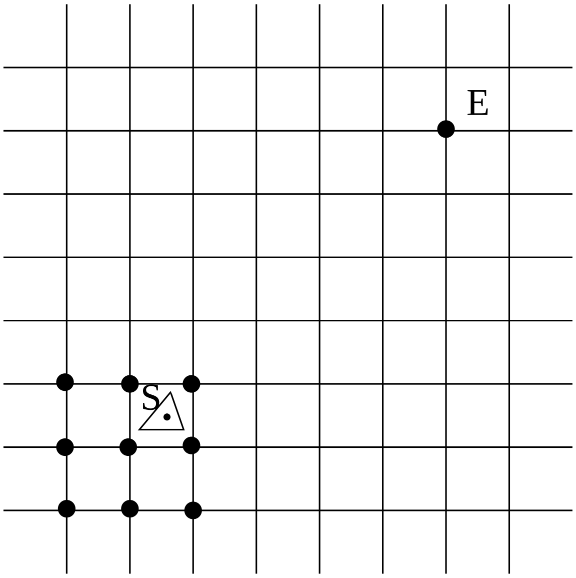

Figure 4-4: 2-D pictorial representation of the projection step

where β stands for x, y or n. We again want to find a set of grid charges $\bar{\sigma}_\beta$ on the projection grid points such that they generate the same potentail at point **E**, i.e.,

$$\phi_E^{(2)} = \sum_i \sigma_{\beta,i} G(\vec{r}_i, \vec{r}_E) = (\bar{\sigma}_\beta)^t \bar{\phi}_g = \phi_E^{(1)} \tag{4.32}$$

Equations (4.31) and (4.32) imply that the projection charges are

$$(\bar{\sigma}_\beta)^t = \frac{\partial}{\partial\beta(\vec{r}_s)}[\rho_p \bar{f}^t(\vec{r}_s) F^{-1}]. \tag{4.33}$$

Similar to the single-layer operator case, the projection charges for a charge distribution $b_j(\vec{r})$ on the jth basis function support is

$$\left(\bar{\sigma}_\beta^{(j)}\right)^t = [\int_{\Delta_j^b} dS b_j(\vec{r}) \frac{\partial}{\partial\beta(\vec{r})} \bar{f}^t(\vec{r})][F]^{-1}. \tag{4.34}$$

The projection matrix for the kernel with a differential operator is structurely identical to the matrix $[P]$ in equation (4.30). The non-zero elements in the j-th column of the matrix are the elements of the column vector $\bar{\sigma}_\beta^{(j)}$ in equation (4.34).

4.3.3 Convolution matrix and fast convolution by FFT

By definition, the relation between the grid potential $\overline{\phi_g}$ in (4.21) and grid charge $\overline{Q_g}$ in (4.30) is

$$\phi_{g,j} = \sum_i G(\vec{r'}_i, \vec{r}_j) Q_{g,i} \tag{4.35}$$

In matrix form, it is

$$\overline{\phi_g} = [H]\overline{Q_g} \tag{4.36}$$

where the matrix H is the so-call convolution matrix. Since the Green's function is position invariant and $\overline{\phi_g}$ and $\overline{Q_g}$ are defined on the same set of uniform grid, we have

$$H_{i,j} = G(\vec{r'}_i, \vec{r}_j) = G(\vec{r}_i, \vec{r}_j) = G(\vec{r}_i - \vec{r}_j, 0). \tag{4.37}$$

Matrix H is a multilevel Toeplitz matrix [6]. The number of levels is 2 and 3 for 2D cases and 3D cases, respectively. It is well-known that the storage of a Toeplitz matrix only needs $O(N)$ memory and a Toeplitz matrix vector product can be computed in $O(Nlog(N))$ operations using FFT [6], where N is the total number of grid points. It should be pointed out that convolution matrix H being a Toeplitz matrix is hinged upon the position invariance of the Green's function. Fortunately most commonly used Green's functions, even the ones in the layered medium, are position invariant [33].

4.3.4 Direct matrix and pre-correction

Substituting equation (4.36) and (4.30) into (4.21) yields

$$\overline{\Psi} = [I][H][P]\overline{\alpha} \tag{4.38}$$

In view of (4.18), (4.7) and (4.9), this implies

$$A = [I][H][P]. \tag{4.39}$$

As pointed out in previous three sections, the sparse representation of matrix A in (4.39) reduces the memory usage and computing time for matrix vector product dramatically. Unfortunately, the calculations of the potential on th grid using (4.39) do not accurately approximate the nearby interaction. It is proposed in [33] that the nearby interaction should be computed directly and the inaccurate contributions from the use of grid should be removed. Figure 4-5 shows how the nearby neighboring basis supports are defined. The empty circle in middle of the solid dots are the center of the so-called direct stencil and the stencil size in figure 4-5 is 2. The shaded triangle represents the source, and the other empty triangles represent the targets where Ψ in equation (4.18) is to be evaluated. Only those triangles within the region covered by the direct stencil are considered to be nearby neighbors to the source. And the direct interaction between this list of nearby neighbors and the source is just $A_{i,j}$ defined in (4.10), where i is the index of the shaded triangle representing the source and $j \in \mathcal{N}_i$, the nearby neighbor set for the ith source. The pre-corrected direct matrix element is

$$D_{i,j} = A_{i,j} - (\overline{W}_{\beta}^{(i)})^t [H_L] \overline{\rho}_g^{(j)}, \quad j \in \mathcal{N}_i \tag{4.40}$$

where $(\overline{W}_{\beta}^{(i)})^t$ is defined in equation (4.19), $\overline{\rho}_g^{(j)}$ is defined in equation (4.28) and (4.34), and $[H_L]$ is a small convolution matrix (not to be confused with $[H]$ in (4.39)) that relates the potential on the grid points around basis support Δ_i^t and the charge on the grid points around basis support Δ_j^b. It is intuitive from figure 4-5 that $\mathcal{N}_i$ is a very small set. Hence the direct matrix D is very sparse and the sparsity of D is dependent upon the size of the direct stencil. Larger stencil size means more neighboring triangles in figure 4-5 and hence more computation in (4.40). It will be shown later in section 5.1 that the setup time of the pFFT algorithm is directly related to the direct stencil size.

Since matrix $[H_L]$ in (4.40) is rather small, the FFT does not speed up the computation much. However, there are other ways to reduce the operation count. Because the grid is uniform and the Green's function is position invariant, only a few matrices $[H_L]$ are unique. So we could pre-compute them once and use them to pre-correct all the nearby interactions in the direct matrix $[D]$.

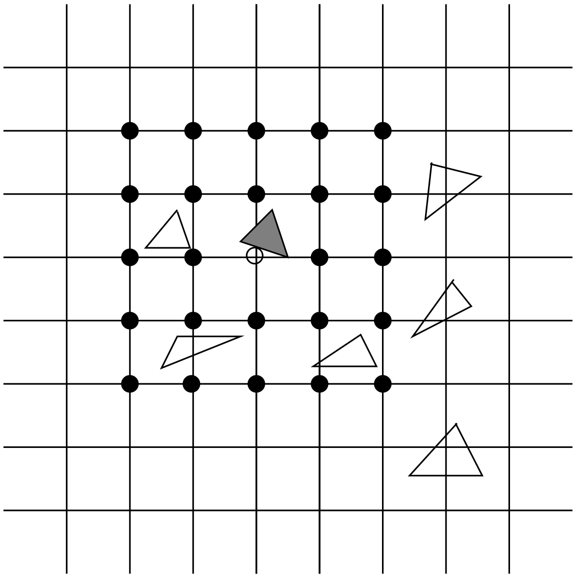

Figure 4-5: 2-D pictorial representation of the nearby interaction. Direct stencil size is 2.

Table 4.1: Relation between operator pair and the interpolation matrix and the projection matrix

$\mathcal{F}_1$	$U(\cdot)$	$\frac{d(\cdot)}{dx'}$, $\frac{d(\cdot)}{dy'}$	$\frac{d(\cdot)}{dn'}$
interpolation	$\overline{W}_0^{(i)}$ in (4.19)	$\overline{W}_x^{(i)}$, $\overline{W}_y^{(i)}$ in (4.19)	$\overline{W}_n^{(i)}$ in (4.19)
$\mathcal{F}_2$	$U(\cdot)$	$\frac{d(\cdot)}{dx}$, $\frac{d(\cdot)}{dy}$	$\frac{d(\cdot)}{dn}$
projection	$\overline{\rho}_g^{(j)}$ in (4.28)	$\overline{\sigma}_x^{(j)}$, $\overline{\sigma}_y^{(j)}$ in (4.34)	$\overline{\sigma}_n^{(j)}$ in (4.34)

4.3.5 A summary of the four matrices

In view of (4.38), (4.39) and (4.40), the matrix vector product is computed efficiently using

$$[A]\overline{\alpha} = ([D] + [I][H][P])\overline{\alpha}. \tag{4.41}$$

Sections 4.3.1 and 4.3.2 are summarized in table 4.1. It is clear by now that the interpolation matrix $[I]$ and the projection matrix $[P]$ are independent of the Green's function. Matrix $[I]$ is only related to the operator $\mathcal{F}_1$ and the testing functions. And matrix $[P]$ is only related to the operator $\mathcal{F}_2$ and the basis functions.

The direct matrix, however, is dependent upon all the above information. So we have to set up one direct matrix for each $\mathcal{F}_1$ and $\mathcal{F}_2$ operator pair. The convolution matrix, on the other hand, is only related to the Green's function and the location of grid points. It is not related to $\mathcal{F}_1$ or $\mathcal{F}_2$. So we only need to set up one convolution matrix for each unique Green's function.

In addition, if the Galerkin's method is used, the basis function $b_j(\vec{r})$ in equation (4.28) or (4.34) is identical to the testing function $t_i(\vec{r})$ in equation (4.19). It is easy to check that $\overline{W}_0^{(i)} = \overline{\rho}_g^{(j)}$, $\overline{W}_x^{(i)} = \overline{\sigma}_x^{(j)}$, $\overline{W}_y^{(i)} = \overline{\sigma}_y^{(j)}$ and $\overline{W}_n^{(i)} = \overline{\sigma}_n^{(j)}$. This implies a duality relation

$$[I] = [P]^t. \tag{4.42}$$

4.4 Implementation

Base upon the algorithm described above, we have developed a C++ program called pfft++, using the generic programming technique [43, 21, 16]. The whole algorithm includes two major parts: forming the four matrices I, P, D and H, and computing the matrix vector product using (4.41). Since the matrices I and P are not related to the kernel, they are formed separately so that they could be used for different kernels. This is particularly useful when for example a Helmholtz equation is to be solved at various wave numbers or frequencies. The following is a high level description of the implementation of the pfft++.

```
Algorithm 1: construct kernel Independent sparse matrices.
Input: source elements, target elements, differential operator pairs ($\mathcal{F}_1$, $\mathcal{F}_2$),
projection stencil size, interpolation stencil size, direct stencil size
Output: interpolation matrix $[I]$ and projection matrix $[P]$
(1)    find the optimal grid size
(2)    setup grid and element association
(3)    setup interpolation stencil
(4)    setup projection stencil
(5)    setup direct stencil
(6)    form the interpolation matrix $[I]$ for each $\mathcal{F}_1$
(7)    form the projection matrix $[P]$ for each $\mathcal{F}_2$
```

Using pfft++ to solve a single kernel integral equation such as (4.4) is straight forward. We could simply treat pfft++ as a black box that could perform the matrix vector product efficiently. After forming the four matrices by calling algorithms 1 and 2, algorithm 3 is

```
Algorithm 2: construct kernel dependent sparse matrices.
Input: source elements, target elements, kernel, integration scheme, differential operator pairs ($\mathcal{F}_1$, $\mathcal{F}_2$)
Output: direct matrix $[D]$ and convolution matrix $H$
(1)    form the sparse representation of $[H]$
(2)    compute the FFT of $[H]$
(3)    form the direct matrix $[D]$ for each pair of ($\mathcal{F}_1$, $\mathcal{F}_2$)
```

```
Algorithm 3: compute matrix vector product.
Input: vector x, differential operator pair ($\mathcal{F}_1$, $\mathcal{F}_2$)
Output: vector y
(1)    find the index $m$ of $[I]$ from $\mathcal{F}_1$
(2)    find the index $n$ of $[P]$ from $\mathcal{F}_2$
(3)    find the index $k$ of $[D]$ from operator pair ($\mathcal{F}_1$, $\mathcal{F}_2$)
(4)    $y_1 = [P_m]x$
(5)    $y_1 = fft(y_1)$
(6)    $y_2 = [H]y_1$
(7)    $y_2 = ifft(y_2)$
(8)    $y_3 = [I_n]y_2$
(9)    $y = y_3 + [D_k]x$
```

to be called repeatedly in the inner loop of an iterative solver. To solve the integral equations with multiple kernels, we could simply repeat the above procedure for each integral operator individually.

4.5 Comparison to the original pFFT algorithm

The basic sparsification ideas in this paper are very similar to those in the original pre-corrected FFT algorithm [32]. The difference lies primarily in the ways the interpolation matrix and the projection matrix are generated. And this difference turns out to be important.

In the original pFFT algorithm [32, 33], the projection matrix and the interpolation matrix are all related to the Green's function or kernel. If one wants to solve a Helmholtz equation with different wave numbers or at different frequencies, these two matrices have to be re-generated for each frequency. As explained in section 4.4, the interpolation matrix and the projection matrix are only generated once in pfft++.

In the original pFFT algorithm, the convolution matrix is directly related to the kernel, which includes the effect of the operator $\mathcal{F}_2$. The convolution matrix in this work is directly related to the Green's function, not the operator $\mathcal{F}_2$. To see why this difference is important, suppose we want to compute the double-layer integral

$$\int_S d\vec{r'} \frac{\partial G(\vec{r}, \vec{r'})}{\partial n(\vec{r'})} \rho(\vec{r'}). \tag{4.43}$$

Using the original pFFT algorithm, it has to be done as the following

$$\int_S d\vec{r'} [n_x \frac{\partial G(\vec{r}, \vec{r'})}{\partial x(\vec{r'})} + n_y \frac{\partial G(\vec{r}, \vec{r'})}{\partial y(\vec{r'})} + n_z \frac{\partial G(\vec{r}, \vec{r'})}{\partial z(\vec{r'})}] \rho(\vec{r'}). \tag{4.44}$$

This suggests that three convolution matrices $[H_x]$, $[H_y]$ and $[H_z]$ corresponding to $\frac{\partial G}{\partial x}$, $\frac{\partial G}{\partial y}$ and $\frac{\partial G}{\partial z}$ have to be generated and foreward FFT has to be performed for each of them. For each operation of the double-layer integral operator, $[H_x]\overline{\rho}$, $[H_y]\overline{\rho}$ and $[H_z]\overline{\rho}$ have to be carried out separately. As shown in section 4.3.3, pfft++ only needs one convolution matrix and hence only one convolution will be carried out in the matrix vector product step. This is a significant reduction in memory usage and CPU time.

Chapter 5

Numerical Results

5.1 Performance of pfft++

The pfft++ has been tested on random distributions on the surface of a sphere shown in figure 5-1. After discretizing the surface, the integral operator in equation (4.4) is turned into either the dense matrix $[A]$ in (4.9) or the sparse matrix representation in (4.41).

We assume a random vector α and compute the matrix vector product in (4.9) directly as $y_1 = [A]\bar{\alpha}$. We then compute the matrix vector product using the pfft++ as $y_2 = pfft(\bar{\alpha})$. The relative error in the pFFT approximation is

$$error = (\frac{\sum_{i=1}^{N}(y_{1,i} - y_{2,i})^2}{\sum_{i=1}^{N} y_{1,i}^2})^{1/2} \tag{5.1}$$

For the largest simulations, with number of triangle panels N being 50000, we have carried out the direct calculation on a subset of only 100 panels. The CPU times are computed by extrapolation and the errors are obtained by restricting the formulae (5.1) to this subset.

To verify that the pfft++ works well for different kernels, we have carried out the simulations for Laplace kernel and its normal direvative, and Helmholtz kernel with different wave numbers and their normal direvative. The results of our experiments, relative error, CPU time and memory usage, are summarized in figures 5-2 - 5-17.

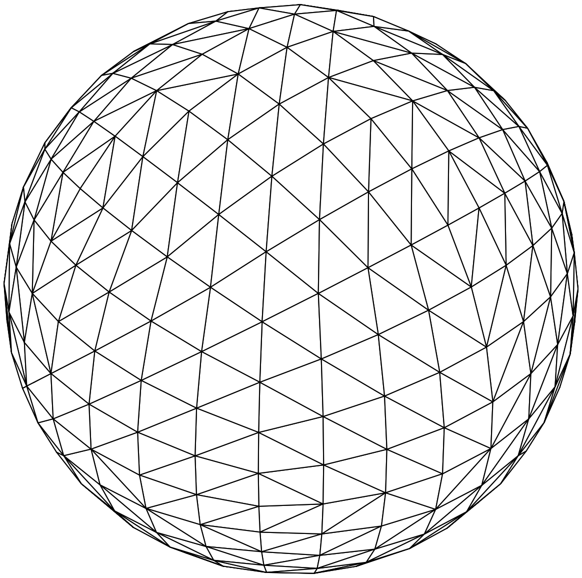

Figure 5-1: Triangulation on the surface of a sphere

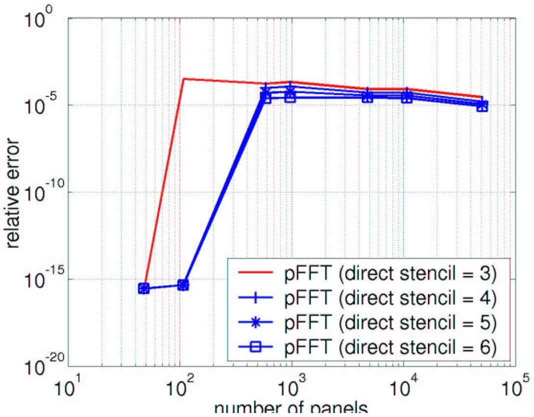

Figure 5-2: relative error in matrix vector product for $1/r$ kernel

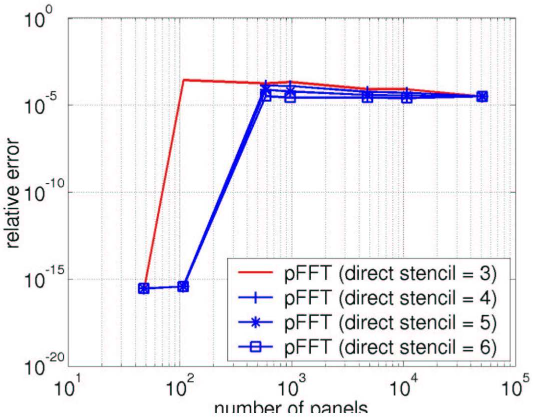

Figure 5-3: relative error in matrix vector product for e^{ikr}/r kernel, $k = 188849555.92$ *or* 0.1885

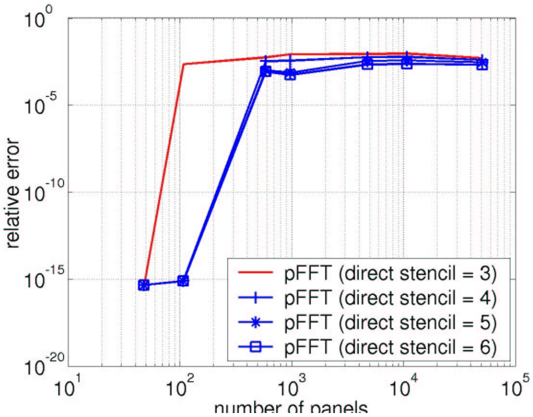

Figure 5-4: relative error in matrix vector product for $\frac{d}{dn}1/r$ kernel

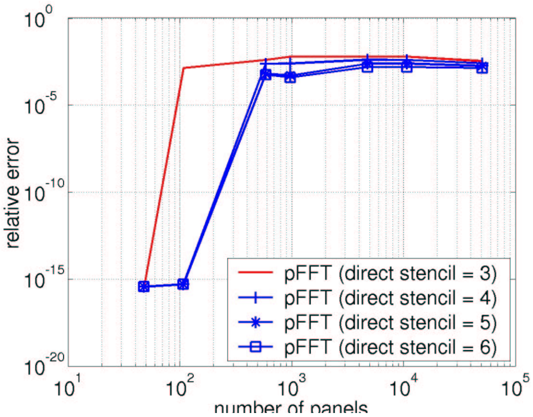

Figure 5-5: relative error in matrix vector product for $\frac{d}{dn}e^{ikr}/r$ kernel, $k = 0.1885$

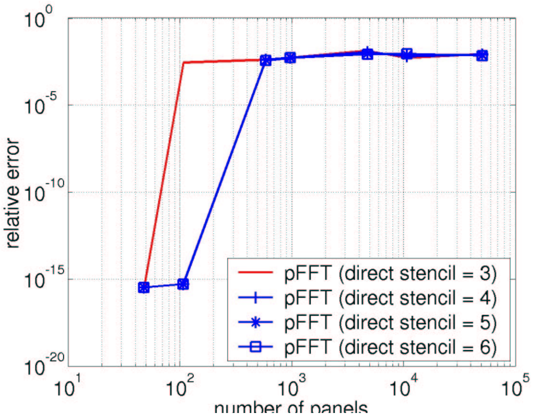

Figure 5-6: relative error in matrix vector product for $\frac{d}{dn}e^{ikr}/r$ kernel, $k = 188849555.92$

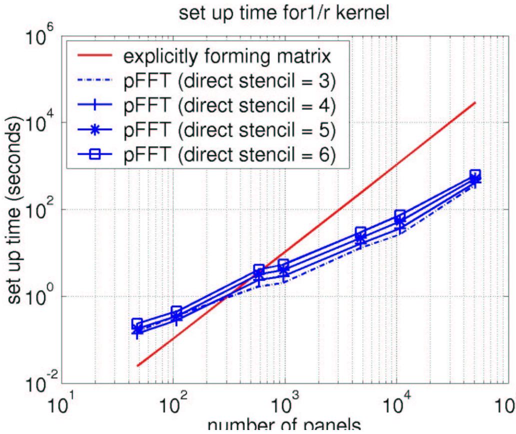

Figure 5-7: set up time for $1/r$ kernel

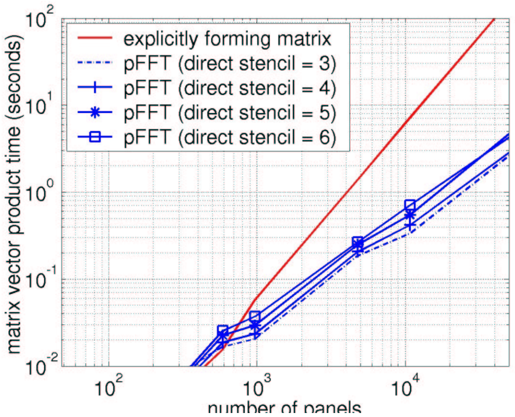

Figure 5-8: matrix vector product time for $1/r$ kernel

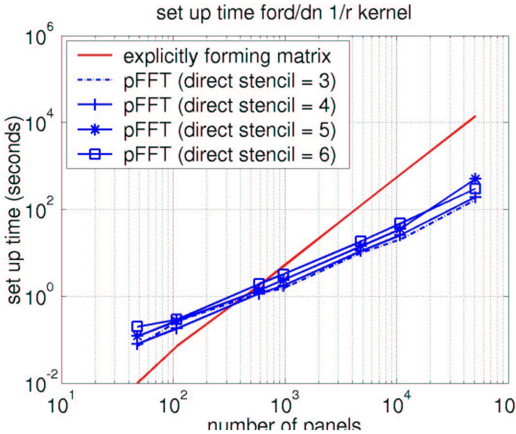

Figure 5-9: set up time for $\frac{d}{dn}1/r$ kernel

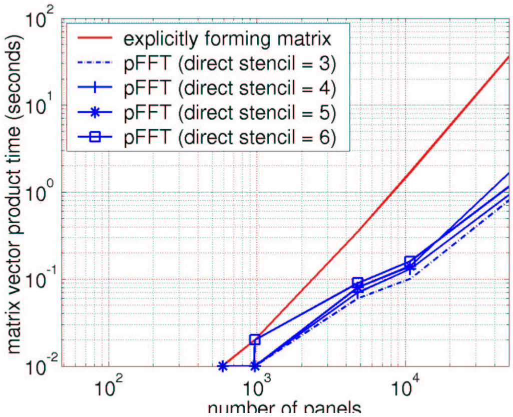

Figure 5-10: matrix vector product time for $\frac{d}{dn}1/r$ kernel

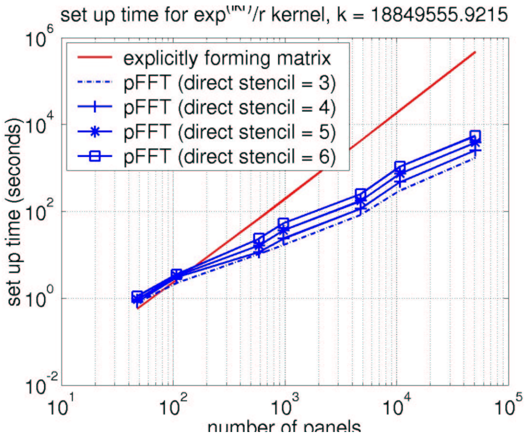

Figure 5-11: set up time for e^{ikr}/r kernel, $k = 188849555.92$ *or* 0.1885

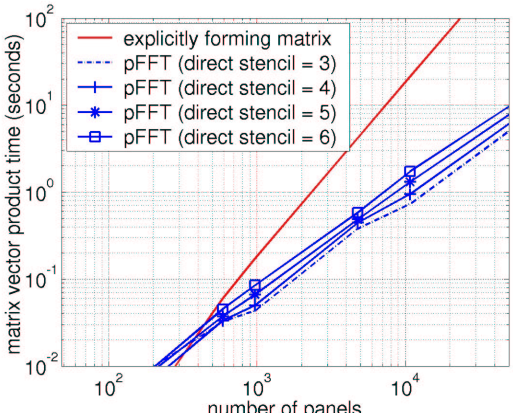

Figure 5-12: matrix vector product time for e^{ikr}/r kernel, $k = 188849555.92$ *or* 0.1885

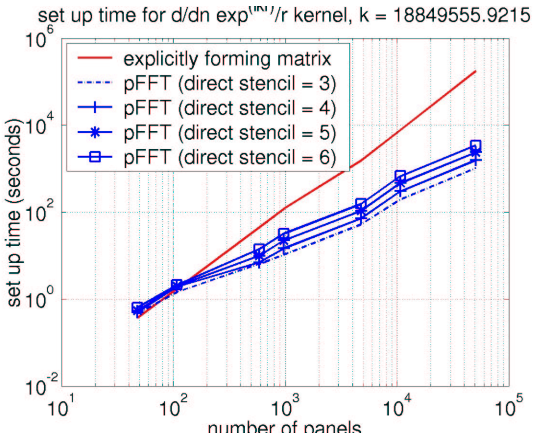

Figure 5-13: set up time for $\frac{d}{dn}e^{ikr}/r$ kernel, $k = 188849555.92$ *or* 0.1885

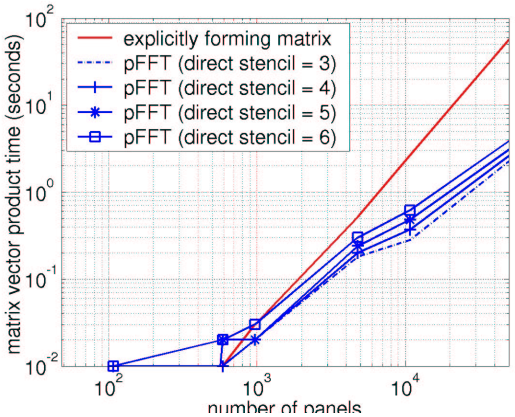

Figure 5-14: matrix vector product time for $\frac{d}{dn}e^{ikr}/r$ kernel, $k = 188849555.92$ *or* 0.1885

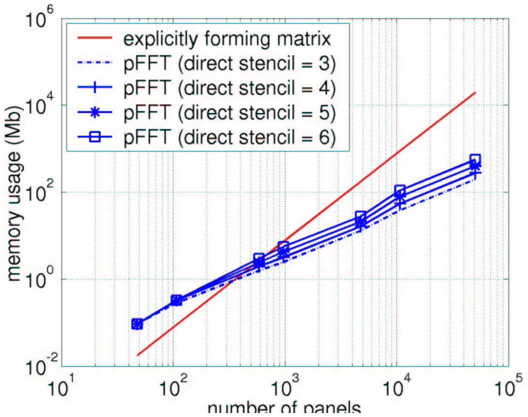

Figure 5-15: memory usage for $1/r$

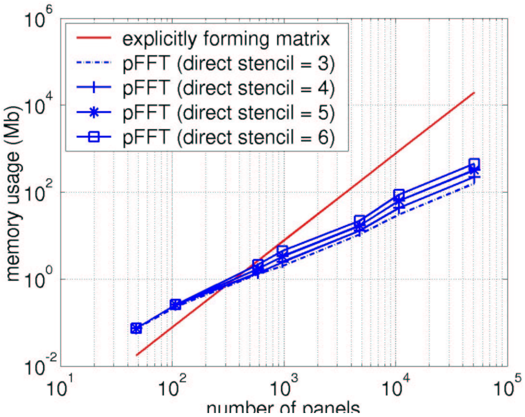

Figure 5-16: memory usage for $\frac{d}{dn}1/r$

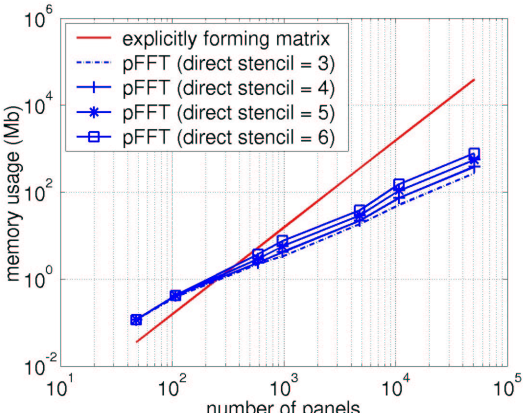

Figure 5-17: memory usage for e^{ikr}/r and $\frac{d}{dn}e^{ikr}/r$, $k = 188849555.92$ *or* 0.1885

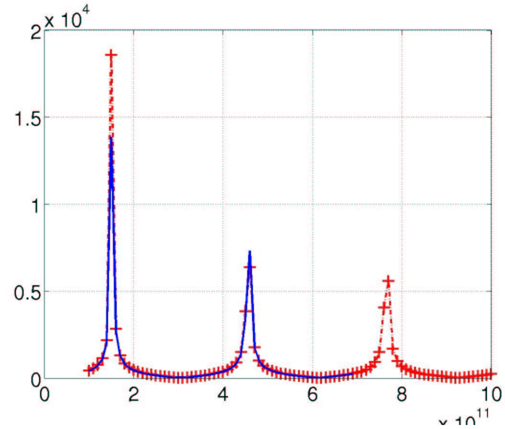

Figure 5-18: Shorted transmission line with and without substrate ground

5.2 Testing of fastImp

In this section, we first use a small example to demonstrate fastImp's accuracy. We then use a few large practical examples to demonstarte fastImp's speed and capacity.

5.2.1 Shorted transmission line with and without substrate ground

The behavior of a shorted transmission line is well understood. The expected resonance frequencies are clearly shown in the plot.

5.2.2 A four-turn spiral over ground

In total, we used 15162 panels to discretize the whole structure. For MQS analysis, the number of unknowns is 106k. The CPU time is 69 minutes and the memory usage is 348 Mb. For EMQS analysis, the number of unknowns is 121k. The CPU time is 93 minutes and memory usage is 379 Mb.

5.2.3 Multiple conductor crossover bus

There are three-layer of conductors, each layer has 10 conductors and the conductors on different layer are orthogonal to each other. In total, we used 12540 panels to discretize the whole structure. For MQS analysis, the number of unknowns is 87.5k. The CPU time is 41

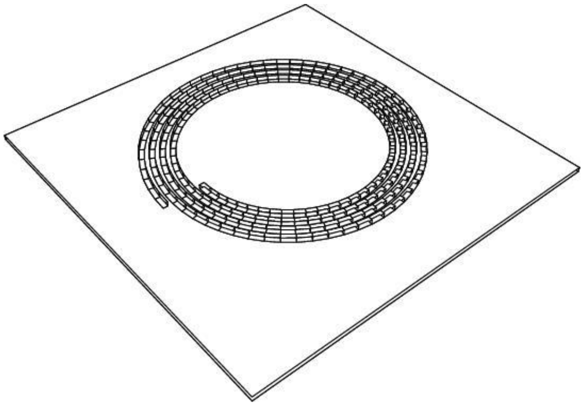

Figure 5-19: A four-turn spiral over ground

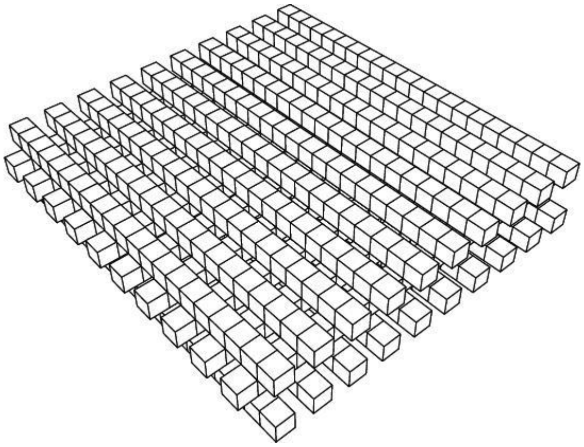

Figure 5-20: Multiple conductor bus

minutes and the memory usage is 165 Mb. For EMQS analysis, the number of unknowns is 100k. The CPU time is 61 minutes and memory usage is 218 Mb.

Chapter 6

Conclusions

We have derived a recently developed surface integral formulation from a different perspective. And we have shown the connections between this formulation and the classical EFIE formulation and MPIE formulation. These connections help us better understand why this formulation is widebanded. Using a piecewise quadrature scheme to improve the accuracy of panel integration, we have fixed the low-frequency problem in the original formulation. We have also generalized the pre-coorected FFT algorithm to the accelaration of complicated integral operators. Based on this generalization we have developed a flexible and extensible fast integral equation solver, pfft++. This solver could be applied to a wide range of problems. Using pfft++ as the engine, we have developed a fast impedance extraction program, fastImp. Numerical examples show that fastImp can perform MQS, EMQS and fullwave analysis of 3D general structures across wide frequency range, from zero frequency to at least hundreds of giga hertz. It only takes fastImp a few hours to solve problems with hundreds thousands of unknowns.

Bibliography

[1] A. Brandt and A. A. Lubrecht. Multilevel matrix multiplication and fast solution of integral equations. 90:348–370, 1990.

[2] Rainer Bunger and Fritz Arndt. Efficient MPIE approach for the analysis of three-dimensional microstrip structures in layered media. *IEEE Transactions on Microwave Theory and Techniques*, 45:1141–1753, August 1997.

[3] David C. Chang and Jian X. Zheng. Electromagnetic modeling of passive circuit elements in MMIC. *IEEE Transactions on Microwave Theory and Techniques*, 40:1741–1747, September 1992.

[4] W.C. Chew. *Waves and fields in inhomogeneous media*. IEEE Press, New Jersey, 1995.

[5] D. Colton and R. Kress. *Integral Equation Methods in Scattering Theory*. Krieger Publishing Company, Malabar, Florida, 1992.

[6] G. H. Golub and C. F. Van Loan. *Matrix Computation*. The Johns Hopkins University Press, second edition, 1989.

[7] L. Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems*. M.I.T. Press, Cambridge, Massachusetts, 1988.

[8] L. Greengard and V. Rohklin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73(2):325–348, December 1987.

[9] Frederick Warren Grover. *Inductance calculations, working formulas and tables*. Govt. Print. Off., New York, NY, 1946.

[10] Wolfgang Hackbush. *Integral Equations, Theory and Numerial Treatment*. Birkhauser Verlag, Basel, Switzerland, 1989.

[11] R. F. Harrington. *Field Computation by Moment Methods*. MacMillan, New York, 1968.

[12] H.A. Haus and J.R. Melcher. *Electromagnetic fields and energy*. Prentice-Hall, Englewood Cliffs, 1989.

[13] Hansruedi Heeb and Albert E. Ruehli. Three-dimensional interconnect analysis using partial element equivalent circuits. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 39(11):974–982, November 1992.

[14] J. L. Hess and A. M. O. Smith. Calculation of potential flow about arbitrary bodies. *Progress in Aeronautical Science*, pages 1–138, 1966.

[15] L.T. Huang and I. Turlik. A review of the skin effect as applied to thin film interconnections. *IEEE Transactions on Components, Packaging, and Manufacturing Technology*, 15:43–55, February 1992.

[16] N. M. Josuttis. *The C++ Standard Library : A Tutorial and Reference*.

[17] S. Kalaicheluan and J.D. Lavers. *BEM for eddy current problems*. Vol.6: Topics in BE Research, edited by C.A. Brebbia, London, 1989, pp. 79-116.

[18] M. Kamon, N.A. Marques, L.M. Silveria, and J.K. White. Automatic generation of accurate circuit models of 3D interconnect. *IEEE Transactions on Components, Packaging, and Manufacturing Technology–Part B*, 21(3):225–240, August 1998.

[19] M. Kamon, M. J. Tsuk, and J.K. White. FastHenry: A multipole-accelerated 3-D inductance extraction program. *IEEE Transactions on Microwave Theory and Techniques*, 42(9):1750–1758, September 1994.

[20] S. Kapur and J. Zhao. A fast method of moments solver for efficient parameter extraction of MCMs. 34^{th} *ACM/IEEE Design Automation Conference*, pages 141–146, 1997.

[21] Andrew Koenig and Barbara E. Moo. *Accelerated C++: Practical Programming by Example*.

[22] J.A. Kong. *Electromagnetic wave theory*. EMW Publishing, Cambridge, Massachusetts, 1999.

[23] A. Krawczyk. *Numerial modeling of eddy currents*. Clarendo Press, Oxford, 1993.

[24] Feng Ling, Dan Jiao, and Jian-Ming Jin. Efficient eletromagnetic modeling of microstrip structures in multilayer media. *IEEE Transactions on Microwave Theory and Techniques*, 47:1810–1818, September 1999.

[25] D. Luca, Alberto Sangiovanni-Vincentelli, and Jacob K. White. Using conduction modes basis functions for efficient electromagnetic analysis of on-chip and off-chip interconnect. *ACM/IEEE Design Automation Conference*, June 18-22, 2001.

[26] Yehia M. Massoud. *Simulation algorithms for inductive effects*. Ph.D. thesis MIT EECS Department, Cambridge, MA, 1999.

[27] J.R. Mautz and R.F. Harrington. An E-field solution for a conducting surface small or comparable to the wavelength. *IEEE Transactions on Antennas and Propagation*, 32:330–339, April 1984.

[28] Juan R. mosig. Arbitrarily shaped microstrip structures and their analysis with mixed potential integral equation. *IEEE Transactions on Microwave Theory and Techniques*, 36:314–323, February 1988.

[29] K. Nabors and J. White. FASTCAP: A multipole-accelerated 3-D capacitance extraction program. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 10:1447–1459, November 1991.

[30] T. Nakata, N. Takahashi, K. Fujifwara, K. Muramatsa, and Z.G. Cheng. Comparison of various methods for 3-D eddy current analysis. *IEEE Transactions on Magnetics*, 24:3159–3161, November 1988.

[31] A.F. Peterson. The interior resonance problem associated with surface integral equations of electromagnetics: Numerical consequences and a survey of remedies. *Electromagnetics*, pages 293–312, October 1990.

[32] J. R. Phillips and J. K. White. A precorrected-FFT method for electrostatic analysis of complicated 3D structures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pages 1059–1072, 1997.

[33] Joe R. Phillips. *Rapid solution of potential integral equations in complicated 3-dimensional geometries*. Ph.D. thesis MIT EECS Department, 1997.

[34] S. Ramo, J.R. Whinnery, and T.V. Duzer. *Fields and waves in communication electronics*. John Willey and sons, Inc., New York, 1994.

[35] S.M. Rao, D.R. Wilton, and A.W. Glisson. Electromagnetic scattering by surfaces of arbitrary shape. *IEEE Transactions on Antennas and Propagation*, 30:409–418, May 1982.

[36] V. Rokhlin. Rapid solution of integral equations of scattering theory in two dimensions. *J. Comp. Phys.*, 86:414–439, 1990.

[37] V. Rokhlin. Diagonal forms of translation operators for the Helmholtz equation in three dimensions. *Applied and Computational Harmonic Analysis*, 1:82–93, 1993.

[38] Wolfgang M. Rucker, Robert Hoschek, and Kurt R. Richter. Various BEM formulations for calculating eddy current in terms of field variables. *IEEE Trans. on Magnetics*, 31:1336–1341, May 1995.

[39] Y. Saad. *Iterative methods for sparse linear systems*. PWS Publishing, Boston, MA, 1996.

[40] Youcef Saad and Martin Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.*, 7(3):856–869, July 1986.

[41] J.M. Song, C.C. Liu, W.C. Chew, and S.W. Lee. Fast illinois solver code (FISC) solves problems of unprecedented size at center for computational electromagnetics, university of illinois. *IEEE Antennas and Propagation Magzine*, 40:27–34, June 1998.

[42] J.A. Stratton. *Electromagnetic Theory*. McGraw-Hill Book Company, New York, 1941.

[43] Bjarne Stroustrup. *The C++ Programming Language Special Edition*.

[44] C.T. Tai. *Dyadic green's functions in electromagnetic theory*. IEEE Press, Piscataway, New Jersey, 1994.

[45] L.N. Trefethen and D. Bau. *Numerical linear algebra*. SIAM, Philadelphia, 1997.

[46] J. Wang and J. K. White. A wide frequency range surface integral formualtion for 3D RLC extraction. *International Conference on Computer Aided-Design*, 1999.

[47] J.J.H. wang. *Generalized moment methods in electromagnetics*. John Willey and sons, Inc., New York, 1991.

[48] Junfeng Wang. *A new surface integral formulation of EMQS impedance extraction for 3-D structures*. Ph.D. thesis MIT EECS Department, 1999.

[49] X. Wang, P. Mucha, and J.K. White. Fast fluid analysis for multibody micromachined devices. *Proceedings of MSM*, pages 19–22, Hilton Head island, SC, 2001.

[50] Wen-Liang Wu, Allen W. Glisson, and Darko Kajfez. A study of two numerial solution precedures for the electric field integral equation at low frequency. *Applied computational electromagnetics*, 10:69–80, November 1995.

[51] J.S. Zhao and W.C. Chew. Integral equation solution of maxwell equations from zero frequency to microwave frequencies. *IEEE Transactions on Antennas and Propagation*, 48:1635–1645, October 2000.

[52] Zhenhai Zhu, Jingfang Huang, Ben Song, and J. K. White. Improving the robustness of a surface integral formulation for wideband impendance extraction of 3D structures. *International Conference on Computer Aided-Design*, pages 592–597, 2001.