Accurate and Efficient Three-Dimensional Electrostatics Analysis using Singular Boundary Elements and Fast Fourier Transform on Multipoles (FFTM)

ONG ENG TEO

NATIONAL UNIVERSITY OF SINGAPORE

2003

Accurate and Efficient Three-Dimensional Electrostatics Analysis using Singular Boundary Elements and Fast Fourier Transform on Multipoles (FFTM)

ONG ENG TEO
(*B. ENG. (Hons.) NUS*)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2003

Acknowledgements

First and foremost, I would like to thank late Associate Professor Lee Kwok Hong for giving me this opportunity to pursue a Ph.D in Engineering. His passion for research has being a great source of motivation for me during my candidature. Without him, this thesis would not have been possible.

I would also like to thank my co-supervisor Dr. Lim Kian Meng. Although he only came into the picture in the latter part of my candidature, his guidance and encouragement is most appreciated.

Also special thanks to Dr. Su Yi for providing the pre-processing program, which otherwise would take me another few more months if I were to write it myself.

Last but not least, I would like to thank the university for providing the financial supports for the three and a half years of study in NUS, and the augmentation from A*STAR (formerly NSTB). And also many thanks are conveyed to the Department of Mechanical Engineering, Centre for Advanced Computations in Engineering Science (ACES), and Centre for IT and Applications (CITA), for their material support to every aspect of this work.

Table of Contents

Summary

There are two main contributions in this thesis, namely: (i) improving the accuracy of the Boundary Element Method (BEM) in the analysis of electrostatic problems by using singular boundary elements, and (ii) developing a fast algorithm, namely the Fast Fourier Transform on Multipoles (FFTM) for rapid solution of the integral equation in the BEM.

It is well known that the electric flux or surface charge density can become infinite at sharp corners and edges, and standard boundary elements with shape functions of low order polynomials fail to produce accurate results at these singular locations.

This thesis describes the formulation and implementation of new singular boundary elements to deal with these corner and edge singularity problems. These singular elements can accurately represent the singularity behaviour of the edges and corners because they include the correct order of singularity in the formulations of the shape functions. The main contribution here is the development of a general methodology for formulating singular boundary elements of arbitrary order of singularity.

It is demonstrated that the use of the singular elements can produce more accurate results than the standard elements. Furthermore, it is also shown to be more accurate than the "regularized function method" (for two-dimensional analysis) and *h*- mesh refinement method (for three-dimensional analysis). The singular elements are also used in electromechanical coupling simulations of some micro-devices. It is observed that using the singular elements gives rise to larger deformation in comparison to the standard elements. This indicates that ignoring the corner and edge singularities (as in standard elements) in the electrostatic analysis is likely to underestimate the true deformation of the micro-structures in the simulations. However, in terms of the pull-in voltage, the effect of the singular elements is less significant due to the pull-in phenomenon.

BEM generates a dense linear system, which requires $O(n^3)$ and $O(n^2)$ operations if solved using direct methods, such as Gaussian Elimination, and iterative methods, such as GMRES, respectively. This obviously becomes computationally inefficient as the problem size *n* increases.

In this thesis, a fast algorithm, called the Fast Fourier Transform on Multipoles (FFTM) method, is proposed and implemented for the rapid solution of the integral equation in the BEM. The speedup in the algorithm is achieved by: (i) using the multipole expansion to approximate "distant" potential fields, and (ii) evaluating the approximate potential fields by discrete convolution via FFT.

It is demonstrated that the FFTM provides relatively good accuracy, and is likely to be more accurate than the Fast Multipole Method (FMM) for the same order of multipole expansion (at least up to the second order). It is also shown that the FFTM has approximately linear growth in terms of computational time and memory storage requirements. This means that it is as efficient as existing fast methods, such as the FMM and precorrected FFT approach.

List of Figures

List of Tables

1

Introduction

In the computational arena, researchers strive continuously to improve numerical simulations, both in terms of accuracy and efficiency. The needs for better performance in numerical simulations are forever in demands, as their roles in the design and development of new products become more important. This is further promoted by the rapid increases in the size of the problems people are solving.

One typical application is the simulations of Micro-Electro-Mechanical Systems (MEMS), also known as Micro-System Technology (MST). MEMS is a new process technology, device concept and application that generates new markets for the field of integrated micro-sensors and micro-actuators. Some existing MEMS devices are pressure-sensing devices, inkjet print heads, airbag accelerometers, micro-gyroscope, micro-optical devices, micro-fluidic systems and micro-actuators/motors. Every new MEMS product is essentially a research project that has a long and expensive development cycle. To improve on the situation, Computer-Aided-Design/Engineering (CAD/CAE) tools are often used [1-3], which help MEMS designers to explore the unknown in hours instead of months. Some of the existing design tools that are specially developed for MEMS designs are MEMCAD[a] [4-6], IntelliCAD[b] [7] and SOLIDIS[c] [8].

This thesis investigates the physical simulations of multiple coupled energy domains, where the two coupling domains are the electrostatics and mechanical domains. Coupling arises when electrostatics forces, which are generated by the applied electrical voltages, deform parts of the structures that in turn induce mechanical restoring forces within the structures. Electromechanical coupling analysis is required to solve for the self-consistent state, where the electrostatics forces counter-balance the mechanical forces [9-15]. Boundary Element Method (BEM) is often employed to solve the electrostatics analysis, whereas Finite Element Method (FEM) does the mechanical analysis. In this study, we aim to improve the electrostatics analysis, both in term of the accuracy and efficiency.

[a] **MEMCAD**. Conventor Inc., 4001 Weston Parkway, Cary, NC 27513.
[b] **IntelliCAD**. IntelliSense Corp., Wilmington, MA 01887, USA.
[c] **SOLIDIS**. ISE Integrated Systems Engineering AG, Zurich Switzerland.

1.1 Improving Accuracy of Electrostatics Analysis

The first part of this thesis aims to improve the accuracy of the electrostatics analysis in MEMS device simulations. Generally, the major sources of errors in BEM are:

(1) *Modeling errors* - Due to the simplifications made when transforming real physical problems into numerical models. They can occur in geometrical modeling, applied boundary conditions and material properties.

(2) *Implementation errors* - They arise from the numerical techniques used in the implementation of BEM. One such error is due to the numerical integrations of the boundary integrals, especially dealing with the singular integrals.

(3) *Discretization errors* - This contributes to significant errors in BEM analysis, which consist of geometrical and variable discretization errors. The former is due to partitioning of boundary domains into many smaller panels/elements, which in most cases do not represent the original domains exactly. On the otherhand, variable discretization error arises because the basis functions used for the variables (usually of low order polynomials) cannot adequately describe the true solution. This is especially significant when the problem contains singularity solutions, such as in fracture mechanics [16-23], and corner singularities in potential problems [24-36].

This thesis aims to reduce the third source of errors, specifically to deal with the singularities that arise from sharp corners and edges of electrical conductor [24, 25, 34]. In this thesis, we have adopted the singular element method. Hence, the objective for the first part of the thesis is to develop and implement singular boundary elements for two and three-dimensional electrostatics analysis.

1.2 Improving Efficiency of Solution Method

It is well-known that BEM generates a dense linear system, which requires $O(n^3)$ and $O(n^2)$ operations if solved using direct methods, such as Gaussian Elimination, and iterative methods, such as GMRES [37], respectively. This obviously becomes computationally inefficient when the problem size n increases. Recent developments in the solution of dense linear system utilize the matrix-free feature of the iterative methods, which only requires computing matrix-vector products that can be seen

as a potential evaluation process. This important observation has led to the developments of numerous fast algorithms. In general, these fast algorithms work by classifying the potential contributions into "near" and "distant" regions, where the "near" contributions are computed exactly as in standard BEM, while the "distant" ones are approximated. The various algorithms differ in the way the "distant" potential contributions are computed. Two such fast algorithms are the Fast Multipole Method (FMM) [38, 39, 40, 41, 42, 43, 44, 45] and the precorrected-FFT approach [46, 47, 48].

In this thesis, we propose an alternate fast algorithm that can also evaluate the dense matrix-vector products rapidly. The core of the method lies on recognizing the fact that potential calculations using multipole expansions can be expressed as discrete convolutions, which are computed rapidly using Fast Fourier Transform (FFT) algorithms [49]. We refer to it as the Fast Fourier Transform on Multipoles (FFTM) method. Hence, the objective of the second part of the thesis is to develop and implement FFTM for solving large three-dimensional electrostatics problems using BEM.

1.3 Thesis Organization

This thesis comprises of two main parts. Chapters 3 to 5 are concerned with improving the accuracy of the analysis, by using singular boundary elements. On the other hand, Chapters 6 and 7 discuss improving the computational efficiency for solving the dense linear system generated by BEM, with the development of FFTM.

Chapter 2 begins with an overview of the implementation of BEM for solving electrostatics problems. Chapter 3 reviews on the existing methods that were employed to improve the BEM accuracy. Chapters 4 and 5 describe the implementation and application of the singular element method in two and three-dimensional electrostatics analysis, respectively. Both chapters begin with discussions on the nature of the singularity problem. This is then followed by the formulation of the singular boundary elements. The numerical techniques that are employed to evaluate the boundary integrals are also discussed. Some examples are then solved to demonstrate the significant improvement in the accuracy achieved by using the singular boundary elements. Finally, concluding remarks are given at the end of both chapters.

In Chapter 6, we review some existing fast methods for solving large dense linear system of equations. This discussion leads to Chapter 7, the main text of the second part of the thesis on the development of

an alternate fast algorithm, namely FFTM. It begins with a detailed description of the algorithm, which is followed by a simple complexity analysis. It is then applied it to solve some numerical examples to investigate the accuracy and efficiency of the method. Last but not least, in Chapter 8, we summarize the main ideas and major contributions of this piece of work. Some recommendations on the future work are also discussed in the chapter.

This thesis also includes several appendices, which are denoted alphabetically. Appendix A describes the iterative solution method for dense linear system, namely GMRES, which is used extensively in this thesis. Appendix B presents the closed form singularity solution for two-dimensional corners, and also the numerical techniques used to determine the order of singularity for three-dimensional corners. Appendix C discusses the numerical integration techniques used to evaluate the singular boundary integrals. Appendix D describes a preprocessing program, which is implemented to identify and classify the singular boundary elements automatically. Appendix E briefly describes the solution method for the electromechanical coupling analysis. Finally, the real-valued version of multipole expansion is derived, and recursive formulas for the associated Legendre and trigonometric functions are given in Appendix F.

2

BEM for Electrostatics Analysis

Electrostatics analysis is performed to solve for the surface charge density distributions induced on the conductors due to applied electrical potentials. They are then used to compute the capacitance and electrostatics forces, which are very important in the functioning of many MEMS devices. Capacitance sensors, such as pressure sensors, accelerometers and micro-gyroscope, require the capacitance to be computed accurately. Similarly, accurate evaluation of electrostatics force is essential since it is the driving force of many micro-devices, such as comb-drive actuators, micro-optical switch devices, micro-pumps/valves and micro-motors.

This chapter begins with the formulations of Boundary Integral Equation (BIE), both in the direct and indirect approaches. Although indirect BIE is very effective in solving exterior problems, where problem domains are infinite or semi-infinite, care must be exercised when applying the appropriate boundary conditions. This issue is discussed in Section 2.2. Finally, an overview on the implementation of the BEM is presented in Section 2.3.

2.1 Formulations of Boundary Integral Equation

The governing equation for the electrostatics analysis of electrical conductors embedded in an infinite homogeneous dielectric, such as free space, is the Laplace equation,

$$\nabla^2 f(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \Omega \tag{2.1}$$

where $f(\boldsymbol{x})$ is the electrical potential at point $\boldsymbol{x}$, and Ω corresponds to the domain in which (2.1) is satisfied. The following sub-sections discuss the formulations of the BIE for (2.1).

2.1.1 Direct formulation by weighed residual technique

The direct boundary integral equation (DBIE) formulation, derived using weighted residual technique together with Divergence theorem and Green's identities, can be found in many BEM textbooks, such as [50, 51]. DBIE for potential problem is generally given by

$$a(\boldsymbol{x})f(\boldsymbol{x}) = \int_{\Gamma}\left[\frac{\partial f(\boldsymbol{x}')}{\partial n}G(\boldsymbol{x},\boldsymbol{x}') - f(\boldsymbol{x}')\frac{\partial G(\boldsymbol{x},\boldsymbol{x}')}{\partial n}\right]d\Gamma(\boldsymbol{x}') \tag{2.2}$$

where $\boldsymbol{x}$ and $\boldsymbol{x}'$ denote the field and source points, respectively, and $a(\boldsymbol{x})$ is generally known as the jump term, which arises when $\boldsymbol{x}$ is moved to the boundary and is dependent on the geometry of the boundary at $\boldsymbol{x}$. $G(\boldsymbol{x},\boldsymbol{x}')$ is the fundamental solution for potential problems and is given by

$$\begin{aligned} G(\boldsymbol{x},\boldsymbol{x}') &= \frac{1}{2p}\ln\frac{1}{\|\boldsymbol{x}-\boldsymbol{x}'\|}, \quad \text{for 2D analysis.} \\ G(\boldsymbol{x},\boldsymbol{x}') &= \frac{1}{4p\|\boldsymbol{x}-\boldsymbol{x}'\|}, \quad \text{for 3D analysis.} \end{aligned} \tag{2.3}$$

where $\|\boldsymbol{x}-\boldsymbol{x}'\|$ is the distance between point $\boldsymbol{x}$ and $\boldsymbol{x}'$. The second integral on the right hand side of (2.2) exists only in the sense of Cauchy Principle Value (CPV) when $\boldsymbol{x} = \boldsymbol{x}'$. Generally, this integral together with $a(\boldsymbol{x})$ can be obtained indirectly by using the constant potential condition (analogous to the rigid body motion condition in elastostatic problem).

Although DBIE is widely regarded as the standard BEM formulation, it is not efficient in solving exterior problems, as it requires a bounded problem domain. This implies that an artificially large boundary is needed to represent the infinite boundary, which increases the problem size significantly. Hence, for exterior problems, it is preferable to employ the indirect formulation.

2.1.2 Indirect formulation using surface layer sources

There are two possible kinds of sources that can exist on the surface of the electrical conductors when subjected to applied potentials. They are the single layer (surface charge) and double layer (dipole) sources. For purely Dirichlet problems, only the single layer source exists. In this case, the potential at any point $\boldsymbol{x}$ in the problem domain Ω is given by the Fredholm integral equations of the first kind,

$$f(\boldsymbol{x}) = \int_{\Gamma} s(\boldsymbol{x}')\frac{1}{4pe\|\boldsymbol{x}-\boldsymbol{x}'\|}d\Gamma(\boldsymbol{x}'), \quad \boldsymbol{x}\in\Omega \tag{2.4}$$

where $s(\boldsymbol{x}')$ is the surface charge distribution on the boundary Γ. Equation (2.4) is essentially based on the principle of superposition, which states that the potential at $\boldsymbol{x}$ is generated by summing the effects from all the surface charges that exist in the domain. Indirect boundary integral equation (IDBIE) is then derived from (2.4) by taking point $\boldsymbol{x}$ to the boundary Γ, which is done in a limiting

process (see appendix of [45]). This process however does not alter the governing equation, that is, (2.4) is still valid when $\boldsymbol{x}$ is on the boundary.

2.1.3 Indirect formulation derived from direct formulation

This alternate formulation is presented because it reveals an important issue regarding the use of IDBIE, which is not obvious from (2.4). That is, (2.4) alone does not govern the electrostatics problem completely.

For electrical conductors, the surface charge density is related to the normal potential flux by the following relation,

$$s(\boldsymbol{x}) = e\,\frac{\partial f(\boldsymbol{x})}{\partial n} \tag{2.5}$$

where e is the dielectric constant of the medium. For uniform Dirichlet problems, (2.2), after substituting (2.5) and assuming e = 1.0, can be rewritten as,

$$\begin{aligned} a(\boldsymbol{x}) f(\boldsymbol{x}) = \sum_{i=1}^{m} \left(\int_{\Gamma_i} G(\boldsymbol{x}, \boldsymbol{x}') s(\boldsymbol{x}') d\Gamma(\boldsymbol{x}') - f_i \int_{\Gamma_i} \frac{\partial G(\boldsymbol{x}, \boldsymbol{x}')}{\partial n} d\Gamma(\boldsymbol{x}') \right) \\ + \left(\int_{\Gamma_\infty} G(x, x') s(x') d\Gamma(x') - f_\infty \int_{\Gamma_\infty} \frac{\partial G(x, x')}{\partial n} d\Gamma(x') \right) \end{aligned} \tag{2.6}$$

where f_i and Γ_i denote the potential and boundary of the i-th conductor, for i=1…m, respectively, while f_∞ and Γ_∞ corresponds to that on an artificially large surface that approximates the boundary at infinity. By using the constant potential condition, the jump term is derived as

$$a(\boldsymbol{x}) = -\sum_{i=1}^{m} \int_{\Gamma_i} \frac{\partial G(\boldsymbol{x}, \boldsymbol{x}')}{\partial n} d\Gamma(\boldsymbol{x}') - \int_{\Gamma_\infty} \frac{\partial G(\boldsymbol{x}, \boldsymbol{x}')}{\partial n} d\Gamma(\boldsymbol{x}') \tag{2.7}$$

Note that when $\boldsymbol{x}$ falls on the *i-th* conductor, the contributions from the other conductors to (2.7) are zeros, and that from Γ_∞ is equal to -1. This observation comes from the property of the Green's function, which states that

$$\int_{\Gamma_i} \frac{\partial G(\boldsymbol{x}, \boldsymbol{x}')}{\partial n} d\Gamma(\boldsymbol{x}') = \begin{cases} -1 & \text{if } \boldsymbol{x} \in \Omega_i \\ 0 & \text{otherwise} \end{cases} \quad \begin{pmatrix} \text{property of} \\ \text{Green's function} \end{pmatrix} \tag{2.8}$$

where Ω_i corresponds to close domain bounded by *i-th* conductor's surfaces, and since $\boldsymbol{x}$ always falls within the domain bounded by Γ_∞, hence its contribution is -1.

It is also noted that

$$\lim_{\|x-x'\|\to\infty} \int_{\Gamma_\infty} G(\boldsymbol{x},\boldsymbol{x}')s(\boldsymbol{x}')d\Gamma(\boldsymbol{x}') \approx G_\infty(\boldsymbol{x},\boldsymbol{x}')\int_{\Gamma_\infty} s(\boldsymbol{x}')d\Gamma(\boldsymbol{x}') = 0 \tag{2.9}$$

since $G_\infty(\boldsymbol{x},\boldsymbol{x}') \to 0$ as $\|\boldsymbol{x}-\boldsymbol{x}'\|_\infty \to \infty$, and $\int_{\Gamma_\infty} s(\boldsymbol{x}')d\Gamma(\boldsymbol{x}')$ is a finite quantity as explained as follows.

For a closed system, the total charges induced on the surfaces of the conductors and the infinite boundary must be conserved, that is

$$\begin{aligned} &\sum_{i=1}^{m}\int_{\Gamma_i} s(\boldsymbol{x}')d\Gamma(\boldsymbol{x}') + \int_{\Gamma_\infty} s(\boldsymbol{x}')d\Gamma(\boldsymbol{x}') = 0 \\ \Rightarrow &\sum_{m=1}^{m}\int_{\Gamma_m} s(\boldsymbol{x}')d\Gamma(\boldsymbol{x}') = -\int_{\Gamma_\infty} s(\boldsymbol{x}')d\Gamma(\boldsymbol{x}') = Q \end{aligned} \tag{2.10}$$

where Q is the total induced charge on the conductors' surfaces only, which is equal in magnitude to the total charge induced on the infinite boundary.

Hence, combining (2.6) to (2.9), we obtain

$$f(\boldsymbol{x}) - f_\infty = \sum_{i=1}^{m}\int_{\Gamma_i} G(\boldsymbol{x},\boldsymbol{x}')s(\boldsymbol{x}')d\Gamma(\boldsymbol{x}') \tag{2.11}$$

Unlike the IDBIE presented in Section 2.1.2, this approach leads to two governing equations, namely (2.10) and (2.11), that must be satisfied for exterior potential problems. However, there are three unknowns (s, Q and f_∞) in the two equations, which renders the problem undetermined. In order to resolve the problem, either Q or f_∞ need to be specified as applied boundary condition to eliminate one of the unknowns. This issue on the appropriate choice of boundary conditions is discussed in the following section.

2.2 Boundary Conditions for Exterior Problems

2.2.1 Potential at infinity is zero, $f_\infty = 0$

For a system of m conductors, each at potential of f_i and with charge Q_i, for i=1…m, the electrostatics potential energy can be expressed in terms of the potentials and capacitance [52]. The capacitance defines the ability of the conductors to store electric charges. For a given configuration of conductors, the total charge induced on i-th conductor is related to the potentials and capacitance by

$$Q_i = \sum_{j=1}^{m} C_{ij} f_j , \quad i = 1 \ldots m \tag{2.12}$$

where C_{ii} corresponds to the self-capacitance, and C_{ij} are the induced capacitance that represents the capacitive coupling between conductors *i* and *j*, where *i*, *j*=1…*m*, and $i \neq j$.

Suppose the infinite boundary is also regarded as a conductor, then (2.12) becomes

$$Q_i = \sum_{j=1}^{m} C_{ij} f_j + C_{i\infty} f_\infty , \quad i = 1 \ldots m \tag{2.13}$$

where $C_{i\infty}$ is the induced capacitance of the infinite boundary with respect to the *i*-th conductor. To determine the self-capacitance C_{ii}, a unit voltage is applied on conductor *i*, while the others are set to zeros (including the infinite boundary, that is, $f_\infty = 0$). From (2.13), the positive charges induced on conductor *i* is equivalent to the self-capacitance of the conductor for the given configuration of conductors, while the negative charges on the other conductors correspond to the induced capacitance. Notice that by setting $f_\infty = 0$, (2.11) is reduced to (2.4).

2.2.2 Total induced charge on infinite boundary is zero, *Q = 0*

In most electrical circuitry, potentials are defined in a relative sense, usually with respect to the ground that is assumed to be zero. Hence, (2.4) cannot be used directly since it only computes absolute potential, which is usually not given. In other words, the assumption that $f_\infty = 0$ may not be appropriate. In this case, one possible solution is to set $Q = 0$, implying that no electrical fluxes that emit from the conductors can reach the infinite boundary. This assumption is obviously more appropriate for problems where the conductors are packed closely together. One such scenario is when a system of conductors is placed over an infinitely large planar ground. This can approximately be seen in many MEMS devices, where microstructures are suspended over a large substrate (usually grounded).

For such problems, the computational cost can be reduced significantly by using the method of images [52] with the grounded plane placed at $x_3 = 0$. This approach is based on the principle of superposition, where the potential above the ground plane is induced by two sets of charges; namely the actual charges above the ground plane, and its image charges that are mirrored about the ground plane. By

setting the potential at the ground plane to zero explicitly defined the datum for the potential. In other words, the potentials at all other field points are relative potential with respect to this datum potential. The potential at point $\boldsymbol{x}$ due to a unit charge at $\boldsymbol{x}'$ is

$$f(\boldsymbol{x}) = \frac{1}{4\pi\varepsilon\sqrt{(x_1 - x_1')^2 + (x_2 - x_2')^2 + (x_3 - x_3')^2}} - \frac{1}{4\pi\varepsilon\sqrt{(x_1 - x_1')^2 + (x_2 - x_2')^2 + (x_3 + x_3')^2}} \tag{2.14}$$

The first term corresponds to the effect due the actual charge at position (x_1, x_2, x_3), and the second term is that due to image charge, which has an opposite charge polarity and at the mirror position $(x_1, x_2, -x_3)$. Equation (2.14) is modified Green's function, which is useful in this scenario because it removes the necessity to model the large planar ground at $x_3 = 0$. Hence, using the modified Green's function, more realistic simulations of the MEMS devices can be performed at a reasonable cost.

2.3 Implementation of BEM for Electrostatics Analysis

This section briefly summarizes the implementation of BEM for electrostatics analysis. Generally, it comprises of the following steps: 1) boundary element discretization, 2) choosing the BEM schemes, and 3) solving the dense linear system of equations generated by BEM.

2.3.1 Boundary element discretization

The starting point of the discretization process consists of approximating the boundary by a set of N_E curves (two-dimensional) or polygons (three-dimensional), often referred to as panels or elements, such that $\hat{\Gamma} = \bigcup_{e=1}^{N_E} \Gamma_e$, is the approximated boundary. Both the geometry and variables of the boundary elements are then approximated in the following form as

$$\hat{u}_e(x) = \sum_{k=1}^{n_e} J_k(x)\hat{u}_e^k \tag{2.15}$$

where $J_k(x)$ are the basis functions, $\hat{u}_e^k$ are the nodal values at the k *-th* node of the element, and n_e is the number of nodes on the element. Note that the basis functions for the geometry and variables need

not be the same. But when they are identical, the element is referred to as iso-parametric element. Equation (2.15) can be written more compactly as

$$\hat{u}(x) = \sum_{i=1}^{N} \hat{u}_i \Theta_i(x) \tag{2.16}$$

where $\hat{u}(x)$ is represented as a linear combination of a set of N linearly independent expansion functions $\Theta_i(x)$ that is weighed by $\hat{u}_i$ at N discrete points.

After the discretization process, the DBIE given in (2.2) and IDBIE in (2.4) become

$$\begin{aligned} R(\boldsymbol{x}) = a(\boldsymbol{x}) f(\boldsymbol{x}) + \int_{\hat{\Gamma}} \left(\sum_{k=1}^{N} f_k \Theta_k^f(\boldsymbol{x}') \frac{\partial G(\boldsymbol{x}, \boldsymbol{x}')}{\partial n} \right) d\Gamma(\boldsymbol{x}') \\ - \int_{\hat{\Gamma}} \left(\sum_{k=1}^{N} \frac{\partial f_k}{\partial n} \Theta_k^q(\boldsymbol{x}') G(\boldsymbol{x}, \boldsymbol{x}') \right) d\Gamma(\boldsymbol{x}') \end{aligned} \tag{2.17}$$

$$R(\boldsymbol{x}) = f(\boldsymbol{x}) - \int_{\hat{\Gamma}} \left(\sum_{k=1}^{N} s_k \Theta_k^s(\boldsymbol{x}') \frac{1}{4\pi\varepsilon \|\boldsymbol{x} - \boldsymbol{x}'\|} \right) d\Gamma(\boldsymbol{x}') \tag{2.18}$$

where $\Theta_k^f(\boldsymbol{x})$, $\Theta_k^q(\boldsymbol{x})$ and $\Theta_k^s(\boldsymbol{x})$ are the expansion functions of f, q and s, respectively, and $R(\boldsymbol{x})$ is the residual error function that arises from the approximations in the discretization process. For well-conditioned problems, $R(\boldsymbol{x})$ is a good measure of the discretization errors, and hence the next step is to minimize it. The simplest approach to carry out this task is to use the point collocation scheme.

2.3.2 Collocation BEM

In this approach, the residual is forced to be zero at N points in the solution domain, usually chosen to coincide with the interpolation nodes. Hence, the collocation BEM equations for (2.17) and (2.18) are

$$\begin{aligned} a(\boldsymbol{x}^i) f(\boldsymbol{x}^i) = -\int_{\hat{\Gamma}} \sum_{k=1}^{N} f_k \Theta_k^f(\boldsymbol{x}') \frac{\partial G(\boldsymbol{x}^i, \boldsymbol{x}')}{\partial n} d\Gamma(\boldsymbol{x}') \\ + \int_{\hat{\Gamma}} \sum_{k=1}^{N} q_k \Theta_k^q(\boldsymbol{x}') G(\boldsymbol{x}^i, \boldsymbol{x}') d\Gamma(\boldsymbol{x}'), \quad \text{for } i = 1,2,..., N \end{aligned} \tag{2.19}$$

$$f(\boldsymbol{x}^i) = \int_{\hat{\Gamma}} \sum_{k=1}^{N} s_k \Theta_k^s(\boldsymbol{x}') \frac{1}{4\pi\varepsilon \|\boldsymbol{x}^i - \boldsymbol{x}'\|} d\Gamma(\boldsymbol{x}'), \quad \text{for } i = 1,2,..., N \tag{2.20}$$

2.3.3 Solving dense linear system of equations

After applying the collocation BEM scheme and the boundary conditions in (2.19) and (2.20), the problem is reduced a dense linear system of equations

$$\boldsymbol{A}\vec{x} = \vec{b} \tag{2.21}$$

where $\boldsymbol{A}$ is a fully-populated N x N coefficients matrix, $\vec{x}$ is a vector that contains all the unknowns, and $\vec{b}$ is a known vector as a result of the applied boundary conditions.

Solving (2.21) by direct methods, such as Gaussian Elimination, require $O(N^3)$ operations, which is computationally expensive if N exceeds several thousands. To improve on the situation, iterative methods were developed [53, 54], which require only $O(N^2)$ operations. Generalized Minimal RESidual (GMRES) is one such iterative solver that is most suitable for solving dense matrix equations generated by BEM. A comprehensive discussion and implementation of GMRES is presented in Appendix A. The computational cost can be further reduced by utilizing the matrix-free feature of the iterative methods, which only requires computing matrix-vector products that correspond to potential calculations. This important observation has led to the development of numerous fast algorithms, such as FMM [38-45] and precorrected-FFT [46-48], which is only $O(N)$ or $O(N \log N)$. A more detailed literature review on the fast algorithms is given in Chapter 6, and in Chapter 7, we present an alternate fast algorithm, the Fast Fourier Transform on Multipoles (FFTM).

3

Approaches to Improve BEM Accuracy

As mentioned in Section 1.1, one major source of error in BEM comes from discretization of the variables. This error is especially significant when low order basis functions are used in the problem that contains singular solutions. This chapter reviews on the approaches that were developed to reduce this error.

Broadly speaking, the methods that were developed to improve the accuracy to singular problems can be classified into three major groups, namely the mesh refinement techniques, the singular elements and singular function methods. Mesh refinement techniques tend to be less accurate than the other two methods, because they are not specially designed to deal with the singularity problem. Rather, it is the nature of the adaptive algorithms that reveal and treat the singularities indirectly. This means that they require no prior information about the singularities, which is an advantage over the other two methods. The singular elements and singular function methods require prior knowledge of the locations of the singularity fields. In addition, they also need to know the actual singularity behaviors, in terms of the order of singularities and the singularity profiles (corresponding to the eigenvalues and eigenvectors of the eigenproblem that is associated with a given geometry). The singular element method usually needs to know the order of singularity (eigenvalues) only, whereas the singular function approach also requires the singularity field variations (eigenfunctions). In general, the inclusion of the eigenfunctions by the singular function method can produce more accurate solutions. However, the difficulty to derive these eigenfunctions has limited the extension of the singular function method to three-dimensional analysis.

In the following sections, the three methods will be discussed in greater details. It is remarked the literature review here is far from being a complete one. Nevertheless, it should provide readers with good overviews of the three approaches.

3.1 Adaptive Mesh Refinement Techniques

Adaptive mesh refinement techniques are iterative in nature, where one is often required to solve a given problem a few times before attaining a good solution. In general, they comprise of the following three processes:

(i) ***Error estimation process***: This estimates the discretization error of the solution, and provides an error indicator for the refinement process, which is also used as a termination criterion for the iteration.

(ii) ***Mesh refinement process***: This improves the solution by the *h*-, *p*- and *r*-refinement schemes, or their combinations.

(iii) ***Adaptive tactics process***: This determines the elements to be refined by using the error estimator in (i), and the mesh refinement scheme in (ii) is then carried.

Mesh refinement is an intensively researched area, especially during the late 1980's and the early 1990's. Readers are referred to [55-57] for more detailed reviews on this topic. The following sub-sections briefly discuss the error estimations and the mesh refinement processes. The adaptive tactics process is not further elaborated, since the adaptive algorithms follow naturally once the choices of the error estimation and the mesh refinement schemes were made.

3.1.1 Error estimations

Residual error type

As mentioned in Section 2.3.1, the residual of the BIE, as given in (2.17) and (2.18), is a good indication of the variables errors, and is often used to estimate the variables errors by assuming the variations of the residual functions on the element [58-64]. Figure 3.1 shows the residual interpolation function for the linear element used by Dong and Parreira [64], where the residual R_3 is obtained by applying the residual equation at the midpoint of the linear element.

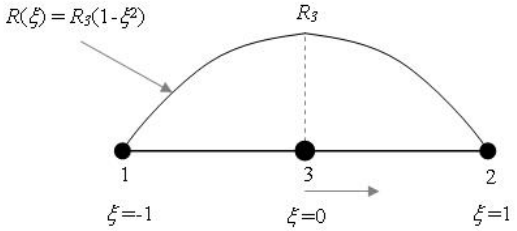


Figure 3.1. Residual interpolation approximation for linear element.

Interpolation error type

"Exact" solution is assumed to be that obtained by using higher order interpolation functions. The error estimator is the difference between the numerical and "exact" solutions [65-67]. Consider a simple example as depicted in Figure 3.2. Suppose $f(x)$ is approximated by piecewise linear interpolation functions defined at some discrete points. Then fitting a cubic interpolation function through three adjacent points gives the estimated variable error as indicated by the shaded regions.

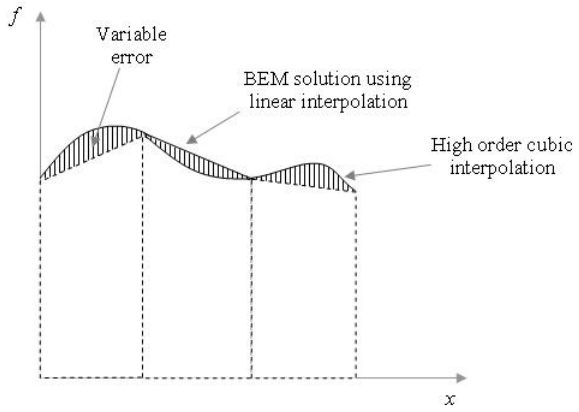


Figure 3.2. Error estimation by higher interpolation function.

Boundary integral equation error type

Suppose f^* and q^*, and $\hat{f}$ and $\hat{q}$ denote the exact solutions and the numerical solutions, respectively. Substituting both sets of solutions into the boundary integral equation associated with the collocation point $\boldsymbol{x}^i$, and taking the difference gives [57]

$$a\left(\boldsymbol{x}^i\right)e_f\left(\boldsymbol{x}^i\right)=\int_\Gamma\left[e_q\left(\boldsymbol{x}'\right)G\left(\boldsymbol{x}^i,\boldsymbol{x}'\right)-e_f\left(\boldsymbol{x}'\right)\frac{\partial G\left(\boldsymbol{x}^i,\boldsymbol{x}'\right)}{\partial n}\right]d\Gamma\left(\boldsymbol{x}'\right) \tag{3.1}$$

where $e_f=f^*-\hat{f}$ and $e_q=q^*-\hat{q}$ are the variable errors. Equation (3.1) is the BIE for the variable errors. Hence, it can be solved using BEM if the residual of (3.1) is known or approximately computed. Kawaguchi and Kamiya [68] presented a sample point error analysis to solve (3.1).

3.1.2 Mesh refinement schemes

Mesh refinement schemes determine how the elements are to be refined in order to improve the numerical solutions. They can be classified into *h*-, *p*-, *r*- versions, and also their combinations.

h- refinement schemes

The solution is improved by increasing the number of elements, while the order of interpolation functions remains invariant (usually of low order polynomials). This refinement technique is simple to implement in BEM. However, the coefficient matrix has to be rebuilt after every mesh refinement, which makes this approach inefficient. To improve on the situation, the *h*- hierarchical refinement schemes were proposed [63, 64, 65, 66, 67, 69], which used the *h*- hierarchical interpolation functions to simulate the effects of the conventional *h*- refinement schemes, without having to physically subdivide the elements. A comparison of the standard and *h*- hierarchical linear interpolation functions is shown in Figure 3.3.

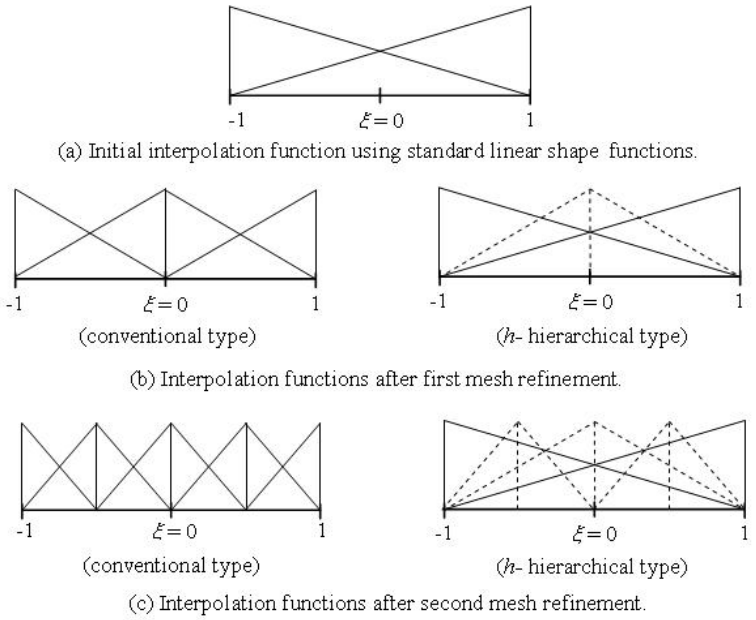


Figure 3.3. Standard versus *h*- hierarchical linear interpolation functions.

For the *h*- hierarchical approach, the previous set of interpolation functions is not affected by the current mesh refinement, and hence the coefficient matrix formed in the previous analysis can be used in the current analysis. This greatly improves the efficiency of the *h*- refinement scheme over the conventional approach, but it was reported by Zhao and Wang [69] that the coefficient matrix becomes ill-conditioned with increasing refinements.

p- refinement scheme

In the *p*- refinement scheme, the element mesh remains unchanged, but the order of the interpolation functions is increased. The improvement in the solution is achieved because higher order interpolation functions are more versatile in capturing the true solution. The conventional *p*- refinement scheme used the Lagrange interpolation formula to generate polynomial interpolation functions. But just like in the *h*- refinement scheme, this approach is inefficient. Hence, an alternate scheme was proposed, which is of the "hierarchical type" [70, 71, 72]. There exist two types of *p*- hierarchical interpolation functions, namely the Legendre polynomials [71, 72] in (3.2), and Peano's functions [70] in (3.3):

$$N_k = \frac{1}{2^{(k-2)}(k-1)!}\frac{d^{(k-2)}}{dx^{(k-2)}}\left[\left(1-x^2\right)^{(k-1)}\right], \quad \text{for } k \geq 2 \tag{3.2}$$

$$N_k = \frac{1}{k!}\left(x^k - b\right), \quad \text{where } k \geq 2, \text{ and } b\begin{cases} 1 \text{ if } k \text{ even} \\ x \text{ if } k \text{ odd} \end{cases} \tag{3.3}$$

r- refinement scheme

The *r*- refinement scheme is also known as the mesh redistribution method [60, 61, 73, 74, 75]. In this scheme, both the number of elements and the order of interpolation function remain invariant, but the collocation nodes are relocated so as to minimize an object function, such as the maximum error norm or the global error derived from the residual of the integral equation. In this sense, this approach can be seen as an optimization process, which utilizes limited degree of freedoms to achieve the best performance in term of accuracy. However, this scheme does not guarantee convergence to the exact solution, since this cannot be achieved by simply rearranging the nodal points alone. On the other hand, the exact solution can theoretically be attained by *h*- and *p*- schemes, by using infinitesimal elements for the *h*- method, and infinite order of interpolation functions for the *p*- method.

Combination schemes

The above-mentioned schemes have their pros and cons. Hence, different combinations of these schemes are employed to devise new schemes that make use of the advantages to compromise the disadvantages. Two combined schemes were developed, namely the *hp*- [58, 76] and *hr*- [62, 77] refinement schemes.

3.2 Singular Elements Method

Singular elements have their interpolation functions modified from those of the standard elements, mostly in an ad hoc manner, so that the singularity behavior of the field variables is correctly described. Usually only the first term of the singularity solution is considered. It is remarked that this approach is not being widely used in the potential analysis [19, 24], but has received much greater attention in fracture mechanics research [16-23]. Generally, two ways of deriving the singular shape functions have been identified, namely modifying reference nodes, and modifying shape functions.

3.2.1 Modifying reference nodes

The most widely used singular element based this approach is the traction singular elements, which is used to model the $\frac{1}{\sqrt{r}}$ variation of the traction in the vicinity of the crack-tip or crack front. The idea is to shift the middle node of a two-dimensional quadratic element to the quarter-point position, as shown in Figure 3.4.

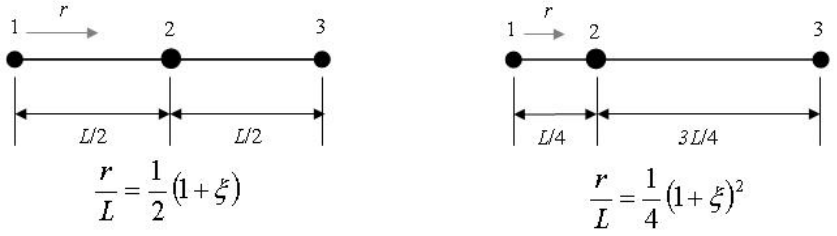


Figure 3.4. (a) Standard quadratic element, (b) Quarter-point quadratic element.

Substituting the quarter-point quadratic mapping function into the standard quadratic shape functions produces the $\sqrt{r}$ effect in the displacement field, that is,

$$u_i = A_i^1 + A_i^2\sqrt{r} + A_i^3 r \tag{3.4}$$

where $A_i^1 = u_i^1$, $A_i^2 = \frac{1}{\sqrt{L}}\left[-3u_i^1 + 4u_i^2 - u_i^3\right]$ and $A_i^3 = \frac{1}{L}\left[2u_i^1 - 4u_i^2 + 2u_i^3\right]$, and u_i^j is the nodal displacement at node j and in the i direction. The $\frac{1}{\sqrt{r}}$ singularity variation in the traction fields can be obtained by modifying (3.4). Blandford *et al*. [16], and Martinez and Dominguez [17] simply multiply

(3.4) by $\sqrt{\frac{L}{r}}$ to derive the singular shape functions for the traction field. Ariza *et al*. [18] further extended this concept to three-dimensional fracture mechanics analysis. Some researchers went on to employ this node shifting methodology to formulate singular elements for arbitrary order of singularity, by determining the optimum location of the middle node, through some curve-fitting process [22, 78]. However, it was pointed by Qian and Hasebe [79] that this approach is erroneous, because the behavior in the vicinity of the singular point is still $\sqrt{r}$, regardless of where the middle node is shifted in a quadratic element.

3.2.2 Modifying shape functions

In this approach, the shape functions for the displacement and the traction are usually derived in an ad hoc manner. Jia and Shippy [20] presented the following shape functions for the displacement and traction fields, respectively.

$$\begin{aligned} N_d^1 &= 1+\left(1+\frac{\sqrt{2}}{2}\right)\sqrt{1+x}+\frac{\sqrt{2}}{2}(1+x) \\ N_d^2 &= \left(2+\sqrt{2}\right)\sqrt{1+x}-\left(1+\sqrt{2}\right)(1+x) \\ N_d^3 &= -\left(1+\frac{\sqrt{2}}{2}\right)\left[\sqrt{1+x}-(1+x)\right] \end{aligned} \tag{3.5}$$

$$\begin{aligned} N_t^1 &= \frac{1}{\sqrt{1+x}}-\left(1+\frac{\sqrt{2}}{2}\right)+\frac{\sqrt{2}}{2}\sqrt{1+x} \\ N_t^2 &= \left(2+\sqrt{2}\right)-\left(1+\sqrt{2}\right)\sqrt{1+x} \\ N_t^3 &= -1-\sqrt{2}+\left(1+\sqrt{2}\right)\sqrt{1+x} \end{aligned} \tag{3.6}$$

where the traction shape functions are derived from (3.5) by dividing N_d^1 and N_d^2 by $\sqrt{1+x}$ and N_d^3 by $\sqrt{\frac{1+x}{2}}$. They also commented that the formulation of the singular shape functions was by no means unique. In fact, they developed four different sets of singular shape functions for the traction variable; the one presented above was chosen based upon numerical experiments. They later further extended their work to the three-dimensional crack problems in [21].

3.3 Singular Function Method

For two-dimensional potential problems, it is well known that the potential field in the vicinity of sharp corner is given by the asymptotic series

$$f(r,q) = f_o + \sum_{i=1}^{\infty} a_i r^{l_i} f_i(l_i q), \quad \text{as } r \to 0. \tag{3.7}$$

where (r,q) is the polar coordinates centred at the corner, l_i and $f_i(l_i q)$ are the eigenvalues and eigenfunctions that can be obtained analytically by separation of variables [52], and a_i are the unknown coefficients dependent on the applied boundary conditions. In general, the singular function method employs the truncated version of (3.7) in the solution process. There also exist many different types of singular function methods, and only some of them are discussed in the following sub-sections.

3.3.1 Subtraction of singularities

This approach removes the singularities from the solution so that the remaining variable field is smooth, and hence can be solved accurately by the standard methods, such as FDM, FEM and BEM. Wigley [28] did it in an iterative manner, which he called the subtraction of singularities approach. A similar method was also proposed by Igarashi and Honma [25], which they called the regularized function method.

Olson *et al*., on the other hand, developed the Integrated Singular Basis Function Method (ISBFM) [27]. The main difference between this approach and Wigley's method is that it is not iterative. This is achieved by using the following relation to generate the additional equations, which is derived from the Green's theorem.

$$\int_{\Gamma} \left(\frac{\partial \hat{u}}{\partial n} g_i - \hat{u} \frac{\partial g_i}{\partial n} \right) d\Gamma = 0, \quad \text{for } i = 1, 2, ..., N_s. \tag{3.8}$$

where $\hat{u} = u - \sum_{i=1}^{N_s} a_i g_i$ and $g_i = r^{l_i} f(l_i q_i)$. The Lagrange multipliers are employed to impose the essential boundary conditions. They later presented a boundary integral version of the ISBFM [26], in which the solution in the whole domain is solely approximated by the singular basis functions.

3.3.2 Boundary approximation methods

The problem domain is first divided into several sub-domains according to the singularity locations. In each singular sub-domain, special functions that can account for the singularities are employed, whereas the standard methods are used in the non-singular regions. Finally, the solution is obtained by enforcing the compatibility conditions at the sub-domains inter-boundaries.

Li *et al*. [29-33] proposed a combined method that used Ritz-Galerkin in the singular sub-domains, and FEM in the rest of the solution domains. In general, the asymptotic series in (3.7) are chosen to be the basis functions for the Ritz-Galerkin method. The compatibility conditions at the inter-boundaries are then enforced in a least squares sense [29], by hybrid-combined methods [30, 31], penalty-combined methods [32] and also their combinations [33].

3.4 Comments on the Three Approaches

3.4.1 Mesh refinement techniques

The mesh refinement techniques are iterative in nature, where a problem often has to be solved a number of times in order to arrive at the "correct" solution. The number of iterations depends on the convergence tolerance, and the refinement scheme employed. It is also dependent on the smoothness of the solution. For problems that contain singular solutions, it is expected to require more iterations to attain convergence. Hence, the computational cost may become too expensive to handle for singular problems.

Global error is often taken as the convergence criterion, such as the residual norms. However, "small" global error does not necessarily correspond to "small" local error. This is especially true in singularity problems where the local errors, in the vicinity of the singular regions, remain large despite small global error. In other words, the solutions in the singular regions are still poorly represented even when the convergence criterion is satisfied.

3.4.2 Singular element method

Singular elements incorporate the singular variations in their shape functions, often in a rather ad-hoc manner, by either modifying the reference nodes, or modifying the shape functions. Although the singular shape functions do not exactly describe the asymptotic solution, they are still able to produce

accurate solution, especially in the singular regions. This is because the solution in the singular region is usually dominated by the singular term of the asymptotic solution, which can be accurately represented by the singular shape functions.

The singular elements are used only in the regions where singularity solution is expected, and hence the exact singularity locations must be known a priori. Fortunately, this does not pose a difficult problem for the types of singularities investigated in this study, as they are due to sharp corners and edges, which can be identified easily using a pre-processing program. The geometry dependence also indicates that different singular elements have to be formulated to handle different types of singularity fields. Hence, this complicates the implementation of the singular elements method in three-dimensional analysis, as presented in Chapter 5.

3.4.3 Singular functions method

This approach has not been widely adopted by the engineering community. One possible reason is because the closed form singularity solutions for many practical engineering applications, such as fracture in a bi-material interface, are not available. Likewise, there is also no report of three-dimensional singularity analysis using this approach. In our opinion, it is very difficult and tedious to implement this method to solve three-dimensional singularity problems.

3.4.4 Method adopted in this thesis

In this thesis, we have adopted the singular element method for the following reasons. The singular function approach is first eliminated because no closed form singularity solution exists for three-dimensional problems. Although the singularity solution can be approximated numerically, its implementation is practically too tedious. On the contrary, the other two approaches were already being employed in three-dimensional singularity problems. Bactold *et al*. [76] employed the *hp*-adaptive mesh refinement technique to solve electrical potential problems, and singular elements were used extensively in the three-dimensional fracture mechanics analysis. Finally, the singular element method is preferred in this study because of its superior accuracy over the mesh refinement method.

4

Two-Dimensional Singular Elements

Two-dimensional analysis is first conducted as a preliminary investigation. This chapter begins with a general formulation of the two-dimensional singular elements of an arbitrary order of singularity. This is followed by a discussion of the numerical treatments of the singular integrals. Two numerical examples are then used to demonstrate the accuracy of the singular elements, namely the co-axial conductor and parallel conductor problems. The numerical results show that the present approach gives very accurate solutions. The effect of the size of the singular element is also investigated.

4.1 Formulation of Two-Dimensional Singular Elements

The solution to the two-dimensional Laplace equation is generally given by the asymptotic series in (3.7). For the specific case where uniform Dirichlet boundary condition is applied at the corner, the series solution becomes

$$f(r,q) = f_o + \sum_{k=1}^{\infty} a_k r^{\frac{kp}{y}} \sin\left(\frac{kpq}{y}\right) \tag{4.1}$$

where y is the interior angle bounded by the adjacent boundaries Γ_A and Γ_B as shown in Figure 4.1, and f_o corresponds to the applied potential at the corner. From (4.1), it is easily seen that the normal potential gradient $\frac{\partial f}{\partial n}$ becomes singular for re-entrant corner, that is $y > \pi$, where the first term of the series is singular.

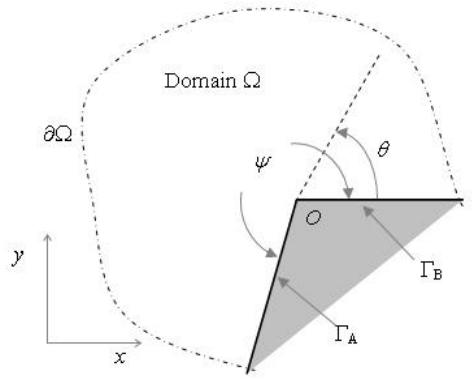


Figure 4.1. Two-dimensional potential field with a singular corner at *O*.

4.1.1 General formulation of singular element

Suppose the normal potential gradient is approximated by the first three terms of the series in (4.1), i.e.

$$\frac{\partial f}{\partial n} = Q_s r^s + Ar^a + Br^b \tag{4.2}$$

where Q_s is generally known as the generalized flux intensity factor, A and B are some constant coefficients, s is the order of singularity (possibly negative in value), and a and b are positive exponents. The values of s, a and b are dependent on the angle of the corner. In particular, for $0 < y < 2p$, it is observed that $s = \frac{p}{y} - 1$, $a = \frac{2p}{y} - 1$ and $b = \frac{3p}{y} - 1$. Now by letting $r = Lh$, where L is the length of the element, and h is the intrinsic coordinate $0 \le h \le 1$, (4.2) can be expressed in the local co-ordinates as

$$\frac{\partial f}{\partial n} = Q_s L^s h^s + A^* h^a + B^* h^b \tag{4.3}$$

where $A^* = L^a A$ and $B^* = L^b B$ are again constants.

It is important to note that the singular coefficient Q_s is retained in the formulation to ensure that the flux intensity factor is consistent for the two singular elements adjacent to the corner. Using the standard approach of formulating shape functions, the following requirements on the potential gradient are specified as

$$\begin{aligned} h \to 0, \quad & \frac{\partial f}{\partial n} \to \infty \\ h = 0.5, \quad & \frac{\partial f}{\partial n} = q_2 \\ h = 1.0, \quad & \frac{\partial f}{\partial n} = q_3 \end{aligned} \tag{4.4}$$

where q_2 and q_3 are the variable unknowns at the respective nodal positions. The first requirement is met naturally due to the singular term in (4.3). Applying the other two requirements and then solving for A^* and B^* gives

$$\begin{aligned} A^* &= \left[\frac{2^{a-s}-1}{2^{a-b}-1} - 1\right] L^s Q_s + \left[\frac{2^a}{1-2^{a-b}}\right] q_2 + \left[1 + \frac{1}{2^{a-b}-1}\right] q_3 \\ B^* &= \left[\frac{1-2^{a-s}}{2^{a-b}-1}\right] L^s Q_s + \left[\frac{2^a}{2^{a-b}-1}\right] q_2 + \left[\frac{1}{1-2^{a-b}}\right] q_3 \end{aligned} \tag{4.5}$$

Substituting them back into (4.3) gives

$$\frac{\partial f}{\partial n} = N_1^s Q_s + N_2^s q_2 + N_3^s q_3 \tag{4.6}$$

where N_i^s is the set of singular shape functions defined as

$$\begin{aligned} N_1^s &= \left\{ h^s + \left[\frac{2^{a-s}-1}{2^{a-b}-1} - 1 \right] h^a - \left[\frac{2^{a-s}-1}{2^{a-b}-1} \right] h^b \right\} L^s \\ N_2^s &= \frac{2^a}{2^{a-b}-1} \left[h^a - h^b \right] \\ N_3^s &= \left[1 + \frac{1}{2^{a-b}-1} \right] h^a - \left[\frac{1}{2^{a-b}-1} \right] h^b \end{aligned} \tag{4.7}$$

4.1.2 Specific formulation for $y = 3p/2$

To date, many MEMS devices have simple geometry, usually "rectangular" with right-angled corners and edges. This special case is considered here, that is, $y = 3\pi/2$. Substituting this value into (4.2) gives $s = -1/3$, $a = 1/3$, $b = 1$. Hence, the singular shape functions in (4.7), as plotted in Figure 4.2, are

$$\begin{aligned} N_1^s &= \left\{ h^{-\frac{1}{3}} - 2.58740\, h^{\frac{1}{3}} + 1.58740\, h \right\} L^{-\frac{1}{3}} \\ N_2^s &= 3.40483 \left(h^{\frac{1}{3}} - h \right) \\ N_3^s &= -1.70241\, h^{\frac{1}{3}} + 2.70241\, h \end{aligned} \tag{4.8}$$

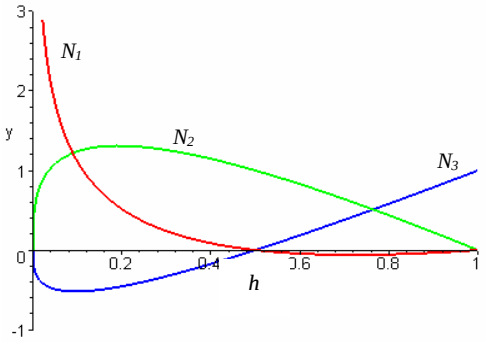


Figure 4.2. Singular shape functions for $s = -1/3$, $a = 1/3$ and $b = 1$.

The singular shape functions derived above are used only in the variations of the potential gradients for those elements with either node 1 or 3 falling on a re-entrant corner. These elements are known as the singular boundary elements.

4.2 Numerical Integration of Boundary Integrals

This section deals with the numerical integration of the boundary integrals that arise from the implementation of BEM. The types of boundary integrals to be dealt with are of the following forms:

$$I_1 = \int_{\Gamma_e} q(\boldsymbol{x}')G(\boldsymbol{x}, \boldsymbol{x}')d\Gamma(\boldsymbol{x}') \tag{4.9a}$$

$$I_2 = \int_{\Gamma_e} f(\boldsymbol{x}')\frac{\partial G(\boldsymbol{x}, \boldsymbol{x}')}{\partial n} d\Gamma(\boldsymbol{x}') \tag{4.9b}$$

where $q(\boldsymbol{x})$ corresponds to the normal potential gradient, which is possibly singular, and $f(\boldsymbol{x})$ is the potential, which is always represented by a quadratic variation. The integrals are usually transformed to the intrinsic co-ordinate, which are convenient forms to be evaluated by Gaussian quadrature schemes, as follows

$$I_1 = \int_{-1}^{1} N_q(x)G(\boldsymbol{x},x)|J(x)|dx \tag{4.10a}$$

$$I_2 = \int_{-1}^{1} N_f(x)\frac{\partial G(\boldsymbol{x},x)}{\partial n}|J(x)|dx \tag{4.10b}$$

where $N_f(x)$ and $N_q(x)$ denote the shape functions for the potential and potential gradient variables, respectively, and $|J(x)| = \sqrt{\left(\frac{dx_1}{dx}\right)^2 + \left(\frac{dx_2}{dx}\right)^2}$ is the Jacobian of transformation. In the following sub-sections, we describe the techniques used to compute (4.10a) and (4.10b) for different situations.

4.2.1 Non-singular integral

When the integrand is nonsingular within the integration limits, the standard Gaussian quadrature (specifically known as Gauss-Legendre [80]) is used, which approximates the integral with the formula

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{n} w_i f(x_i) \tag{4.11}$$

where n is the number of integration points, which also corresponds to the order of the Gaussian quadrature formula, and x_i and w_i denote the abscissa and weights of the *ith* Gauss point of the n-order formula, respectively.

4.2.2 Singular integral due to fundamental solution only

When the collocation point $\boldsymbol{x}$ falls on the element, (4.10a) and (4.10b) become singular due to the singular nature of the fundamental solution. The technique used to treat this singularity for (4.10a) includes a coordinate transformation, which transforms the Euclidean length $\|\boldsymbol{x}-\boldsymbol{x}'\|$ into the following general form [50]

$$\|\boldsymbol{x}-\boldsymbol{x}'\| = hR(h) \tag{4.12}$$

where $R(h)$ is nonsingular. Hence, the fundamental solution $G(\boldsymbol{x},x)$ becomes

$$\frac{1}{2p}\ln\frac{1}{\|\boldsymbol{x}-\boldsymbol{x}'\|} = \frac{1}{2p}\left[\ln\left(\frac{1}{h}\right)+\ln\left(\frac{1}{R(h)}\right)\right] \tag{4.13}$$

Equation (4.10a) is then separated into two components, with one containing the logarithm singularity, and the other is nonsingular. The nonsingular part can be evaluated using the standard Gaussian quadrature, while the singular one can be evaluated with logarithmic Gaussian quadrature [50].

The singular treatment for (4.10b) is as follows. For the case when $i = j$, the integral in (4.10b) becomes strongly singular and exists only in the sense of the Cauchy Principal Value. This integral can be obtained indirectly by using the constant potential condition, which gives

$$g'_{ii} = -\sum_{j=1,i\neq j}^{N} g'_{ij} \tag{4.14}$$

where g'_{ij} denotes the (i, j) entry of the coefficient matrix generated with (4.10b). On the other hand, when $i \neq j$, the singularity in the integrand is removed by the zeros of the shape functions $N_f(x)$, which satisfy the Kronecker delta property, that is,

$$N_i(x^j) = d_{ij} \quad \begin{cases} 0 \text{ if } i \neq j \\ 1 \text{ if } i = j \end{cases} \tag{4.15}$$

4.2.3 Singular integral due to singular shape function only

This singular integral occurs only in (4.10a), when $N_q(x)$ is the singular shape function N_1^s derived in (4.8). Strictly speaking, only the singular term h^s needs special treatment. In this case, the singular integral to be dealt with is of the form

$$I = \int_{-1}^{1} \left[\frac{1}{2}(1+x)\right]^{s} G(\mathbf{x},x)\,|J(x)|\,dx \tag{4.16}$$

A simple way to treat this singular integral is to use the variable transformation, $x = \frac{1}{2}(1+z)^2 - 1$, which transforms (4.16) into

$$I = \int_{-1}^{1} \left(\frac{1}{2}\right)^{2s} \left[(1+z)^{(1+2s)}\right] G(\mathbf{x},z)\,|J(z)|\,dz \tag{4.17}$$

which is no longer singular since $s = \frac{p}{y} - 1 > -\frac{1}{2}$ for $0 < y < 2p$.

4.2.4 Singular integral due to fundamental solution and singular shape function

This situation also only occurs in (4.10a), when the collocation point falls on the singular node of the singular element. The first task is to deal with the logarithmic singularity in the fundamental solution, which is done by using the same technique discussed in Section 4.2.2. After the appropriate transformations, the resulting integral becomes

$$I = \int_{0}^{1} f(h)\ln\left(\frac{1}{h}\right) J(h)\,|dh + \int_{-1}^{1} f(x)\ln\left(\frac{1}{R(x)}\right) |J(x)|\,dx \tag{4.18}$$

However, $f(h)$ and $f(x)$ may still be singular due to the singular shape functions. In this case, the second integral is handled in the same way as described in Section 4.2.3. As for the first integral, the transformation $h = z^2$ is used, thereby giving

$$\int_{0}^{1} f(h)\ln\left(\frac{1}{h}\right) J(h)\,|dh = \int_{0}^{1} [f(z)]4z\,\ln\left(\frac{1}{z}\right) J(z)\,|dz \tag{4.19}$$

in which the Jacobian of transformation, $dh = 2z\,dz$, is used to remove the weak singularity of the shape function.

4.3 Numerical Examples

4.3.1 Coaxial conductor example

This example concerns the capacitance extraction of a square coaxial transmission line, as shown in Figure 4.3. Only a quarter of the problem is analyzed due to symmetry. This problem is also known as the Dirichlet-Laplace problem on the L-shape domain [25], and it is one of the commonly used

benchmark problem for singularity analysis. This is an interior (closed domain) problem, and hence the direct BEM is used.

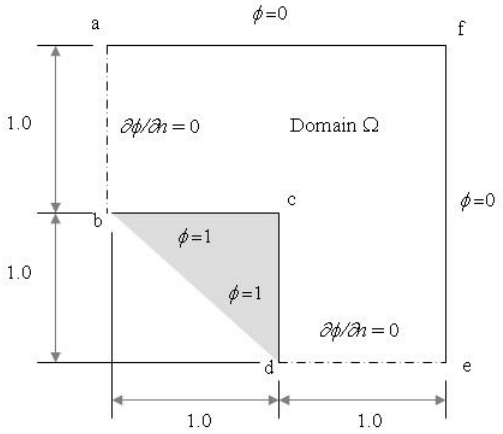


Figure 4.3. One quarter of the square coaxial conductor problem.

It is clear that sharp corner is actually a geometrical idealization, when the radius of curvature of the "corner" is very small compared the overall dimensions of the structure. The question of how small this radius should be is raised here. Hence, this example is first used to study the sharp corner idealization.

A non-dimensional variable $\boldsymbol{R}$, which is defined as the ratio of the curvature radius r to the characteristic length of the conductor L, is introduced. A very fine mesh, using the standard quadratic element, is used to solve for the cases with $\boldsymbol{R}$ = 0.1, 0.05, 0.02 and 0.01. The computed capacitance is then compared with the exact solution for the idealized case, which is C^* = 2.55852 [25] (scaled by $4\pi\varepsilon$). The capacitance and the normal potential gradient are related by

$$s = e\frac{\partial f}{\partial n} \tag{4.20}$$

$$C = \int_{\Gamma_{f=1}} s \, d\Gamma \tag{4.21}$$

where s is the surface charge density on the conductor and e is the dielectric constant of the medium. C is obtained by summing all the surface charges on the conductor that has a potential equal to one. The relative error is computed and plotted in Figure 4.4.

From Figure 4.4, it is noted that the relative error is very small (< 1%) for all the cases considered. A best fit equation for the relationship between the relative error and R is obtained, and is given by

$$\text{Relative error} = -273.92\boldsymbol{R}^3 + 73.548\boldsymbol{R}^2 + 3.2465\boldsymbol{R} \tag{4.22}$$

For small values of $\boldsymbol{R}$, this relationship can be approximated by the linear term alone. In this study, which is directed towards accurate numerical analyses of micro-devices, the ratio $\boldsymbol{R}$ is probably of the order from $1\text{x}10^{-3}$ to $1\text{x}10^{-6}$. This means that the relative error induced when making the sharp corner idealization is also of that order of magnitude. Hence, the sharp corner idealization is a valid assumption in this study, as the error induced is usually less than the tolerance needed.

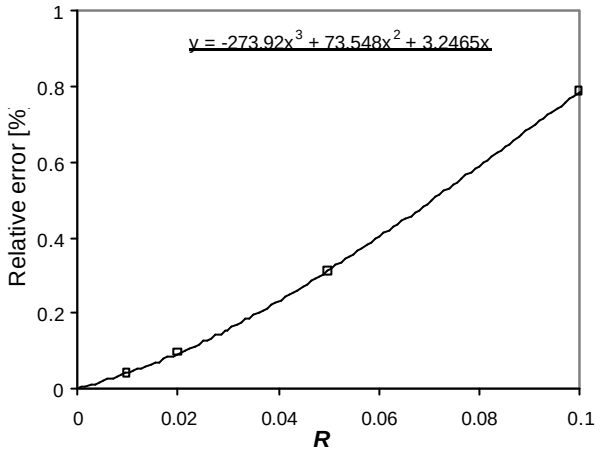


Figure 4.4. The results for the sharp corner idealization with different radius of curvature $\boldsymbol{R}$ values.

Next, this example is solved using the standard BEM with (i) constant, (ii) linear, (iii) quadratic elements (no singular treatment for these three cases), and (iv) quadratic with singular elements (present formulation). The relative error is again computed and plotted in Figure 4.5 as a function of the number of elements M. The results of Igarashi and Honma [25], employing the regularized function method, are also included.

From Figure 4.5, it is easily seen that the present formulation produces excellent accuracy in comparison to the rest. It is also noted that its convergence rate is approximately the same as the standard elements, whereas the regularized function method has a faster convergence rate. However, the present singular element is very accurate even for very coarse meshes. Generally, it is more accurate than the regularized function method by about two orders of magnitude. The generalized intensity factor Q_s of (4.2), which are obtained directly from the nodal variable, are tabulated in Table 4.1. These values compare very well with the results reported by Igarashi and Honma [25], and that

obtained by the linear extrapolation method on a very fine mesh, as shown in Figure 4.6. The results are tabulated in Table 4.1, and they show good agreement.

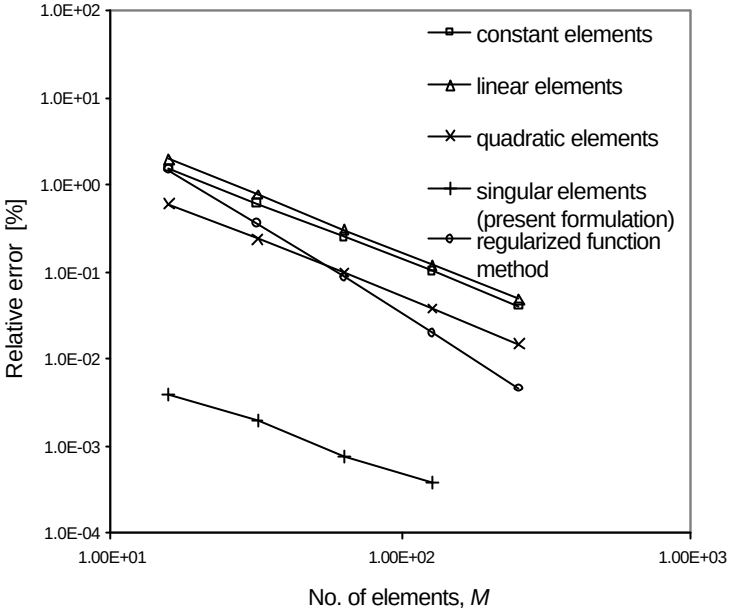


Figure 4.5. Convergence of the capacitance for coaxial conductor problem.

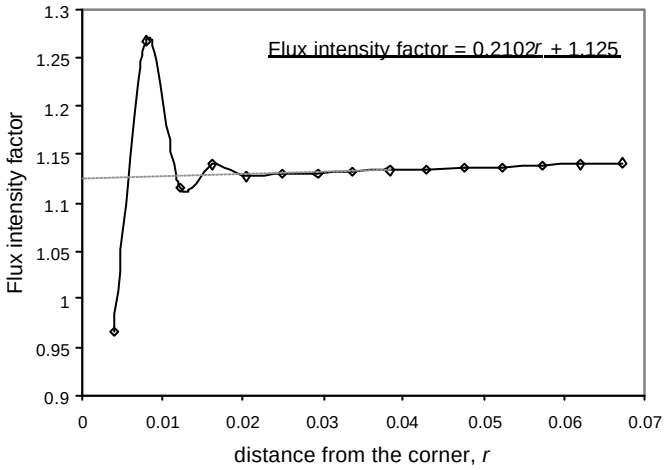


Figure 4.6. Extraction of the flux intensity factor Q_s by extrapolation method.

Table 4.1. Generalized flux intensity factor Q_s for different meshes and methods.

Number of elements, *M*	**Coefficients, Q_s**
16	1.12172
32	1.12313
64	1.12348
128	1.12363
256	1.12374
Linear Extrapolation	1.1250
256 (c.f. [25])	1.1280

Next, the effect of using different values for a and b, but retaining the singularity exponent s is examined. Two cases are considered as follows:

(1) $a = 0$ and $b = 1$, that is, $q(h) = f_1 h^{-\frac{1}{3}} + A + Bh$

$$\begin{aligned} N_1^s &= \left\{ h^{-\frac{1}{3}} - 1.51984 + 0.51984\, h \right\} L^{-\frac{1}{3}} \\ N_2^s &= 2(1-h) \\ N_3^s &= 2h - 1 \end{aligned} \tag{4.23}$$

(2) $a = 1$ and $b = 2$, that is, $q(h) = f_1 h^{-\frac{1}{3}} + Ah + Bh^2$

$$\begin{aligned} N_1^s &= \left\{ h^{-\frac{1}{3}} - 4.03968\, h + 3.03968\, h^2 \right\} L^{-\frac{1}{3}} \\ N_2^s &= 4h(1-h) \\ N_3^s &= h(2h-1) \end{aligned} \tag{4.24}$$

This study is carried out in order to determine whether the first term of the series, which contains the singularity, is the only term of importance to the computations. The two set of shape functions are given in Figures 4.7(a) and 4.7(b), respectively.

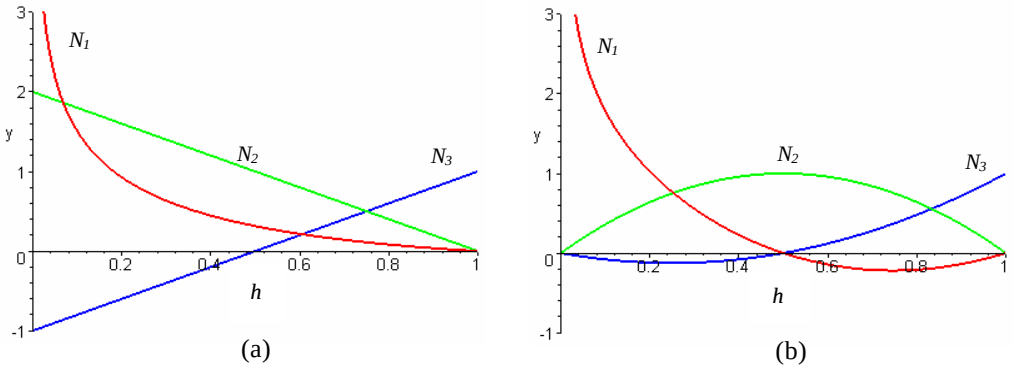


Figure 4.7. Singular shape functions for (a) s = -1/3, a = 0 and b = 1, and (b) s = -1/3, a = 1 and b = 2.

Figure 4.8 shows the surface charge density distribution on the interior conductor from point c to b in Figure 4.3, for the three sets of shape functions and with 32 elements. The three curves are observed to coincide almost exactly with one another. This suggests that the first term in the series is indeed of primary importance and is solely responsible for the vast improvement in the numerical results.

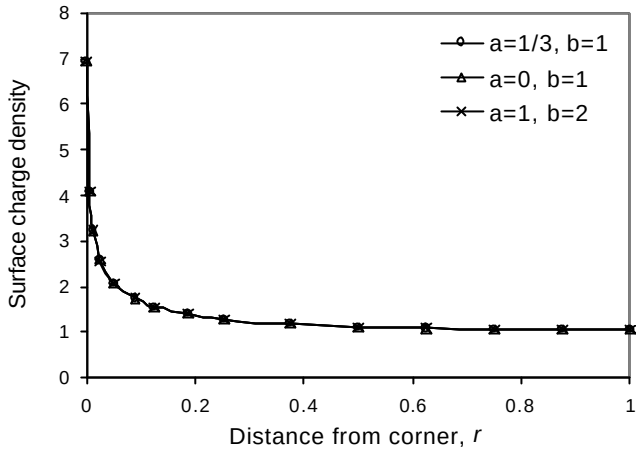


Figure 4.8. Distribution of surface charge density along interior conductor for different set of singular shape functions.

4.3.2 Parallel conductor example

Most simulations of electrostatics actuation in micro-devices are exterior problems, that is, the problem domain is infinite. Hence, the present formulation is extended to exterior problems, where the indirect BEM approach is used. This example can be viewed as two infinitely long conductors placed parallel to each other, and separated by a distance *D*. Figure 4.9 shows the cross-sectional view of the conductors. In this example, apart from the capacitance, the resultant force acting on the left conductor is also computed.

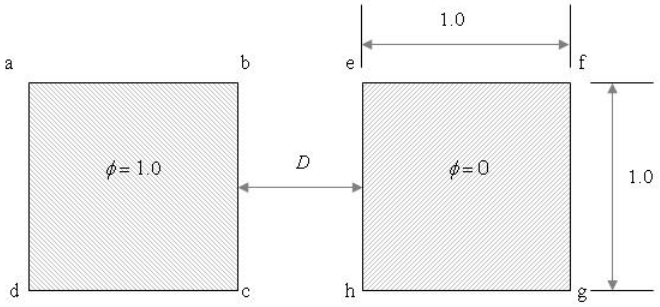


Figure 4.9. Parallel conductors with square cross-section.

Electrostatics pressure p is related to the surface charge density s by

$$p = \frac{1}{2}\frac{s^2}{e} \tag{4.25}$$

where e is the dielectric constant of the medium. The electrostatics force F acting along a straight boundary Γ of the conductor surface is hence given by

$$F = \int_{\Gamma} p \, d\Gamma \tag{4.26}$$

and it acts in the outward normal direction to the boundary.

For the present analysis, the number of elements used for each side of the conductors is denoted by M. The "exact" solution is approximated using a very fine mesh of standard quadratic elements with M=100.

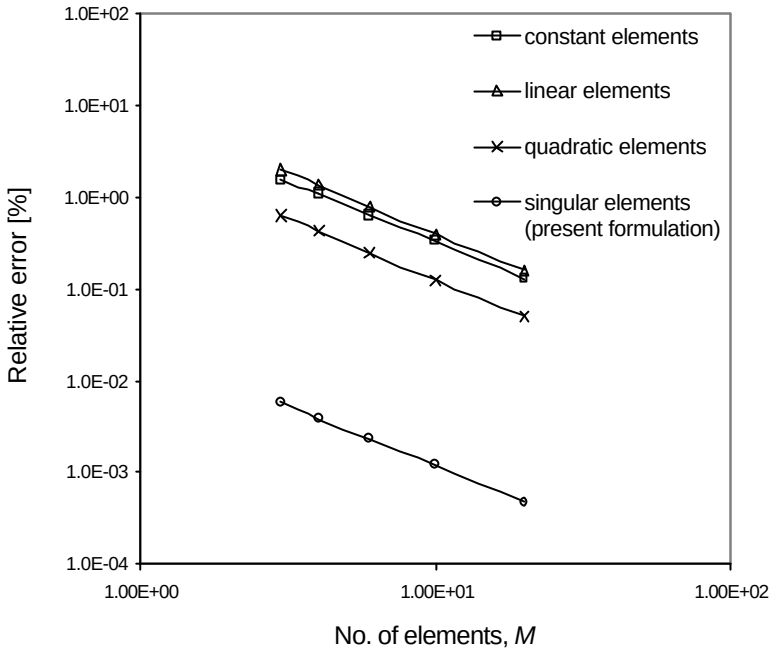


Figure 4.10. Convergence behavior of capacitance for parallel conductor problem.

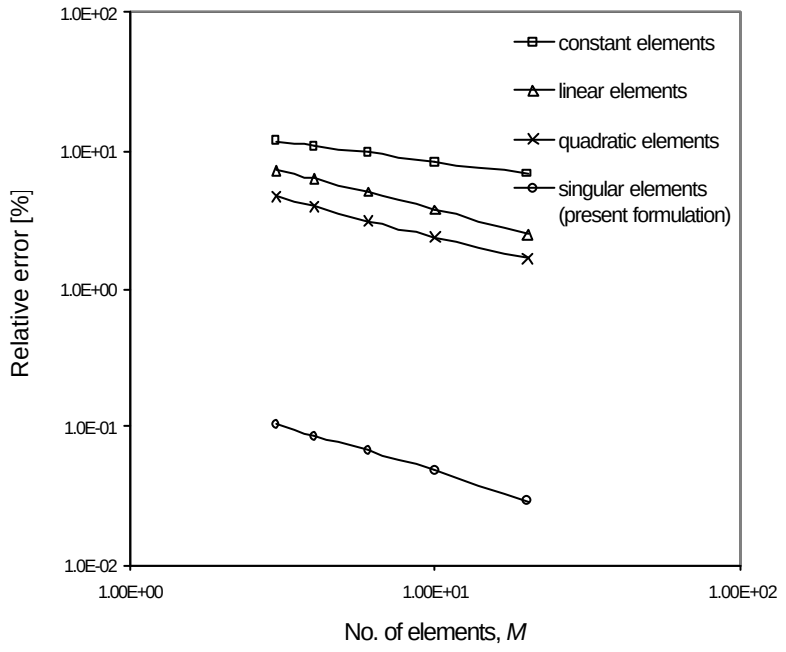


Figure 4.11. Convergence behavior of resultant force acting on the left conductor.

The convergence behaviors for the capacitance and resultant force are plotted in Figures 4.10 and 4.11, respectively, for the various types of elements under investigation. Again, the singular element formulation produces accurate results when compared to the standard elements. However, it is noted that the results for the capacitance are generally more accurate than those for the forces.

This observation can be explained as follows. Recall that the surface charge density s is the primary unknown variable in the boundary integral equation. Hence, the implementation of the BEM will minimize the error of s. Suppose the numerical solution for s is expressed as

$$s_{Numerical} = s_{Exact} + s_{Error} \tag{4.27}$$

Hence, relative error for s and the capacitance are simply given by

$$Error^{s} = \frac{s_{Error}}{s_{Exact}} \tag{4.28}$$

$$Error^{C} = \frac{\int_{\Gamma_{f=1}} s_{Error}\, d\Gamma}{\int_{\Gamma_{f=1}} s_{Exact} d\Gamma} \tag{4.29}$$

On the other hand, the electrostatics force is a quantity that is derived from s by using (4.25) and (4.26). The numerical solution for the electrostatics pressure is

$$p_{Numerical} = p_{Exact} + p_{Error} = \frac{1}{2e}\left(s_{Exact} + s_{Error}\right)^2 \tag{4.30}$$

where $p_{Exact} = \frac{1}{2e} s_{Exact}^{\ 2}$ and $p_{Error} = \frac{1}{2e}\left(2 s_{Exact} s_{Error} + s_{Error}^{\ 2}\right)$.

The relative error for the electrostatics pressure is then

$$Error^{p} = \frac{p_{Error}}{p_{Exact}} = 2\frac{s_{Error}}{s_{Exact}} + \left(\frac{s_{Error}}{s_{Exact}}\right)^2 \tag{4.31}$$

Similarly, the relative error for the electrostatics force can be expressed as

$$Error^{F} = \frac{\int_{\Gamma_{f=1}} \left(2 s_{Exact} s_{Error} + s_{Error}^{\ 2}\right) d\Gamma}{\int_{\Gamma_{f=1}} s_{Exact}^{\ 2} d\Gamma} \tag{4.32}$$

A comparison of (4.28) and (4.31) reveals that the relative error for the pressure is at least two times larger than the error for the surface charge density. Hence, in general, this means that the computed

force is expected to be less accurate than the capacitance. Furthermore, notice that s_{Num} may be greater than s_{Exact} at some parts on the boundary and smaller at others. This means that s_{Error} can be positive or negative along the whole boundary. Due to the randomness of s_{Error}, the relative error of the capacitance may still be small even when the surface charge density distribution is not correctly represented. This is probably the main reason for the standard elements, namely the constant, linear and quadratic elements, to give good results for the capacitance even though they do not capture the singularity behaviour at the corners properly. However, this is not the case for the electrostatics force. From (4.32), the error in the force comprises two components, namely $2s_{Exact}s_{Error}$ and $s_{Error}{}^2$. The first component may again be small due to randomness in s_{Error}. But the second component is positive definite (unless the exact solution is obtained), and when integrated over the boundary may not be small. Hence, in order to obtain good results for the computed force, it is important that the surface charge distribution be accurately represented in the first place, by treating the corner singularity problem.

4.3.3 Biased element distribution effect for $M = 3$

From Figure 4.8, it is noted that the singularity region is confined to a small part in the vicinity of the corner, and the remaining part of the field variable has a relatively flat distribution. This suggests that the accuracy can be improved by using an appropriately biased distribution of elements along the edge. In other words, for fixed number of element used, a small singular element is used for the singularity region, and a large quadratic element is used for the remaining part.

A parameter known as the bias ratio ***B.S*** is defined as $\boldsymbol{B.S} = \dfrac{\text{length of singular element}}{\text{length of square conductor}}$ is introduced and is varied from 1/3 (no bias for $M = 3$) to 0.1. The length of the edge of the conductor here is equal to 1 and is discretized into 3 elements. This study is carried out for different distances between the conductors, namely, D = 0.2, 0.3, 0.5, 1.0 and 2.0. The estimated "exact" solutions are obtained from the numerical results of a fine mesh (M = 100). The results are normalized with respect to the "exact" solutions, and are plotted in Figure 4.12 for different values of the distance D between the parallel conductors.

The intersections of the curves with the dashed line indicate the bias ratios that produce the most accurate results for the respective cases. Notice that only two of the curves cut the dashed line, namely for $D = 0.2$ and $D = 0.3$. This means that the optimum bias ratio is likely to vary with distance D. This behaviour can be explained as follows. First, it is noted that the sides *bc* and *eh* in Figure 4.9 resemble parallel plates. For the parallel plate, the electric field (and hence the surface charge density) is uniform in the inner portion, with some fringing effects near the edges. These fringing effects diminish as the parallel plates move closer to each other. Likely, the singular region also becomes smaller as the distance between the two parallel conductors decreases.

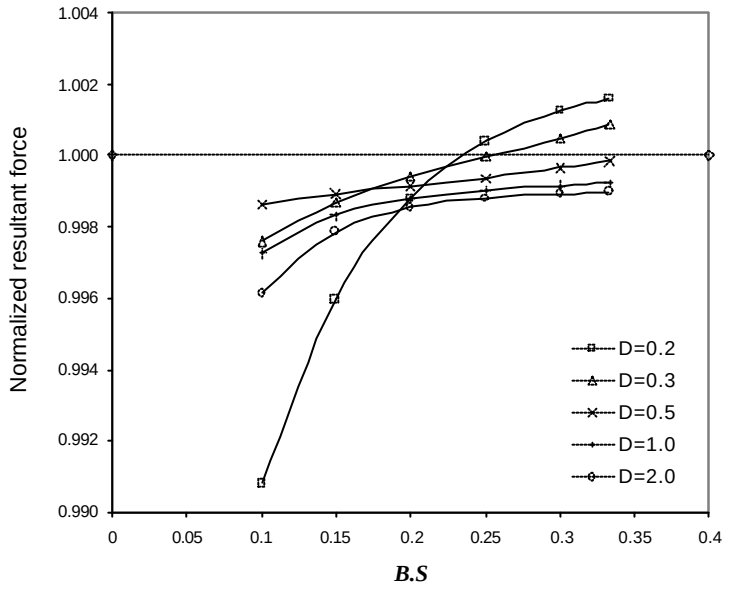


Figure 4.12. Effect of biased element distribution on accuracy of resultant force for different distances.

From Figure 4.12, it is also observed that there is no intersection for $D = 0.5$, 1.0 and 2.0. In fact for these three cases, the best results are obtained when there is no bias at all. This seems to suggest that if the distance between the conductors is greater than the nominal size of the conductors, there is no need to have a biased distribution of elements. The reason for this is that the surface charge distribution in the inner portion is no longer very flat when D is relatively large. Hence, three elements are not sufficient to approximate the actual distribution closely. In this case, better accuracy can only be obtained by using more elements. In Figure 4.13, the surface charge density distributions on the side *bc* for $D = 0.2$, 1.0 and 2.0 are plotted. The distributions are normalized with respect to the surface

charge density at the centre of *bc*. It is observed that the distributions are consistent with the earlier comments, where the inner portion becomes flatter as *D* decreases.

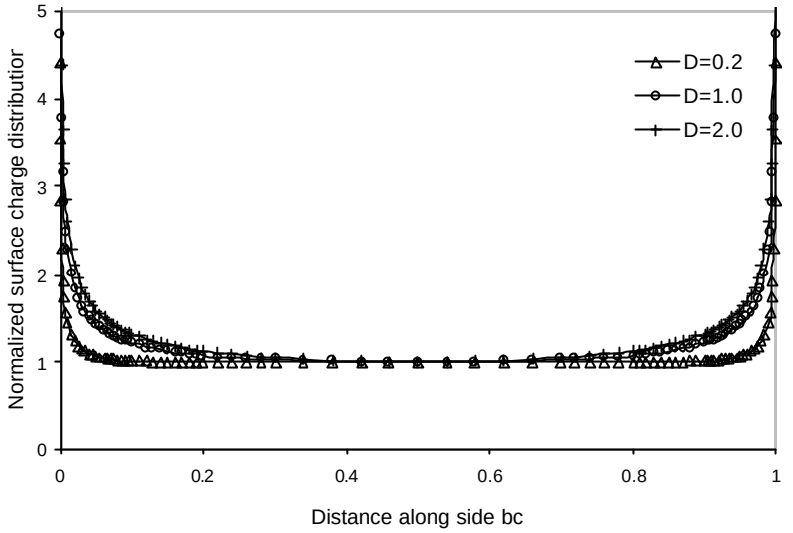


Figure 4.13. Normalized surface charge distribution on side *bc* for D = 0.2, 1.0 and 2.0.

4.4 Conclusion for Two-Dimensional Singular Elements

In this chapter, a singular element approach has been presented for the analysis of corner singularities in two-dimensional potential problems. The shape functions of the singular element are formulated to incorporate the singular behaviour of the normal potential gradient. This method requires only a minor modification in the formulation of the boundary element equations, by using the singular shape functions in (4.8), instead of the standard quadratic shape functions, when either node 1 or 3 of the element coincides with a corner.

This method has been applied to two numerical examples, namely the coaxial conductor example (interior problem), and the parallel conductor example (exterior problem). The results are very accurate in comparison with the standard elements, namely, constant, linear and quadratic elements. Furthermore, for the first example, this method also shows better accuracy over the 'regularized function method' by Igarashi and Honma [25]. In conclusion, this method is capable of producing accurate results, in terms of the capacitance, force per unit length and also the generalized flux intensity factor, even for coarse meshes. Furthermore, the generalized flux intensity factor, Q_s of (4.2), is computed directly in the analysis with only little extra effort needed.

However, the evaluation of the boundary integral for the singular element is more complicated. The computational effort for this singular element integral is estimated to be four to five times more than that required by the standard quadratic element. However, this extra effort can be easily compensated by the reduction in the total number of elements used, since for a given accuracy, this method requires much fewer elements than when using standard elements.

5

Three-Dimensional Singular Elements

The two-dimensional results motivate us to extend the singular element method to three-dimensional analysis. However, it is shown in here that this extension is not trivial. The complication arises due to the additional dimension, where two-dimensional corners (represented as points) now become edges in the three-dimensional context, and in addition, there are three-dimensional corners that are formed when the edges meet or terminate.

This chapter is organized as follows. Section 5.1 covers the aspects of identifying the singular features that can exist in a 'rectangular' structure, and defining the different types of singular elements needed. Unlike in the two-dimensional case, the singular fields for these three-dimensional corners cannot be expressed in closed forms. Numerical methods have to be employed to approximate the singular solutions, which are presented in Section 5.2. In Section 5.3, we present the core of this chapter, that is concerned with the formulation of the singular shape functions. A general methodology for formulating arbitrary singular element is first described, followed by the specific implementation for the various singular elements that are identified in Section 5.1. The numerical integration of the boundary integrals, where two sources of singularities can exist is also discussed. The singular elements are used to solve some numerical examples to evaluate the performance of these singular elements. It is shown that they can improve the accuracy of the results for capacitance and electrostatics forces quite significantly. We also investigate the effects of using the singular elements on the functionality of some MEMS devices. Finally, some concluding remarks are provided.

5.1 Identifying Singular Features

This section describes the identification of the different types of singular elements that exist in a "rectangular" structure. The first part identifies the singularity features in the structure, namely the edges and corners, and based on the different combinations of the singular features, the various types of singular element are defined.

5.1.1 Identify singular edges and corners

In this study, we confine ourselves to structures that are 'rectangular', that is, the edges and corners are right-angled. This is an important special case because many of the MEMS structures are generally 'rectangular'. Figure 5.1 shows a typical 'rectangular' structure, where different types of edges and corners are identified. The aim here is to determine the singular features in the given structure.

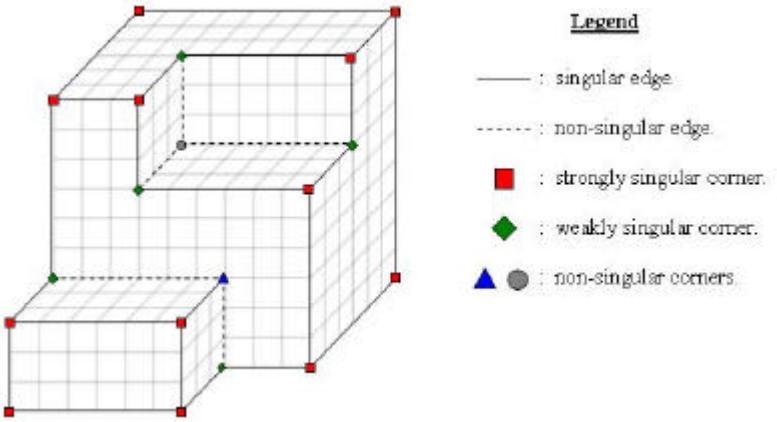


Figure 5.1. A "rectangular" structure with identified edges and corners.

To determine whether an edge or a corner is singular, we usually have to solve for the eigenvalue of that geometry. In general, only the smallest eigenvalue $\alpha_{\min}$ is of interest, because $\lambda_{\min} = (1 - \alpha_{\min})$ corresponds to the order of singularity for the potential gradients and surface charge density, in the vicinity of the edges and corners, when $\alpha_{\min} < 1$. The two-dimensional results of the singularity field analysis (see Appendix B.1) can be used directly for the re-entrant edges (continuous solid lines in Figure 5.1).

For three-dimensional corners, the solid angle γ is a good indication of the singularity nature of the corner, where it is expected to be singular if $\gamma < 2\pi$ ($\gamma = 2\pi$ corresponds to a smooth surface). Based on this simple observation, two singular corners are identified in Figure 5.1, namely the strongly and weakly singular corners, denoted by the squares and diamonds markers in Figure 5.1. In contrast, the corners that are marked with triangles and circles are non-singular. In fact, they should be more appropriately identified as 'zero' corners, because theoretically no charge can exist at these locations. However in this study, only the singular features are specially treated because they are possibly the main source of errors in the electrostatics analysis.

5.1.2 Identify possible types of singular elements

Based on the geometry in Figure 5.1, five different types of singular elements are defined, as illustrated in Figure 5.2.

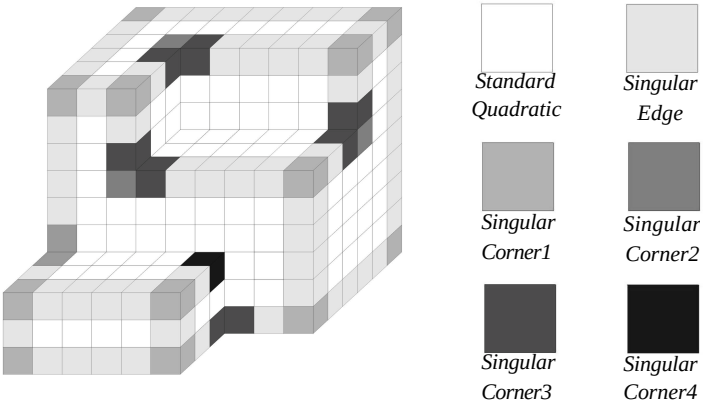


Figure 5.2. Boundary element mesh of "rectangular" structure with various types of singular elements.

The different types of singular elements arise because of the way in which the singularity fields vary on the elements. This is in turn dependent on the numbers and types of edges and corners that fall on them. Hence, this provides a unique way of identifying the various types singular elements. The definitions for the singular elements are as follows:

(1) *Singular Edge*: Contains only one singular edge. The order of singularity (edge singularity) remains the same along this edge.

(2) *Singular Corner1*: Contains the strongly singular corner with the two adjacent singular edges. The singularity order increases along the edge towards the singular corner.

(3) *Singular Corner2*: Contains only the weakly singular corner, and the field is only weakly singular at the corner.

(4) *Singular Corner3*: Contains only one singular edge and the weakly singular corner. In this case, the singularity order varies along the edge towards the singular corner.

(5) *Singular Corner4*: Contains only one singular edge and a non-singular corner. In this case, the singular field decays along the edge towards the non-singular corner.

After identifying these five types of singular elements, the next task is to formulate the singular shape functions for these elements. But prior to that, the order of singularities for the singular edges and corners must be obtained.

5.2 Extraction of the Order of Singularities

In Section 5.1.1, one singular edge (re-entrant edges) and two singular corners (the strongly and weakly singular corners) were identified. In this section, we will determine the order of singularities for these singular features.

5.2.1 Singular edge

The two-dimensional corner actually corresponds to the 'plane-strain' (dimension is infinite in the out-of-plane direction) approximation of the three-dimensional edge. Hence, the results from the two-dimensional singular field analysis can be used directly for the three-dimensional singular edges. From the two-dimensional study, the order of singularity for the right-angled singular edges is $l_E = \frac{1}{3}$.

5.2.2 Strongly singular corner

This type of corner is formed when three singular edges meet. Figure 5.3 shows the geometry of this corner and its problem domain *G* for the eigen-problem.

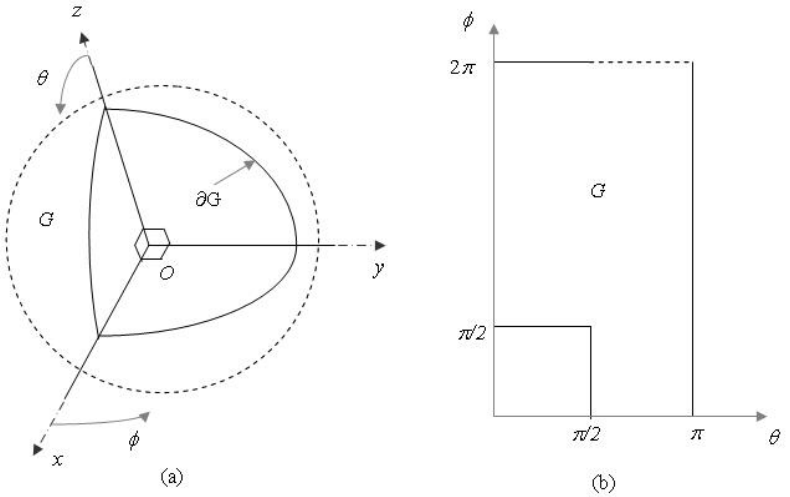


Figure 5.3. (a) Geometry of strongly singular corner, (b) Plot of eigen-problem domain in *(q, f)* plane.

It is noted that this type of corner had previously been studied extensively by Fichera [34], Beagles and Whiteman [35], and Bazant [36]. Fichera obtained a lower bound for the order of singular for the potential field as

$$a_{\min} > 0.4335 \tag{5.1}$$

On the other hand, Bazant reported a value of $a_{\min} = 0.455$, using the finite difference approach. Finally, Beagles and Whiteman summarized their results as follows

$$a^*_{\min} \approx 0.4525, \quad \text{with } 0.4335 < a_{\min} < 0.4542. \tag{5.2}$$

where $a^*_{\min}$ is the 'exact' solution suggested, which was extrapolated from the results obtained with various mesh sizes. In this study, the suggested value $a^*_{\min}$ is used, which gives the order of singularity for this particular corner to be $l_{C1} = 0.5475$.

5.2.3 Weakly singular corner

This weakly singular corner corresponds to the diamond markers in Figure 5.1. The eigen-problem domain *G*, together with the boundary conditions, is depicted in Figure 5.4(b), and only half of it is required due to symmetry.

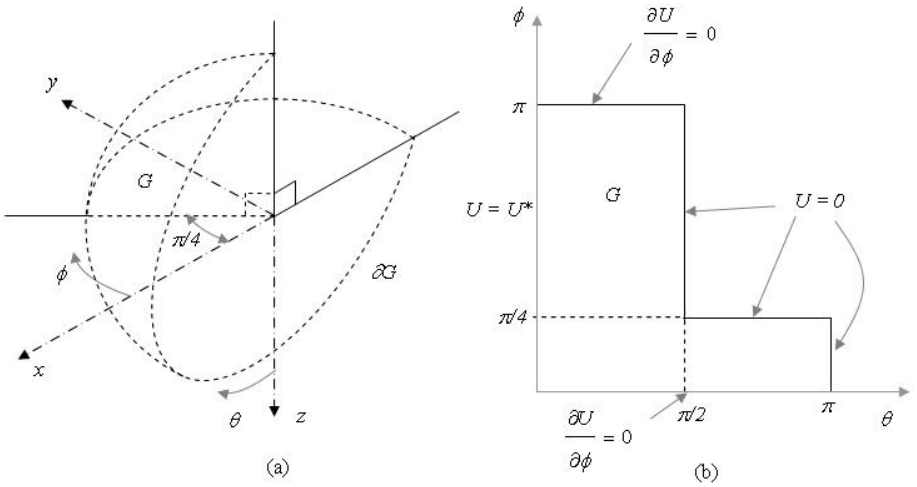


Figure 5.4. (a) Geometry of weakly singular corner, (b) Plot of eigen-problem domain in (q, f) plane.

The boundary conditions imposed at f = 0 and p are due to the symmetry condition, and for those boundaries that fall on the conductor surfaces, U = 0. However, the boundary condition imposed at q = 0 is not obvious. First, this boundary is a fictitious one, which actually corresponds to a point in the original spherical coordinates system. Hence, it is expected that U is single-valued along this boundary, that is, U=U* is itself an unknown. In this case, the problem is not well-posed since the given boundary value problem (BVP) is not fully constrained. However, it is observed that $\frac{\partial U}{\partial q} = 0$ at

the point $(q, f) = (0, p/2)$ due to the symmetry condition. This supplements the extra condition needed to define the problem completely.

The governing equation for the three-dimensional eigen-problem of the potential analysis is (see Appendix B.2 for the derivation)

$$\begin{aligned} [\Delta_q + a(a+1)\boldsymbol{I}]U = 0, \quad & (q,f) \in G \\ U = 0, \quad & (q,f) \in \partial G_D \\ \frac{\partial U}{\partial n} = 0, \quad & (q,f) \in \partial G_N \end{aligned} \tag{5.3}$$

where $a(a+1)$ is the eigenvalue for the given problem, $\boldsymbol{I}$ is the identity operator and Δ_θ is the Laplace-Beltrami operator defined as

$$\Delta_q = \frac{1}{\sin q}\frac{\partial}{\partial q}\left(\sin q \frac{\partial}{\partial q}\right) + \frac{1}{\sin^2 q}\frac{\partial^2}{\partial f^2} \tag{5.4}$$

The objective here is to determine the smallest eigenvalue $a_{\min}(a_{\min}+1)$ that satisfies (5.3). Due to the simple geometry of the domain G, the finite difference method is used to solve this eigenvalue problem. However, the variable U and more importantly its derivatives are not well behaved near the point $\left(\frac{p}{2}, \frac{p}{4}\right)$, because of the singularity ray that passes through it. The order of singularity for this ray is identical to that of the singular edge, which is $a_E = \frac{2}{3}$. This means that both the first and second order partial derivatives of U are singular at that point, and finite difference method fails to give accurate results when used directly to solve (5.3). To overcome this problem, the regularization technique suggested by Bazant [36] is used, which assumes

$$U(q,f) = t^{a_E} u(q,f) \tag{5.5}$$

where $t = \sqrt{\left(q - \frac{p}{2}\right)^2 + \left(f - \frac{p}{4}\right)^2}$ corresponds to the radial distance from the singularity point. Hence, the point singularity of the original variable U is explicitly extracted by the transformation in (5.5), so that the new variable u is nonsingular throughout the problem domain G. Substituting (5.5) into (5.3) and (5.4) gives

$$F_1 u + F_2 \frac{\partial u}{\partial q} + F_3 \frac{\partial u}{\partial f} + F_4 \frac{\partial^2 u}{\partial q^2} + F_5 \frac{\partial^2 u}{\partial f^2} = 0 \tag{5.6}$$

where

$$F_1 = a(1+a)t^2\sin^2 q + a_E(a_E - 1)\left[\sin^2 q\left(\frac{\partial t}{\partial q}\right)^2 + \left(\frac{\partial t}{\partial f}\right)^2\right]$$
$$+ a_E t\left[\sin^2 q\,\frac{\partial^2 t}{\partial q^2} + \frac{\partial^2 t}{\partial f^2} + \frac{\sin 2q}{2}\frac{\partial t}{\partial q}\right]$$
$$F_2 = 2a_E t\sin^2 q\,\frac{\partial t}{\partial q} + t^2\frac{\sin 2q}{2}$$
$$F_3 = 2a_E t\,\frac{\partial t}{\partial f}$$
$$F_4 = t^2\sin^2 q$$
$$F_5 = t^2$$

and $t = \sqrt{\left(q - \frac{p}{2}\right)^2 + \left(f - \frac{p}{4}\right)^2}$, $\frac{\partial t}{\partial q} = \frac{1}{t}\left(q - \frac{p}{2}\right)$, $\frac{\partial t}{\partial f} = \frac{1}{t}\left(f - \frac{p}{4}\right)$, $\frac{\partial^2 t}{\partial q^2} = \frac{1}{t} - \frac{1}{t^3}\left(q - \frac{p}{2}\right)^2$ and

$\frac{\partial^2 t}{\partial f^2} = \frac{1}{t} - \frac{1}{t^3}\left(f - \frac{p}{4}\right)^2$.

The finite difference method is now used to solve for the eigenvalues of the modified eigen-problem. The discretized finite difference domain of eigen-problem and the modified boundary conditions are depicted in Figure 5.5.

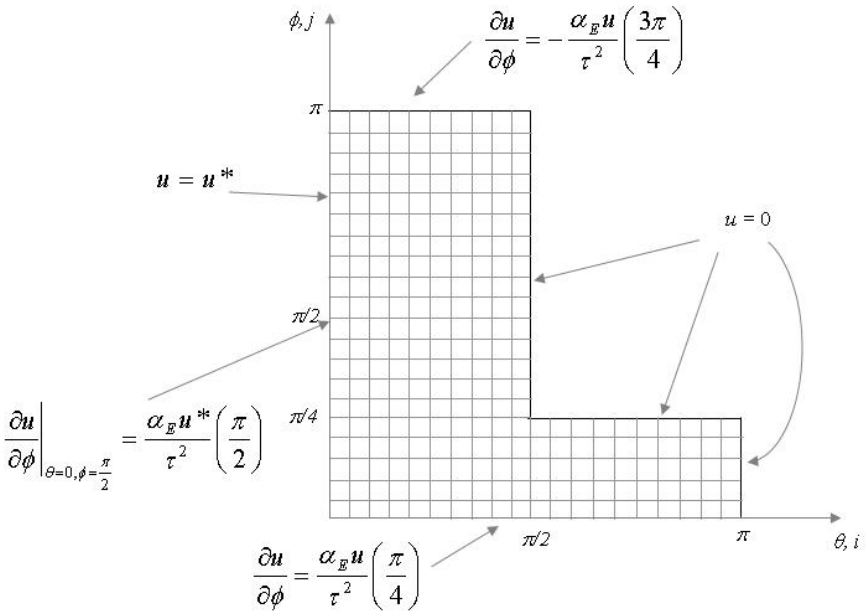


Figure 5.5. Discretized finite difference domain and boundary conditions for the modified eigen-problem.

The Neumann boundary conditions are modified using (5.5), that is,

$$\frac{\partial U}{\partial f}=0 \quad \Rightarrow \frac{\partial u}{\partial f}=-\frac{a_E u}{t^2}\left(f-\frac{p}{4}\right) \tag{5.7a}$$

$$\frac{\partial U}{\partial q}=0 \quad \Rightarrow \frac{\partial u}{\partial q}=-\frac{a_E u}{t^2}\left(q-\frac{p}{2}\right) \tag{5.7b}$$

Finally, applying the finite difference method leads to the linear system of equations,

$$\boldsymbol{M}(u)\boldsymbol{u}=\boldsymbol{0} \tag{5.8}$$

where $\boldsymbol{M}$ is the eigen-matrix with $u = a(1+a)$, and $\boldsymbol{u}$ is the vector of the nodal unknowns. A non-trivial solution of $\boldsymbol{u}$ exists if and only if the determinant of $\boldsymbol{M}(u)$ is zero. In Appendix B.2, two methods of solving (5.8) are presented. For this problem, method B (conversion to non-homogenous equations) is used, because small scanning interval for a can be obtained using the results from Bazant [36]. It is noted that some of the cases were already studied by Bazant (data points in Figure 5.6(b)). By using these known solutions and fitting the data with a best-fit curve (as depicted in Figure 5.6(b)) an estimate of the eigenvalue is derived as $a_{estimate} \approx 0.871$.

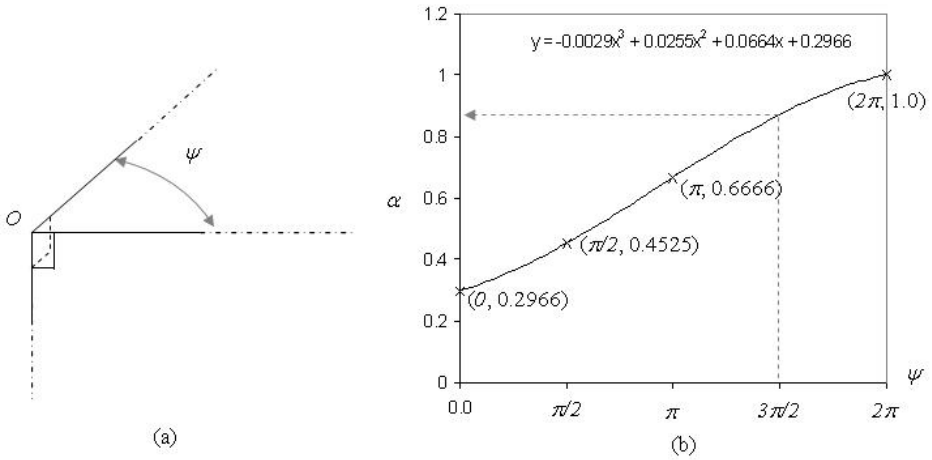


Figure 5.6. (a) A general right-angled corner with varying y angle. (b) Plot of the eigen-values a_{min} versus different y.

Finally, the results for the smallest eigenvalue using different meshes are plotted in Figure 5.7, and by extrapolating these results gives $a^* = 0.8896$, which gives the order of singularity for this weakly singular corner to be $l_{C2} = 0.1104$.

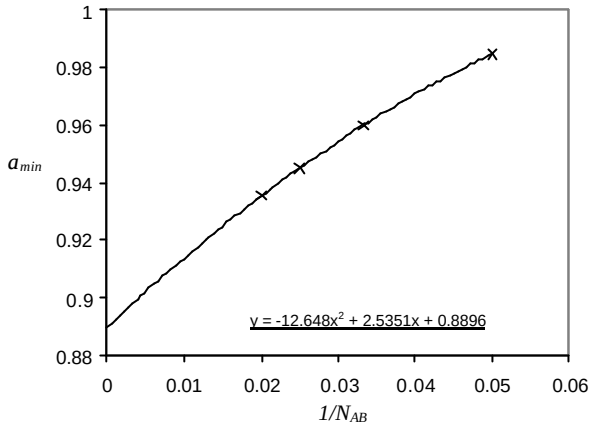


Figure 5.7. Extraction of singularity order for weakly singular corner.

5.3 Formulation of Three-Dimensional Singular Elements

In this section, we present the formulations of the singular shape functions for the 9-node and 8-node singular elements. The 8-node singular elements are also derived in consideration of the serendipity element. It is well known that the serendipity element can be as accurate as the 9-noded Lagrangian element, even though it does not satisfy the complete quadratic form.

5.3.1 General methodology for formulating singular elements

A general methodology for formulating singular elements, with arbitrary order of singularities, is first presented. This approach consists of the following three steps.

Step 1: Approximating the singular solution

The first step is to approximate the variation of the potential flux on the singular elements, which comprises singular and nonsingular parts. Generally, it can be expressed in the following form:

$$f(r_1, r_2) = \sum_{i=1}^{n_s} c_i^s f_i^s(r_1, r_2) + \sum_{j=1}^{n_{ns}} c_j^{ns} f_j^{ns}(r_1, r_2) \tag{5.9}$$

where $f_i^s(r_1, r_2)$ and $f_j^{ns}(r_1, r_2)$ are the singular and nonsingular functions respectively, and c_i^s and c_j^{ns} are the unknown coefficients. Here, r_1 and r_2 correspond to the intrinsic coordinates of the element. The number of terms for the two parts depends on the number of nodes that are located directly on the singularity, but they must satisfy the equality, $n_s + n_{ns} = n$, where n is the number of nodes on the element, which is 8 or 9. Two requirements should satisfy by (5.9), and they are:

(i) The singular functions should closely approximate the actual singular fields. This requires the orders of singularities to be correctly included at the appropriate singular nodes. Although, it seems desirable to have the singularity variations (the eigenvectors) incorporated in the singular functions, they generally violate the second requirement, which is the compatibility condition.

(ii) The singular elements should be compatible, that is, the field variable must be continuous at inter-element boundaries. This is necessary because the surface charge density or the electric flux is expected to be continuous on smooth surfaces. However, from the mathematical viewpoint, this compatibility condition need not be enforced in the BEM, as for example when using the constant boundary elements.

This step is the most difficult part of the process. The two requirements stated above complicate the task of finding appropriate functions for (5.9). But once the functions are formed, the rest of the steps follow naturally.

Step 2: Solving for the singular coefficients, c_i^s

The next task is to solve for the coefficients in terms of the nodal unknowns. The singular coefficients are solved first, and this is done by applying the nodal conditions at the singular nodes. In the vicinity of the singular region, the nonsingular part has negligible effect, and hence (5.9) is reduced to

$$f(r_1, r_2) \approx \sum_{i=1}^{n_s} c_i^s g_i^s(r_1, r_2) h_i(r_1, r_2) \tag{5.10}$$

where $f_i^s(r_1, r_2)$ is conveniently expressed as $g_i^s(r_1, r_2) h_i(r_1, r_2)$, with $g_i^s(r_1, r_2)$ being a non-singular function, and $h_i(r_1, r_2)$ is the singularity form that exists in $f_i^s(r_1, r_2)$. Then, dividing (5.10) with an appropriate de-singularizing term $h_j(r_1, r_2)$ gives

$$\frac{f(r_1, r_2)}{h_j(r_1, r_2)} = c_j^s g_j^s(r_1, r_2) + \sum_{i=1, i\neq j}^{n_s} c_i^s \frac{g_i^s(r_1, r_2) h_i(r_1, r_2)}{h_j(r_1, r_2)} \tag{5.11}$$

Note that (5.11) gives the generalized flux intensity factor j_j^s at the appropriate singular node *j*, which is related to the coefficients c_i^s. By considering the flux intensity factors at all the singular nodes, exactly n_s equations are generated, which allows c_i^s be solved uniquely in terms of j_i^s, for $i = 1,..,n_s$.

Finally, by substituting c_i^s back into (5.10), and gathering the terms with identical j_i^s together, the singular functions in (5.10) can be rewritten as

$$f^s(r_1, r_2) = \sum_{i=1}^{n_s} j_i^s \, p_i^s(r_1, r_2) \tag{5.12}$$

where $p_i^s(r_1, r_2)$ forms the singular part of the shape functions of the singular element.

Step 3: Solving for the nonsingular coefficients, c_i^{ns}

Next, (5.12) is substituted back into the (5.9) to give

$$f(r_1, r_2) = \sum_{i=1}^{n_s} j_i^s \, p_i^s(r_1, r_2) + \sum_{j=1}^{n_{ns}} c_j^{ns} \, f_j^{ns}(r_1, r_2) \tag{5.13}$$

Here, we aim to solve for the nonsingular coefficients c_j^{ns}. The procedure is exactly the same as in *Step 2*, except the nodal values at the nonsingular nodes j_j^{ns} correspond to the actual physical quantities. However, in this case, c_j^{ns} would be functions of j_j^{ns} and j_i^s, since the singular part may be nonzero at the nonsingular nodes. Again, a sufficient number of equations is available to solve for the coefficients c_j^{ns} uniquely. Finally, substituting the results back into (5.9) gives the complete set of shape functions for the singular element,

$$f(r_1, r_2) = \sum_{i=1}^{n_s} j_i^s \left[p_i^s(r_1, r_2) + q_i^{ns}(r_1, r_2) \right] + \sum_{j=1}^{n_{ns}} j_j^{ns} \, q_j^{ns}(r_1, r_2) \tag{5.14}$$

Here the terms in the square brackets are the singular shape functions for the singular nodes, which comprise singular $p_i^s(r_1, r_2)$ and nonsingular $q_i^{ns}(r_1, r_2)$ parts, and $q_j^{ns}(r_1, r_2)$ are the shape functions for the nonsingular nodes. Note that *Step* 2 and *3* can be incorporated into commercial software with symbolic computation abilities, such as *Mathematica* and *Maple*.

5.3.2 Formulating the singular elements

The three-step process described above is used to formulate the shape functions for the various singular elements identified in Section 5.1. However, only the first step is presented here for all the singular elements, because it is the vital step that determines the shape functions. Note also that the expressions

presented in this section are for the 9-node elements only. Those for the 8-node elements can be derived similarly by dropping the highest order term in the nonsingular part of the expressions.

Edge singular element

Without loss in generality, suppose the element is singular along the edge $r_2 = 0$, as shown in Figure 5.8. One possible singularity solution is of the following form

$$f(r_1, r_2) = \frac{g^s(r_1, r_2)}{r_2^{l_E}} + f^{ns}(r_1, r_2) \tag{5.15}$$

where

$$g^s(r_1, r_2) = c_1^s + c_2^s r_1 + c_3^s r_1^2,$$
$$f^{ns}(r_1, r_2) = c_1^{ns} r_2 + c_2^{ns} r_1 r_2 + c_3^{ns} r_2^2 + c_4^{ns} r_1 r_2^2 + c_5^{ns} r_1^2 r_2 + c_6^{ns} r_1^2 r_2^2$$

and $l_E = \frac{1}{3}$.

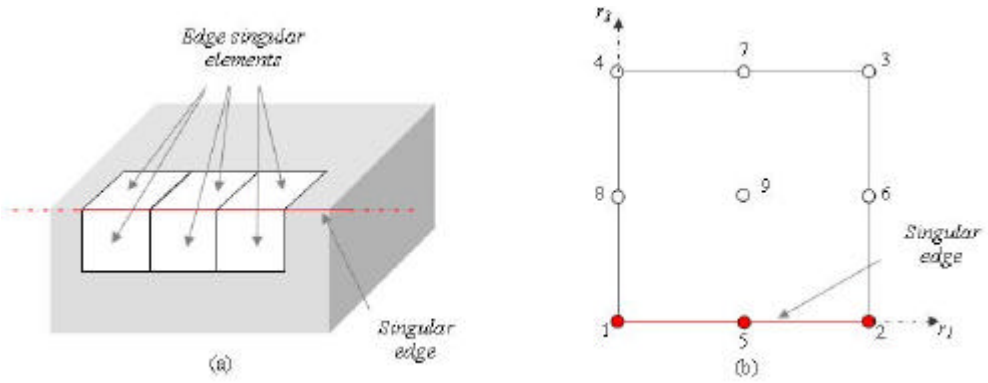


Figure 5.8. (a) Locations of *Edge* singular elements, and (b) *Edge* singular element definitions.

Basically in (5.15), the generalized flux intensity factor j^s assumes a quadratic variation along the singular edge. The nonsingular function also assumes a quadratic form, in which the terms are selected so that compatibility conditions along the element boundaries are satisfied. At the inter-element boundaries, the solution must be of the following forms:

(iii) quadratic along the edge $r_2 = 1$, that is,

$$f(r_1) = a_0 + a_1 r_1 + a_2 r_1^2 \tag{5.16a}$$

(iv) two-dimensional singular form at the edges $r_1 = 0$ and $r_1 = 1$, that is,

$$f(r_2) = \frac{b_0}{r_2^{l_E}} + b_1 r_2 + b_2 r_2^2 \text{†} \tag{5.16b}$$

where *a's* and *b's* are constant coefficients.

† This two-dimensional singularity form is chosen because it blends naturally with the standard quadratic function. Furthermore, the two-dimensional analysis shows that this singularity form can be as accurate as the actual singularity form, as given in Section 4.1.1.

To check for the compatibility conditions: When $r_2 = 1$, (5.15) becomes

$$f^{ns}(r_1) = \left(c_1^s + c_1^{ns} + c_3^{ns}\right) + \left(c_2^s + c_2^{ns} + c_4^{ns}\right) r_1 + \left(c_3^s + c_5^{ns} + c_6^{ns}\right) r_1^2 \tag{5.17}$$

which is a quadratic form. When $r_1 = 0$ and $r_1 = 1$, (5.15) are reduced, respectively, to

$$f(r_2) = \frac{c_1^s}{r_2^{l_E}} + c_1^{ns} r_2 + c_3^{ns} r_2^2 \tag{5.18a}$$

$$f(r_2) = \frac{c_1^s + c_2^s + c_3^s}{r_2^{l_E}} + \left(c_1^{ns} + c_2^{ns} + c_5^{ns}\right) r_2 + \left(c_3^{ns} + c_4^{ns} + c_6^{ns}\right) r_2^2 \tag{5.18b}$$

which also satisfy the required singularity form.

Up to this point, it is shown that the singular solution in (5.15) is feasible. Proceeding with *Step 2* and *3* leads to the sets of shape functions for the *Edge* singular element.

<u>9-node element:</u>

$$\begin{aligned}
N_1 &= -\frac{(2r_1 - 1)(r_1 - 1)}{r_2^{l_E}} + (2r_1 - 1)(1 - r_1) r_2 \left[(2r_2 - 1) + 4a(1 - r_2)\right] \\
N_2 &= \frac{(2r_1 - 1) r_1}{r_2^{l_E}} - (2r_1 - 1) r_1 r_2 \left[(2r_2 - 1) + 4a(1 - r_2)\right] \\
N_3 &= (2r_1 - 1) r_1 (2r_2 - 1) r_2 \\
N_4 &= -(2r_1 - 1)(1 - r_1)(2r_2 - 1) r_2 \\
N_5 &= \frac{4(1 - r_1) r_1}{r_2^{l_E}} - 4(1 - r_1) r_1 r_2 \left[(2r_2 - 1) + 4a(1 - r_2)\right] \\
N_6 &= 4(2r_1 - 1) r_1 (1 - r_2) r_2 \\
N_7 &= 4(1 - r_1) r_1 (2r_2 - 1) r_2 \\
N_8 &= -4(2r_1 - 1)(1 - r_1)(1 - r_2) r_2 \\
N_9 &= 16(1 - r_1) r_1 (1 - r_2) r_2
\end{aligned} \tag{5.19a}$$

<u>8-node element:</u>

$$\begin{aligned}
N_1 &= -\frac{(2r_1 - 1)(r_1 - 1)}{r_2^{l_E}} + (1 - r_1) r_2 \left[(2r_1 - 1) + 2(1 - r_2)(1 - 2a)\right] \\
N_2 &= \frac{(2r_1 - 1) r_1}{r_2^{l_E}} - r_1 r_2 \left[(2r_1 - 1) + 2(1 - r_2)(1 - 2a)\right] \\
N_3 &= r_1 r_2 \left[(2r_1 - 1) - 2(1 - r_2)\right] \\
N_4 &= -(1 - r_1) r_2 \left[(2r_1 - 1) + 2(1 - r_2)\right] \\
N_5 &= \frac{4(1 - r_1) r_1}{r_2^{l_E}} - 4(1 - r_1) r_1 r_2 \\
N_6 &= 4 r_1 (1 - r_2) r_2 \\
N_7 &= 4(1 - r_1) r_1 r_2 \\
N_8 &= 4(1 - r_1)(1 - r_2) r_2
\end{aligned} \tag{5.19b}$$

where $a = (2)^{l_E}$.

Corner1 singular element

This element contains two singular edges and a strongly singular corner located at node 1, as shown in Figure 5.9(b).

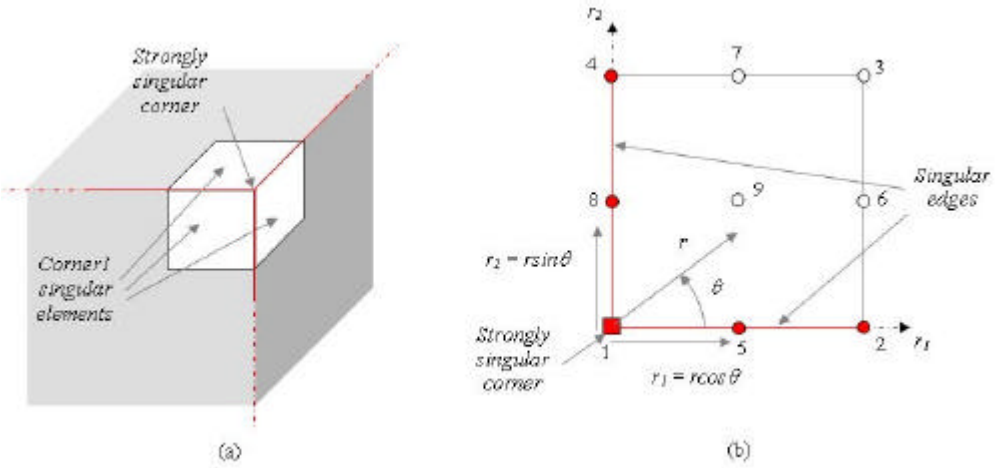


Figure 5.9. (a) Locations of *Corner1* singular elements, and (b) *Corner1* singular element definitions.

In this case, the singular part of (5.9) contains one strongly singular corner and two edges singularity forms. Considering the corner singularity, in the vicinity of the corner node, the corner singularity field is dominant, and has the general form

$$f^s\Big|_{r\to 0} \approx c_1^s r^{-l_{C1}} f(q) \tag{5.20}$$

Here c_1^s is the generalized flux intensity factor, $r=\sqrt{r_1^2+r_2^2}$ is the radial distance from the corner node 1, and $f(q)$ describes the variation of the singular field on the element, which is essentially the solution of the eigenproblem for the strongly singular corner. Then it is also noted that when $q\to 0$ and $\frac{p}{2}$ with $r\neq 0$, the field is also singular, which is of the edge singularity form. Hence, by extracting these edge singularities from $f(q)$, (5.20) gives

$$f(q)=(\sin q)^{-l_E}(\cos q)^{-l_E} g(q) \tag{5.21}$$

where $(\sin q)^{-l_E}$ and $(\cos q)^{-l_E}$ account for the edge singularities along $r_2=0$ and $r_1=0$, respectively, and $g(q)$ is expected to be a nonsingular function. Substituting (5.21) back into (5.20), and also recognizing that $\sin q=\frac{r_2}{r}$ and $\cos q=\frac{r_1}{r}$, gives

$$f^s\Big|_{r\to 0} \approx c_1^s r^{(-l_{C1}+2l_E)}\left(\frac{1}{r_2}\right)^{l_E}\left(\frac{1}{r_1}\right)^{l_E} g(q) \tag{5.22}$$

Finally, to ensure that the compatibility requirements are satisfied along $r_1 = 1$ and $r_2 = 1$, (5.22) is forced to be zero along these two edges by assuming the simple form of $g(q) = (1-r_1)(1-r_2)$. Hence, (5.22) becomes

$$f^s\Big|_{r\to 0} = c_1^s \frac{r^{g_1}(1-r_1)(1-r_2)}{r_1^{l_E} r_2^{l_E}} \tag{5.23}$$

where $g_1 = -l_{C1} + 2l_E = 0.1192$.

The two edge singularity forms follow that given in (5.15) closely, where the generalized flux intensity varies quadratically along the two singular edges. However, it is desirable that these edge singularity fields vanish in the vicinity of the corner node, because these effects are already included in the corner singularity given in (5.23). Hence, the appropriate choices for the edge singularity forms are

$$f^s\Big|_{r_1\to 0} = \frac{\left(c_2^s + c_3^s r_2\right) r_2}{r_1^{l_E}} \tag{5.24a}$$

$$f^s\Big|_{r_2\to 0} = \frac{\left(c_4^s + c_5^s r_1\right) r_1}{r_2^{l_E}} \tag{5.24b}$$

As for the nonsingular function, it turns out that the only choice that will enforce the compatibility conditions at the inter-element boundaries is

$$f^{ns}(r_1, r_2) = c_1^{ns} r_1 r_2 + c_2^{ns} r_1 r_2^2 + c_3^{ns} r_1^2 r_2 + c_4^{ns} r_1^2 r_2^2 \tag{5.25}$$

Therefore, the complete singularity representation for this *Corner1* singular element is

$$\begin{aligned} f(r_1, r_2) = c_1^s \frac{r^{g_1}(1-r_1)(1-r_2)}{r_1^{l_E} r_2^{l_E}} + \frac{\left(c_2^s + c_3^s r_2\right) r_2}{r_1^{l_E}} + \frac{\left(c_4^s + c_5^s r_1\right) r_1}{r_2^{l_E}} \\ + c_1^{ns} r_1 r_2 + c_2^{ns} r_1 r_2^2 + c_3^{ns} r_1^2 r_2 + c_4^{ns} r_1^2 r_2^2 \end{aligned} \tag{5.26}$$

The singular expression satisfies the compatibility conditions at $r_1 = 1$ and $r_2 = 1$, since the solutions along the two edges are

$$f(r_1, r_2) = \frac{\left(c_4^s + c_5^s\right)}{r_2^{l_E}} + \left(c_2^s + c_1^{ns} + c_3^{ns}\right) r_2 + \left(c_3^s + c_2^{ns} + c_4^{ns}\right) r_2^2 \tag{5.27a}$$

$$f(r_1) = \frac{\left(c_2^s + c_3^s\right)}{r_1^{l_E}} + \left(c_4^s + c_1^{ns} + c_2^{ns}\right) r + \left(c_5^s + c_3^{ns} + c_4^{ns}\right) r_1^2 \tag{5.27b}$$

which are the same singularity form as (5.16b).

The shape functions for the *Corner1* singular element are thus given as:

9-node element:

$$
\begin{aligned}
N_1 &= \frac{r^{g_1}(1-r_1)(1-r_2)}{r_1^{l_E} r_2^{l_E}} - \frac{4b_1 r_1(1-r_1)}{r_2^{l_E}} - \frac{4b_1 r_2(1-r_2)}{r_1^{l_E}} \\
&\quad + 4r_1 r_2\left[(1-r_1)(1-r_2)(4b_1+b_2-8b_3) - b_1(2-r_1-r_2)\right] \\
N_2 &= \frac{r_1(2r_1-1)}{r_2^{l_E}} - (2r_1-1)r_1 r_2\left[(2r_2-1)+4a_1(1-r_2)\right] \\
N_3 &= (2r_1-1)r_1(2r_2-1)r_2 \\
N_4 &= \frac{(2r_2-1)r_2}{r_1^{l_E}} - r_1(2r_2-1)r_2\left[(2r_1-1)+4a_1(1-r_1)\right] \\
N_5 &= \frac{4(1-r_1)r_1}{r_2^{l_E}} - 4(1-r_1)r_1 r_2\left[(2r_2-1)+4a_1(1-r_2)\right] \\
N_6 &= 4(2r_1-1)r_1(1\text{-}r_2)r_2 \\
N_7 &= 4(1-r_1)r_1(2r_2-1)r_2 \\
N_8 &= \frac{4(1-r_2)r_2}{r_1^{l_E}} - 4(1-r_2)r_1 r_2\left[(2r_1-1)+4a_1(1-r_1)\right] \\
N_9 &= 16r_1(1-r_1)r_2(1-r_2)
\end{aligned}
\tag{5.28a}
$$

8-node element:

$$
\begin{aligned}
N_1 &= \frac{r^{g_1}(1-r_1)(1-r_2)}{r_1^{l_E} r_2^{l_E}} - \frac{4b_1(1-r_1)r_1}{r_2^{l_E}} - \frac{4b_1(1-r_2)r_2}{r_1^{l_E}} \\
&\quad + 4b_1 r_1 r_2(2-r_1-r_2) \\
N_2 &= \frac{(2r_1-1)r_1}{r_2^{l_E}} - r_1 r_2\left[(2r_1-1)-2(1-r_2)(1-2a_1)\right] \\
N_3 &= r_1 r_2\left[(2r_1-1)-2(1-r_2)\right] \\
N_4 &= \frac{(2r_2-1)r_2}{r_1^{l_E}} - r_1 r_2\left[(2r_2-1)-2(1-r_1)(1-2a_1)\right] \\
N_5 &= \frac{4(1-r_1)r_1}{r_2^{l_E}} - 4(1-r_1)r_1 r_2 \\
N_6 &= 4r_1(1-r_1)r_2 \\
N_7 &= 4(1-r_1)r_1 r_2 \\
N_8 &= \frac{4(1-r_2)r_2}{r_1^{l_E}} - 4r_1(1-r_2)r_2
\end{aligned}
\tag{5.28b}
$$

where $a_1 = (2)^{l_E}$, $a_2 = (2)^{-g_1}$ and $b_1 = \dfrac{a_1 a_2}{2}$, $b_2 = a_1^2\sqrt{a_2}$, $b_3 = a_1 b_1$.

Corner2 singular element

This element contains only one weakly singular corner, which is at node 1, as shown in Figure 5.10. In the vicinity of the corner node, the singular solution has a similar form as (5.20), which is the corner singularity form. The function $f(q)$ needs to be compatible with the adjacent elements at the four element's edges.

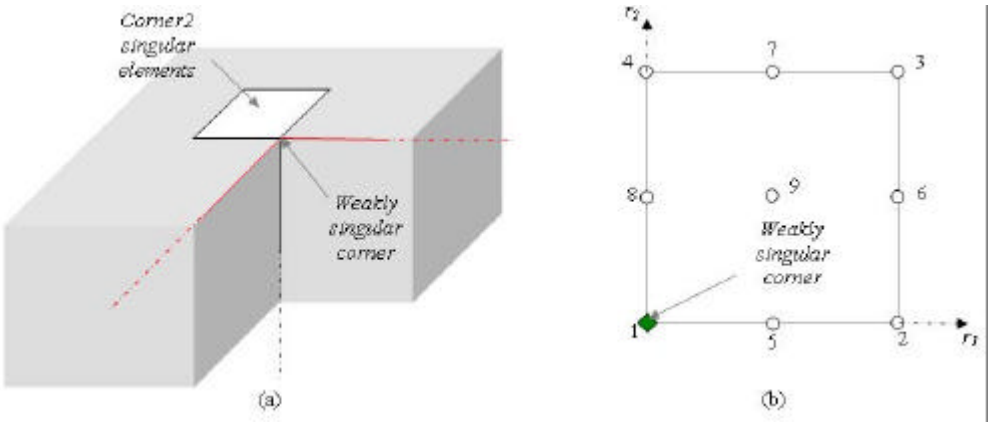


Figure 5.10. (a) Locations of *Corner2* singular elements, and (b) *Corner2* singular element definitions.

The compatibility conditions require the expression to be:

(i) quadratic along the edges $r_1 = 1$ and $r_2 = 1$, as in (5.16a), and

(ii) of two-dimensional singular form along the edges $r_1 = 0$ and $r_2 = 0$, as in (5.16b).

One possible expression is

$$f(q) = \left(1 - r_1^{(1+l_{C2})}\right)\left(1 - r_2^{(1+l_{C2})}\right) \tag{5.29}$$

which gives the corner singularity as

$$f^s\big|_{r\to 0} = c_1^s \frac{\left(1 - r_1^{(1+l_{C2})}\right)\left(1 - r_2^{(1+l_{C2})}\right)}{r^{l_{C2}}} \tag{5.30}$$

The nonsingular solution is taken to be

$$\begin{aligned} f^{ns}(r_1, r_2) = {} & c_1^{ns} r_2 + c_2^{ns} r_1 + c_3^{ns} r_1 r_2 + c_4^{ns} r_2^2 \\ & + c_5^{ns} r_1^2 + c_6^{ns} r_1 r_2^2 + c_7^{ns} r_1^2 r_2 + c_8^{ns} r_1^2 r_2^2 \end{aligned} \tag{5.31}$$

which is the complete quadrilateral form without the constant term. Hence, the complete singular expression for this element is

$$\begin{aligned} f(r_1, r_2) = {} & c_1^s \frac{\left(1 - r_1^{(1+l_{C2})}\right)\left(1 - r_2^{(1+l_{C2})}\right)}{r^{l_{C2}}} + c_1^{ns} r_2 + c_2^{ns} r_1 + c_3^{ns} r_1 r_2 \\ & + c_4^{ns} r_2^2 + c_5^{ns} r_1^2 + c_6^{ns} r_1 r_2^2 + c_7^{ns} r_1^2 r_2 + c_8^{ns} r_1^2 r_2^2 \end{aligned} \tag{5.32}$$

It is trivial to show that (5.32) satisfy the compatibility conditions at all the element edges. Hence, the sets of shape functions are:

9-node element:

$$
\begin{aligned}
N_1 &= \frac{\left(1-r_1^{(1+l_{C2})}\right)\left(1-r_2^{(1l_{C2})}\right)}{r^{l_{C2}}} \\
&\quad + 4(1-r_1)(1-r_2)(2-a)\left[r_1 r_2\left(\frac{2}{a}-\frac{2-a}{b}\right)-\frac{(r_1+r_2)}{2a}\right] \\
N_2 &= -(2r_1-1)r_1(2r_2-1)(1-r_2) \\
N_3 &= (2r_1-1)r_1(2r_2-1)r_2 \\
N_4 &= -(2r_1-1)(1-r_1)(2r_2-1)r_2 \\
N_5 &= -4(1-r_1)r_1(2r_2-1)(1-r_2) \\
N_6 &= 4(2r_1-1)r_1(1-r_2)r_2 \\
N_7 &= 4(1-r_1)r_1(2r_2-1)r_2 \\
N_8 &= -4(2r_1-1)(1-r_1)(1-r_2)r_2 \\
N_9 &= 16(1-r_1)r_1(1-r_2)r_2
\end{aligned}
\qquad (5.33a)
$$

8-node element:

$$
\begin{aligned}
N_1 &= \frac{\left(1-r_1^{(1+l_{C2})}\right)\left(1-r_2^{(1l_{C2})}\right)}{r^{l_{C2}}} \\
&\quad + 2(r_1+r_2)(1-r_1)(1-r_2)\left(1-\frac{2}{a}\right) \\
N_2 &= -r_1(1-r_2)[2(1-r_1)+(2r_2-1)] \\
N_3 &= r_1 r_2[(2r_1-1)-2(1-r_2)] \\
N_4 &= -(1-r_1)r_2[(2r_1-1)-2(1-r_2)] \\
N_5 &= 4(1-r_1)r_1(1-r_2) \\
N_6 &= 4r_1(1-r_2)r_2 \\
N_7 &= 4(1-r_1)r_1 r_2 \\
N_8 &= 4(1-r_1)(1-r_2)r_2
\end{aligned}
\qquad (5.33b)
$$

where $a=(0.5)^{l_{C2}}$ and $b=\sqrt{a}$.

Corner3 singular element

This element contains one singular edge and one weakly singular corner, which is assumed to be at $r_2=0$ and node 1, respectively, as depicted in Figure 5.11.

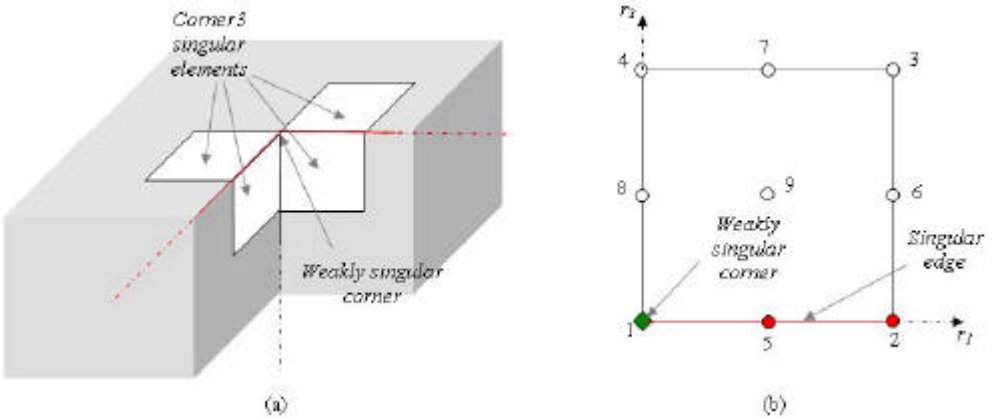


Figure 5.11. (a) Locations of *Corner3* singular elements, and (b) *Corner3* singular element definitions.

The corner singularity form is identical to (5.20), and the field is singular when $q \to 0$. The corner singularity takes the form

$$f^s\Big|_{r\to 0} = \frac{c_1^s r^{g_2}}{r_2^{l_E}} g(q) \tag{5.34}$$

where $g_2 = -l_{C2} + l_E = 0.22293$. Since this element always has *Corner2* singular element as its neighbour at $r_1 = 0$, for compatibility, $g(q)$ is taken to be $\left(1-r_1^{(1+l_{C2})}\right)\left(1-r_2^{(1+l_{C2})}\right)$, which gives the corner singularity as

$$f^s\Big|_{r\to 0} = \frac{c_1^s r^{g_2}}{r_2^{l_E}}\left(1-r_1^{(1+l_{C2})}\right)\left(1-r_2^{(1+l_{C2})}\right) \tag{5.35}$$

The edge singularity assumes the form in (5.24b), which again vanishes at the corner, and the nonsingular function is identical to that in (5.15). Therefore, the complete singular expression is

$$\begin{aligned} f(r_1, r_2) = {} & \frac{c_1^s r^{g_2}}{r_2^{l_E}}\left(1-r_1^{(1+l_{C2})}\right)\left(1-r_2^{(1+l_{C2})}\right) + \frac{\left(c_2^s + c_3^s r_1\right) r_1}{r_2^{l_E}} \\ & + c_1^{ns} r_2 + c_2^{ns} r_1 r_2 + c_3^{ns} r_2^2 + c_4^{ns} r_1 r_2^2 + c_5^{ns} r_1^2 r_2 + c_6^{ns} r_1^2 r_2^2 \end{aligned} \tag{5.36}$$

In this case, the compatibility conditions are identical to (5.16), except at $r_1 = 0$, where the order of singularity is l_{C2}. It is again trivial to show that (5.36) satisfies the compatibility requirements. The sets of shape functions are as follow:

<u>9-node element:</u>

$$\begin{aligned} N_1 = {} & \frac{r^{g_2}\left(1-r_1^{(1+l_{C2})}\right)\left(1-r_2^{(1+l_{C2})}\right)}{r_2^{l_E}} - \frac{4a_3(1-0.5a_2)(1-r_1)r_1}{r_2^{l_E}} \\ & + 2(2-a_2)(1-r_1)r_2\left\{2a_1(3a_1 - b_1(2-a_2))r_1(1-r_2)\right. \\ & \left. + a_3 r_1(2r_2-1) - a_1 a_3(1-r_2)\right\} \\ N_2 = {} & \frac{(2r_1-1)r_1}{r_2^{l_E}} - (2r_1-1)r_1 r_2\left[(2r_2-1) + 4a_1(1-r_2)\right] \\ N_3 = {} & (2r_1-1)r_1(2r_2-1)r_2 \\ N_4 = {} & -(2r_1-1)(1-r_1)(2r_2-1)r_2 \\ N_5 = {} & \frac{4(1-r_1)r_1}{r_2^{l_E}} - 4(1-r_1)r_1 r_2\left[(2r_2-1) + 4a_1(1-r_2)\right] \\ N_6 = {} & 4(2r_1-1)r_1(1-r_2)r_2 \\ N_7 = {} & 4(1-r_1)r_1(2r_2-1)r_2 \\ N_8 = {} & -4(2r_1-1)(1-r_1)(1-r_2)r_2 \\ N_9 = {} & 16(1-r_1)r_1(1-r_2)r_2 \end{aligned} \tag{5.37a}$$

8-node element:

$$
\begin{aligned}
N_1 &= \frac{r^{g_2}\left(1-r_1^{(1+l_{C2})}\right)\left(1-r_2^{(1+l_{C2})}\right)}{r_2^{l_E}} - \frac{4a_3(1-0.5a_2)(1-r_1)r_1}{r_2^{l_E}} \\
&\quad + 4a_3(1-0.5a_2)(1-r_1)r_2[r_1 + a_1(1-r_2)] \\
N_2 &= \frac{(2r_1-1)r_1}{r_2^{l_E}} - r_1 r_2[(2r_1-1) + 2(1-r_2)(1-2a_1)] \\
N_3 &= r_1 r_2[(2r_1-1) - 2(1-r_2)] \\
N_4 &= -(1-r_1)r_2[(2r_1-1) + 2(1-r_2)] \\
N_5 &= \frac{4(1-r_1)r_1}{r_2^{l_E}} - 4(1-r_1)r_1 r_2 \\
N_6 &= 4r_1(1-r_2)r_2 \\
N_7 &= 4(1-r_1)r_1 r_2 \\
N_8 &= 4(1-r_1)(1-r_2)r_2
\end{aligned}
\qquad (5.37b)
$$

where $a_1 = (2)^{l_E}$, $a_2 = (2)^{-l_{C2}}$, $a_2 = (2)^{-g_2}$ and $b_1 = \sqrt{a_2}$.

Corner4 singular element

Finally, the last singular element is considered in Figure 5.12. In this element, there is only one singular edge along $r_2 = 0$, and this singularity field dies down as it approaches the "zero" corner at node 1.

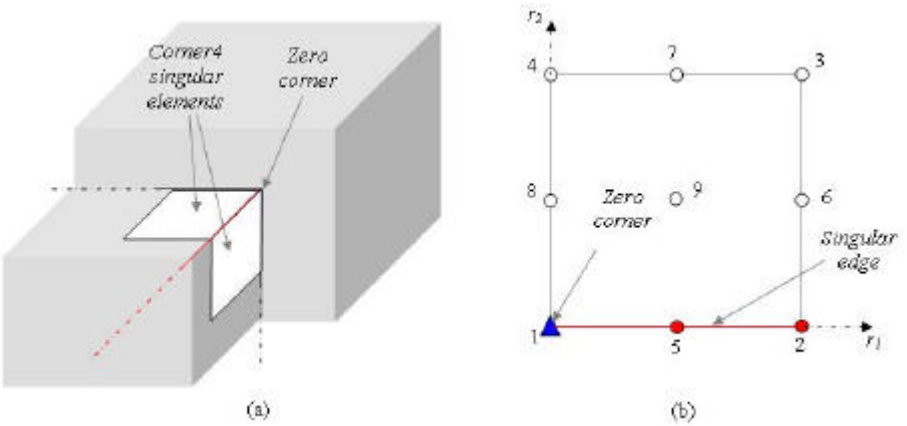


Figure 5.12. (a) Locations of *Corner4* singular elements, and (b) *Corner4* singular element definitions.

Theoretically, the field is zero along the edge $r_1 = 0$. But, as mentioned in the beginning, this zero field effect is not being specially treated. Hence, along this edge, the field is assumed to be the normal quadratic form. The edge singularity form is again that as in (5.24b), and the nonsingular part is

$$
\begin{aligned}
f^{ns}(r_1, r_2) &= c_1^{ns} r_2 + c_2^{ns} r_1 r_2 + c_3^{ns} r_2^2 + c_4^{ns} r_1 r_2^2 + c_5^{ns} r_1^2 r_2 + c_6^{ns} r_1^2 r_2^2 \\
&\quad + c_7^{ns}(0.5 - r_1)(1 - r_1)
\end{aligned}
\qquad (5.38)
$$

where the last term provides the quadratic field along the edge $r_1 = 0$.

This gives the complete singular expression as

$$f(r_1, r_2) = \frac{(c_1^s + c_2^s r_1) r_1}{r_2^{l_E}} + c_1^{ns} r_2 + c_2^{ns} r_1 r_2 + c_3^{ns} r_2^2 + c_4^{ns} r_1 r_2^2 + c_5^{ns} r_1^2 r_2 + c_6^{ns} r_1^2 r_2^2 + c_7^{ns} (0.5 - r_1)(1 - r_1) \tag{5.39}$$

Hence, the sets of shape functions are:

9-node element:

$$\begin{aligned}
N_1 &= (2r_1 - 1)(1 - r_1)(2r_2 - 1)(1 - r_2) \\
N_2 &= \frac{(2r_1 - 1) r_1}{r_2^b} - (2r_1 - 1) r_1 r_2 [(2r_2 - 1) + 4a(1 - r_2)] \\
N_3 &= (2r_1 - 1) r_1 (2r_2 - 1) r_2 \\
N_4 &= -(2r_1 - 1)(1 - r_1)(2r_2 - 1) r_2 \\
N_5 &= \frac{4(1 - r_1) r_1}{r_2^b} - 4(1 - r_1) r_1 r_2 [(2r_2 - 1) + 4a(1 - r_2)] \\
N_6 &= 4(2r_1 - 1) r_1 (1 - r_2) r_2 \\
N_7 &= 4(1 - r_1) r_1 (2r_2 - 1) r_2 \\
N_8 &= -4(2r_1 - 1)(1 - r_1)(1 - r_2) r_2 \\
N_9 &= 16(1 - r_1) r_1 (1 - r_2) r_2
\end{aligned} \tag{5.40a}$$

8-node element:

$$\begin{aligned}
N_1 &= (1 - r_1)(1 - r_2)(1 - 2(r_1 + r_2)) \\
N_2 &= \frac{(2r_1 - 1) r_1}{r_2^{l_E}} - r_1 r_2 [(2r_1 - 1) + 2(1 - r_2)(1 - 2a)] \\
N_3 &= r_1 r_2 [(2r_1 - 1) - 2(1 - r_2)] \\
N_4 &= -(1 - r_1) r_2 [(2r_1 - 1) + 2(1 - r_2)] \\
N_5 &= \frac{4(1 - r_1) r_1}{r_2^{l_E}} - 4(1 - r_1) r_1 r_2 \\
N_6 &= 4 r_1 (1 - r_2) r_2 \\
N_7 &= 4(1 - r_1) r_1 r_2 \\
N_8 &= 4(1 - r_1)(1 - r_2) r_2
\end{aligned} \tag{5.40b}$$

where $a = (2)^{l_E}$.

This completes the entire formulation of shape functions for the various singular elements. It is remarked that the derived shape functions are not unique, since different functions can be assumed for the variations of the fields on the element, for example $g(q)$ in (5.22). However they are reasonably simple forms that correctly describe the singularity behaviours at the singular nodes, and also satisfy the compatibility conditions along the element edges. Hence, they are expected to be effective in capturing the singularity fields in the vicinity of sharp corners and edges.

5.4 Numerical Integration of Boundary Integrals

This section deals with the numerical integration of the boundary integrals that arise from the implementation of the indirect BEM formulation. The boundary integral is generally of the form

$$I = \int_{-1}^{1}\int_{-1}^{1} \frac{N(x_1, x_2)}{4\pi\varepsilon \|x - x'\|} |J(x_1, x_2)| dx_1 dx_2 \tag{5.41}$$

where $N(x_1, x_2)$ is the shape function that describes the surface charge density function $\sigma(x')$, $\|x - x'\|$ is the Euclidean length of $(x - x')$ and $|J(x_1, x_2)|$ is the Jacobian of transformation that maps the element from the global coordinates to the intrinsic coordinates.

Four situations can occur in (5.41), namely,

(i) the integrand is nonsingular,

(ii) only the fundamental solution is singular, that is, the collocation point falls on the standard elements,

(iii) only the shape function is singular, that is, the element is a singular element, and

(iv) both (ii) and (iii) occur together, that is, the collocation point falls on a singular element.

The rest of this section briefly discusses the numerical techniques used to evaluate the four types of integrals above.

5.4.1 Nonsingular Integral

When the integrand is nonsingular, the two-dimensional Gauss-Legendre quadrature schemes can be used. It is essentially the product formula of the one-dimensional Gauss-Legendre quadrature rule [81], which is given by the formula

$$\int_{-1}^{1}\int_{-1}^{1} f(x_1, x_2) dx_1 dx_2 \approx \sum_{i=1}^{n1}\sum_{j=1}^{n2} f(x_i, x_j) w_i w_j \tag{5.42}$$

where x_i and x_j are the abscissae of the Gauss-Legendre formula, w_i and w_j are the corresponding weighing coefficients, and $n1$ and $n2$ are the numbers of Gauss points used in x_i and x_j directions, respectively.

5.4.2 Singular integral due to fundamental solution only

The evaluation of this singular integral is itself a research topic. Numerous methods have being developed to improve its accuracy. Broadly speaking, they can be classified into two groups, namely the weighted Gaussian quadrature formulae, and the transformation techniques. The former methods compute new sets of abscissas and weights, where the singularity is moved to the weights [82-84]. On the other hand, the transformation techniques utilize the Jacobian of transformations to remove or weaken the singularity. One well-known approach is the transformation of triangular elements into squares [85], which is referred as the regularization transformation. A study by Aliabadi and Hall [86] showed that this is a very accurate and efficient method. Another similar approach is the polar coordinate transformation [87], where the rectangular intrinsic coordinates (x_1, x_2) are replaced by the polar coordinate system (r, q), with the singular point at origin. Other types of transformations include the polynomial transformation [88, 89] and co-ordinate multi-transformations [90, 91].

The regularizing transformation technique [85] is used here to resolve this singular integral. The element is first sub-divided into two or more triangles, depending on the collocation point location. These triangles are considered as degenerate squares, which are then mapped into square elements using the following mapping functions.

$$x_i = (L_1 + L_2)x_i^{(1)} + L_3 x_i^{(2)} + L_4 x_i^{(3)}, \quad \text{for } i = 1, 2. \tag{5.43}$$

where $x_i^{(1)}$ is the triangle corner that coincides with the collocation point, and $x_i^{(2)}$ and $x_i^{(3)}$ correspond to the other two corners of the triangle, and

$$\begin{aligned} L_1 &= \frac{1}{4}(1-h_1)(1-h_2), \quad L_2 = \frac{1}{4}(1+h_1)(1-h_2), \\ L_3 &= \frac{1}{4}(1+h_1)(1+h_2), \quad L_4 = \frac{1}{4}(1-h_1)(1+h_2). \end{aligned} \tag{5.44}$$

After applying the bilinear transformation, (5.41) becomes

$$I = \sum_{k=1}^{T} \left\{ \int_{-1}^{1}\int_{-1}^{1} f(x_1, x_2) \frac{\left|J_h(h_1, h_2)\right|}{\|x - x'\|} dh_1 dh_2 \right\} \tag{5.45}$$

where $f(x_1, x_2) = \dfrac{N(x_1, x_2)}{4pe} \left|J(x_1, x_2)\right|$ is a nonsingular function, T is the number of sub-triangles depending on the collocation point, and $\left|J_h(h_1, h_2)\right| = c(1+h_2)$ is the Jacobian of transformation that

maps x onto h, and it cancels the $\frac{1}{\|x-x'\|}$ singularity so that the integrand in (5.45) is smooth throughout the integration domain. Here, c is a constant that also depends on the location of the collocation point. Since the integrals are no longer singular, the standard Gaussian quadrature scheme can be used. The mapping functions and values of T and c for the various cases, where the collocation points fall on different nodes, are summarized in Appendix C.1. Figure 5.13 shows this process for the case where the collocation point at node 1.

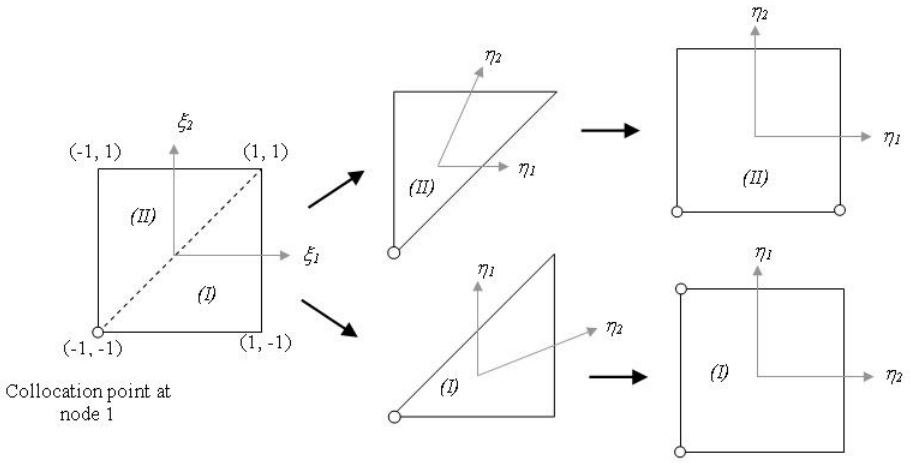


Figure 5.13. Regularization transformation process for collocation point at node 1.

5.4.3 Singular integral due to singular shape function only

This integral occurs when the integrating element is singular, that is, $N(x_1, x_2)$ in (5.41) is a singular function. This can be rewritten as

$$I = \int_{-1}^{1}\int_{-1}^{1} g(x_1, x_2)h(x_1, x_2)dx_1 dx_2 \tag{5.46}$$

where $g(x_1, x_2) = \frac{N_{ns}(x_1, x_2)}{4pe\|x-x'\|}|J(x_1, x_2)|$ is a nonsingular function, and $N_{ns}(x_1, x_2)$ is the nonsingular part of the singular shape function associated with the singularity form $h(x_1, x_2)$. According to the shape functions derived in the previous section, these are

(i) $$h_1 = \frac{1}{(1+x_i)^{l_E}} \quad \text{for } i = 1 \text{ or } 2. \tag{5.47a}$$

(ii) $$h_2 = \frac{r^{g_1}}{(1+x_1)^{l_E}(1+x_2)^{l_E}} \tag{5.47b}$$

(iii) $h_3 = \dfrac{1}{r^{l_{C2}}}$ (5.47c)

(iv) $h_4 = \dfrac{r^{g_2}}{(1+x_i)^{l_E}}$ for $i = 1$ or 2. (5.47d)

and $r = \dfrac{1}{2}\sqrt{(1+x_1)^2 + (1+x_2)^2}$.

Since all these singularities exist only along the boundaries, the Gauss-Jacobi quadrature formula [80] is particularly suitable for evaluating such weakly boundary singular integrals. The one-dimensional Gauss-Jacobi formula is given by

$$\int_{-1}^{1} (1-x)^a (1+x)^b f(x)\,dx \approx \sum_{i=1}^{n} f(z_i) v_i \tag{5.48}$$

where z_i and v_i are the abscissae and weights of the Gauss-Jacobi formula, and *a* and *b* are weakly singular exponents, with value greater than -1.0. However, for numerical stability, *a* should be greater than -0.98 [80]. The Gauss-Jacobi scheme, together with the Gauss-Legendre scheme, are use to evaluate the integral in (5.46) with singularity forms given by (5.47a)-(5.47d). For example, if $h = \dfrac{1}{(1+x_i)^{l_E}}$ for $i = 1$ or 2 , then (5.46) is evaluated as

$$I = \int_{-1}^{1}\int_{-1}^{1} g(x_1, x_2)(1+x_i)^{-l_E}\,dx_1 dx_2 \approx \sum_{i=1}^{n1}\sum_{j=1}^{n2} g(x_j, z_i) w_j v_i, \quad \text{for } i = 1 \text{ or } 2. \tag{5.49}$$

where the Gauss-Jacobi quadrature is used in the *i*-direction, and the Gauss-Legendre quadrature is used in the *j*-direction. Similar approaches are used for the singularity form in (5.47b) and (5.47d), by choosing the correct schemes in the appropriate directions. As for (5.47c), which is only singular at one point, this point singularity can be removed by expressing the integral as

$$I = \int_{-1}^{1}\int_{-1}^{1} k(x_1, x_2)(1+x_i)^{-b}\,dx_1 dx_2 \approx \sum_{i=1}^{n1}\sum_{j=1}^{n2} k(x_j, z_i) w_j v_i, \quad \text{for } i = 1 \text{ or } 2. \tag{5.50}$$

where $k(x_1, x_2) = \dfrac{(1+x_i)^b\, g(x_1, x_2)}{r^{l_{C2}}}$. The term $(1+x_i)^b$ de-singularizes the weakly singular effect due to $\dfrac{1}{r^{l_{C2}}}$, since *b* is chosen to be 0.98.

5.4.4 Singular integral due to fundamental solution and singular shape function

This situation occurs when the collocation point falls on a singular element, where both the fundamental solution and the shape functions are singular. In this case, the integral is more conveniently rewritten as

$$I = \int_{-1}^{1}\int_{-1}^{1} g(x_1, x_2)\left(\frac{1}{\|x - x'\|}\right) h_i(x_1, x_2)\, dx_1 dx_2 \tag{5.51}$$

where $g(x_1, x_2)$ is again a non-singular function, and $h_i(x_1, x_2)$ is one of the singular functions given in the above sub-section. To remove these singularities completely, the following techniques are used.

First, the stronger singularity due to the fundamental solution is removed by using the regularizating transformation technique discussed in Section 5.4.2. After the transformation, (5.51) becomes

$$I = \sum_{k=1}^{T}\int_{-1}^{1}\int_{-1}^{1}\left[g(x_1, x_2)\frac{c(1+h_2)}{\|x - x'\|}\right] h_i(x_1, x_2)\, dh_1 dh_2 \tag{5.52}$$

where the function in the square bracket is nonsingular as $(1+h_2)$ cancels the singularity in $\frac{1}{\|x - x'\|}$.

However, $h_i(x_1, x_2)$ may still be weakly singular along the boundaries. If $h_i(x_1, x_2)$ is nonsingular, then the integration is carried out as usual by using Gauss-Legendre schemes. However, if it contains the edge singularity form as

$$h_i(x_1, x_2) = (1-h_1)^{-c_1}(1+h_1)^{-c_2}(1-h_2)^{-d_1}(1+h_2)^{-d_2} \tag{5.53}$$

where c_1, c_2, d_1 and d_2 correspond to the orders of singularity of $h_i(x_1, x_2)$, the Gauss-Jacobi formulas, with $a = c_1$, $b = c_2$ for h_1 and $a = d_1$, $b = d_2$ for h_2 can then be used to evaluate the singular integral. This approach fully exploits the capability of the Gauss-Jacobi formulae to deal with integrals that contain only weak singularities. Furthermore, if $h_i(x_1, x_2)$ is of the point singularity form, the technique used in Section 5.4.3 can be employed. The final forms of the integrals and the appropriate Gaussian Quadrature formulas for the different singular forms $h_i(x_1, x_2)$ at different collocation nodes are presented in Appendix C.2.

5.5 Numerical Examples

Using the singular shape functions and the appropriate techniques to evaluate the boundary integrals, the singular element method is implemented in a three-dimensional BEM code. In this section, some numerical examples are presented to demonstrate the accuracy of these singular elements in evaluating, (i) the capacitance, and (ii) the electrostatics force for electrostatics problems. The 8-node singular elements are denoted as quadsng8, while 9-node singular elements as quadsng9. The results reveal that the singular elements can produce very accurate results. The improvement is most likely due to the fact that the singular elements can describe the true surface charge distributions (and hence the electrostatics force distributions) more accurately in the singularity regions. Electromechanical coupling analyses are also conducted to investigate the effects of using singular elements on the functionality of some micro-devices, namely micro-beam switches, a comb-drive and a micro-mirror.

5.5.1 Capacitance extraction problems

Two examples, involving self-capacitance extraction, are used to determine the accuracy of the singular elements in obtaining the capacitance of (i) a cube and (ii) an L-shaped conductor. These two examples are chosen because they contain the types of singularity features that are studied here. Furthermore, they are used as test problems by Tausch and White [92] to evaluate the accuracy of their mesh refinement method. In fact, the "exact" capacitance for the two examples are derived from [92], by extrapolating their mesh refinement results.

Cube example

This example, as depicted in Figure 5.14, is discretized with uniform square panels, and it contains both *Edge* and *Corner1* singular elements. The "exact" capacitance is estimated to be 73.51 *pF*.

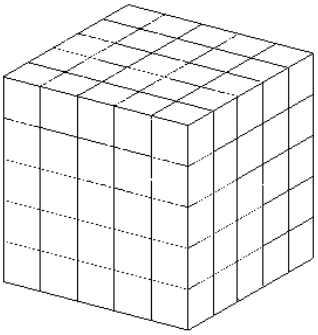


Figure 5.14. Discretization of cube example.

The relative percentage errors with respect to the 'exact' solution are given in the *log-log* plot in Figure 5.15, where the horizontal axis corresponds to the problem size *n*. This plot also includes the results from[92], which employed the adaptive *h*- mesh refinement technique using constant elements.

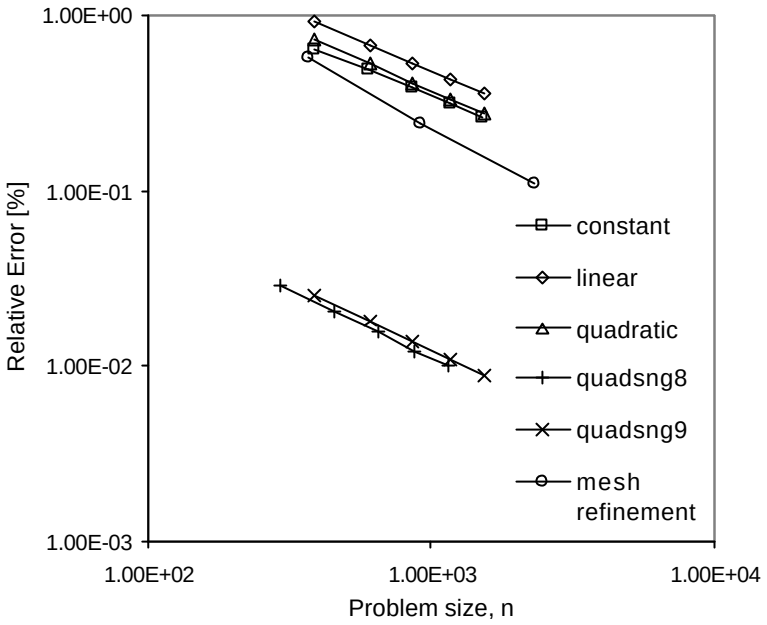


Figure 5.15. Relative percentage errors for the capacitance of cube example. "Exact" solution is 73.51 *pF*.

It is obvious from Figure 5.15 that the singular elements produce very accurate results, even for the coarsest mesh density considered here (only four elements along each edge). In general, it is more accurate than the other standard elements by at least one order of magnitude. It is also noted that the singular element approach has the same convergence rate as the standard elements. On the other hand, the adaptive mesh refinement technique has a better convergence behaviour. However, its results are still much less accurate than the singular element approach because the singularity solutions at the sharp corners and edges cannot be adequately represented by low-order polynomial elements. Furthermore, in order for the mesh refinement approach to attain convergence, it usually has to solve a number of progressively larger problems, which can be quite expensive. From this example, it is shown that the singular shape functions for the *Edge* and *Corner1* singular elements are feasible and accurate, at least in terms of capacitance extraction.

L-shape example

The geometry of this example is shown in Figure 5.16, which contains the *Edge*, *Corner1*, *Corner2* and *Corner3* singular elements. The "exact" solution is estimated to be 112.15 *pF*.

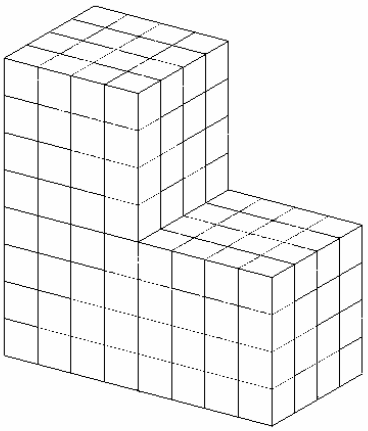


Figure 5.16. Discretization of the L-shaped example.

Likewise, the relative percentage errors for the various elements and the adaptive mesh refinement technique are plotted in Figure 5.17. It is again seen that the singular element approach produces very accurate results.

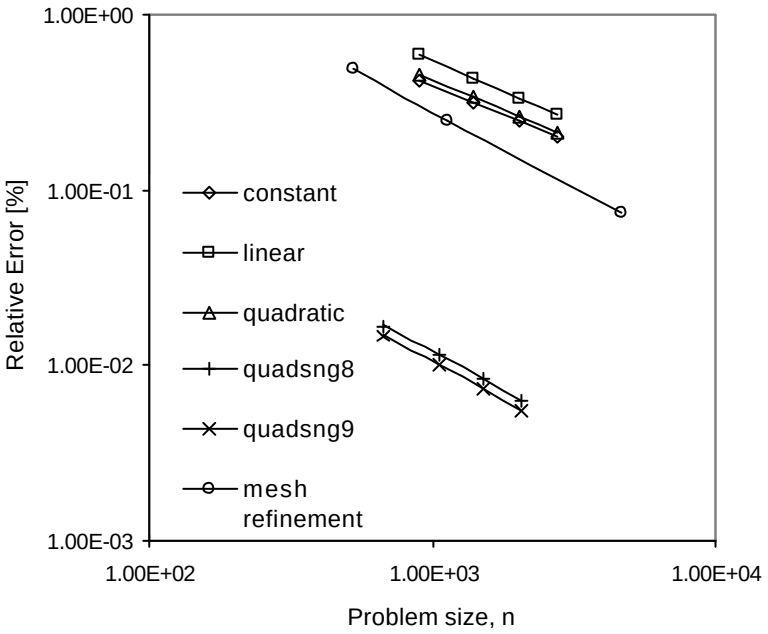


Figure 5.17. Relative percentage errors for the capacitance of the cube example. "Exact" solution is 112.15 *pF*.

Biased element distribution effect

Very often, more accurate results can be obtained by using the *r*-mesh refinement technique near the singularity regions, because the influence of the singularity is usually quite localized. For geometrical induced singularities, which are easily identified from the geometry such as the solid angles, the mesh refinement can be done manually at the preprocessing stage. The cube example (with five elements along each edge) is used to study the biased element distribution effect. The bias ratio *R* is defined to be the ratio of the largest element length (at the center of the cube) to the smallest element length (near to the edges and corners), and is taken to be 1.0, 2.0, 3.0, 4.0 and 5.0 in the present computations. Figure 5.18 shows the meshes for the various *R* values, and the corresponding results are presented in Figure 5.19.

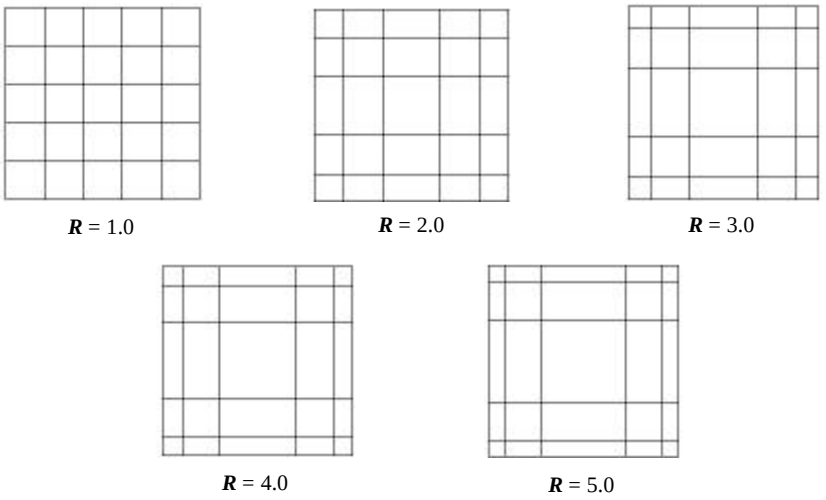


Figure 5.18. Surface meshes for different biased ratio ***R***., ranging from 1.0 to 5.0.

It is noted that the *r*-mesh refinement technique tends to work more consistently for the standard elements. However, their convergence rates decrease with increasing value of ***R***, which suggests that further increasing the bias ratio has little or no effect on the results. This observation is consistent with the general observation that the *r*-mesh refinement approach does not guarantee that the solution converges to the exact one, simply because this cannot be achieved by just rearrangement of the elements alone.

On the other hand, the singular element solutions are better for low bias ratios, and then deteriorate with increasing bias ratios. This observation can be explained as follows. It is recalled that singular elements are used only in the singularity regions, while the standard quadratic elements are used

elsewhere. The singular elements are expected to capture the rapidly varying singularity fields accurately, whereas the standard quadratic elements can describe the regular fields that are remote from the singularity regions. In order for the singular elements to perform at their optimum, it is necessary that the size of the singular elements be comparable with the actual singularity regions. To be more explicit, consider the situation where the singular elements are much larger than the actual singularity regions. In this case, it is obvious that the singular shape functions, specifically the nonsingular parts, are inadequate to represent the actual solutions. On the other hand, if the singular elements are too small, the standard quadratic elements adjacent to them have to represent part of the singularity fields. The major source of error is then due to the inability of the standard quadratic elements to represent the rapidly varying solutions near the singularity regions. This is probably the scenario for this cube example with large bias ratios. The important question to ask here is: when is element-biasing useful, or what determines the sizes of the singularity regions? One possible factor that affects the size of the singularity regions is the proximity between the conductors. A study is conducted in the next section with respect to the electrostatics force analysis.

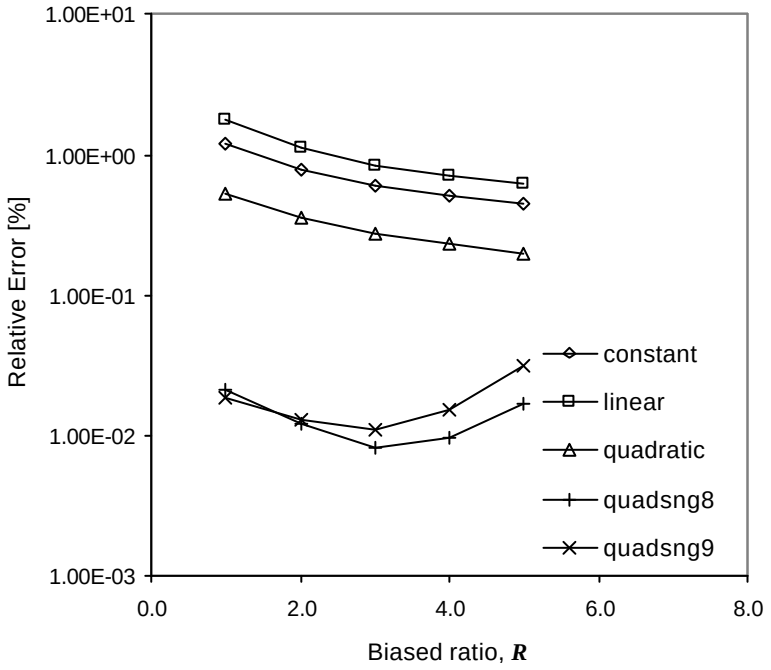


Figure 5.19. Relative percentage errors for the capacitance of the cube example with biased ratio $\boldsymbol{R}$ = 1.0, 2.0, 3.0, 4.0 and 5.0.

5.5.2 Electrostatics force analysis

Many microelectromechanical systems are actuated by electrostatics forces, which are otherwise practically too small for macro-applications. The electrostatics force is directly proportional to the square of the surface charge density. Hence, this provides another motivation for performing electrostatics analysis accurately. This section evaluates the usefulness of the singular elements with the cube example, which is now placed over an infinite ground plane at some distance d. The resultant electrostatics force induced on the face of the cube that is closest to the ground plane, which has the largest magnitude, is calculated in this study. The analysis is conducted for three different distances, namely d = 1.0, 0.5 and 0.2. Note that the 'exact' solutions are approximated from the extrapolation of the uniform mesh refinement results.

Convergence behaviors using uniform meshes

This analysis studies the convergence behaviour of the various elements using uniform meshes. The convergence behaviors for the various elements for different distance d are plotted in Figures 5.20-5.22.

It is again obvious that the use of singular elements can produce more accurate results than using the standard elements alone. However, the results for the constant elements improve as the distance d decreases, whereas the situation is the opposite for the singular elements. This observation can be explained as follows. It is expected that the centre portion of the cube's face generate a uniform electric field with the ground plane. As the distance d decreases, the size of this uniform field grows, which means that a larger portion of the face of the cube has a more or less constant surface charge density. Hence, constant elements can produce good results for smaller values of d. Also, an increase in the size of the uniform field indicates an equivalent decrease in the singularity region. Hence, for the coarse mesh cases, the singular elements are too large to be able to capture the true surface charge distribution accurately (which partly consists of the uniform field). However as the mesh is refined, the singular element results improve significantly, while this is not the case for the constant elements. This is simply because the true fields can be captured more accurately in both the uniform and singularity regions, by the quadratic elements and singular elements, respectively. For the constant elements, the uniform field remains accurate, but the singularity region is still poorly represented. To further

illustrate this point, the surface charge density distributions for the face of the cube are plotted for various distances in the Figure 5.23.

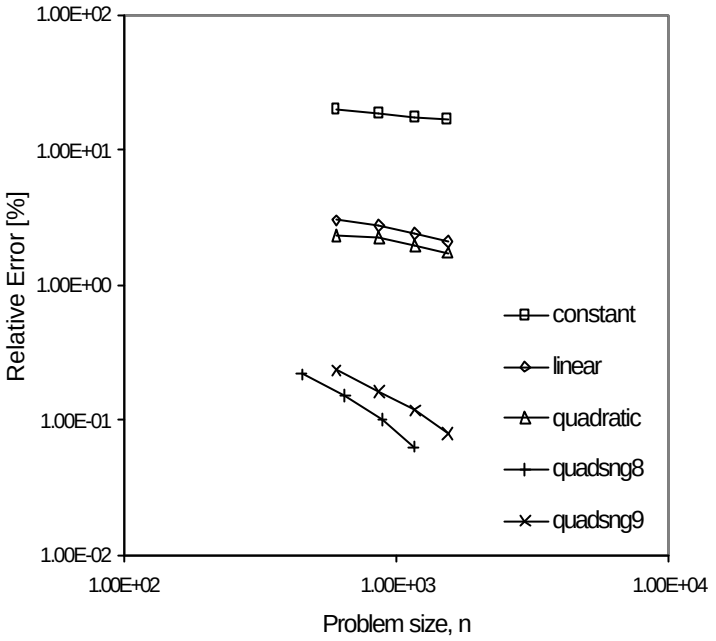


Figure 5.20. Relative percentage errors for the electrostatics force on the cube face at distance $d = 1.0$.

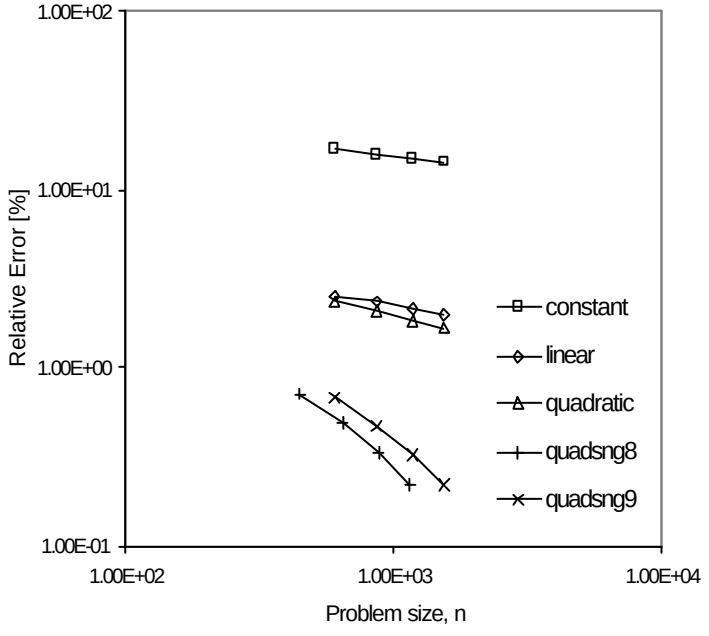


Figure 5.21. Relative percentage errors for the electrostatics force on the cube face at distance $d = 0.5$.

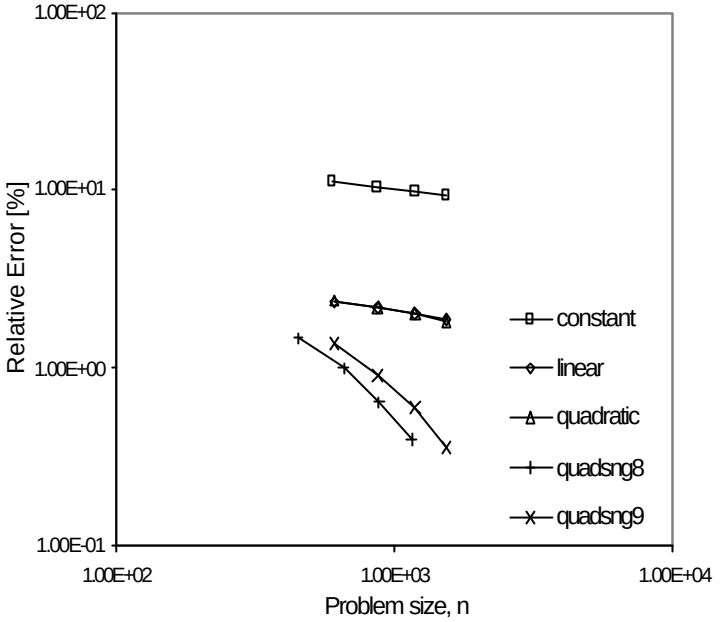


Figure 5.22. Relative percentage errors for the electrostatics force on the cube face at distance $d = 0.2$.

Figure 5.23(a) depicts the general surface charge distribution that one expects, where the centre portion shows a relatively uniform distribution that becomes rapidly varying as it approaches the edges and corners. Figures 5.23(b)-(d) show the contour plots of the distributions for the different distances, namely, for $d = 1.0$, 0.5 and 0.2, respectively.

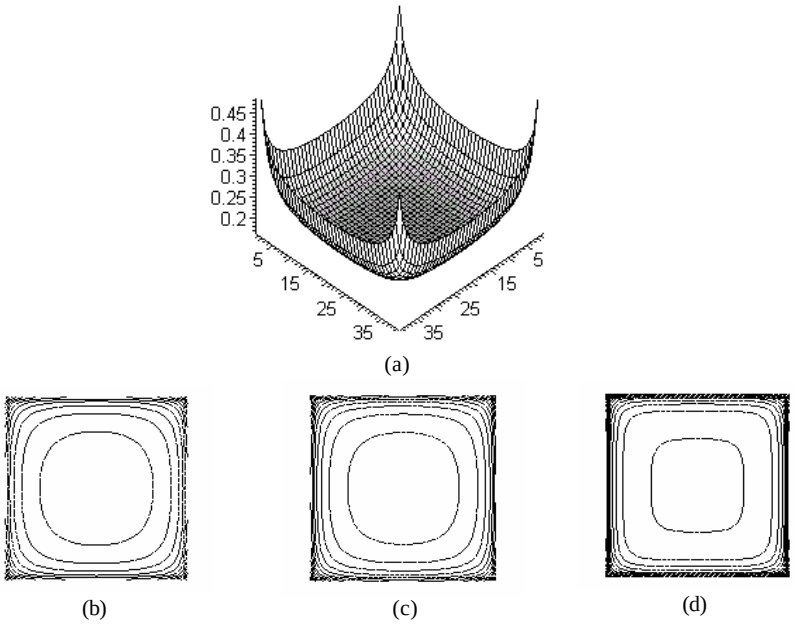


Figure 5.23. (a) A general surface charge distribution, (b) contour plots of surface charge density distributions at $d = 1.0$, (c) $d = 0.5$, and (d) $d = 0.2$.

The density of the contour lines signifies the rate at which the surface charge density is varying. It is obvious from the figures that the size of the uniform/singularity fields for d = 0.2 is significantly larger/smaller than the two other cases. This is consistent with the explanation given above. On the other hand, the difference is not obvious between distributions for d = 1.0 and d = 0.5. This seems to suggest that the surface charge distributions remain similar beyond a certain distance.

Biased elements distribution effects at different distances from ground plane

In light of the above analysis, it is noted that the uniform mesh refinement technique is not an efficient approach, as the additional elements used within the uniform field have negligible effect on the solution. The more favourable approach is the *r*-mesh refinement technique. However, the questions of "when is element biasing necessary?", and "how much is required?", still remain to be answered. The following study aims to draw a general relationship between the distance d and the bias ratio $\boldsymbol{R}$ required to give an optimal solution with the singular element approach, at least for this cube example. The biased meshes are those used in the previous analysis, as depicted in Figure 5.18. The results are normalized with respect to their corresponding 'exact' solutions, and are plotted in Figure 5.24. The points at which the various curves cut the normalized line (dashed line) give the optimal bias ratios that produce the 'exact' solutions. Table 5.1 summarizes the optimal bias ratios for the various distances using the quadsng8 and quadsng9 elements.

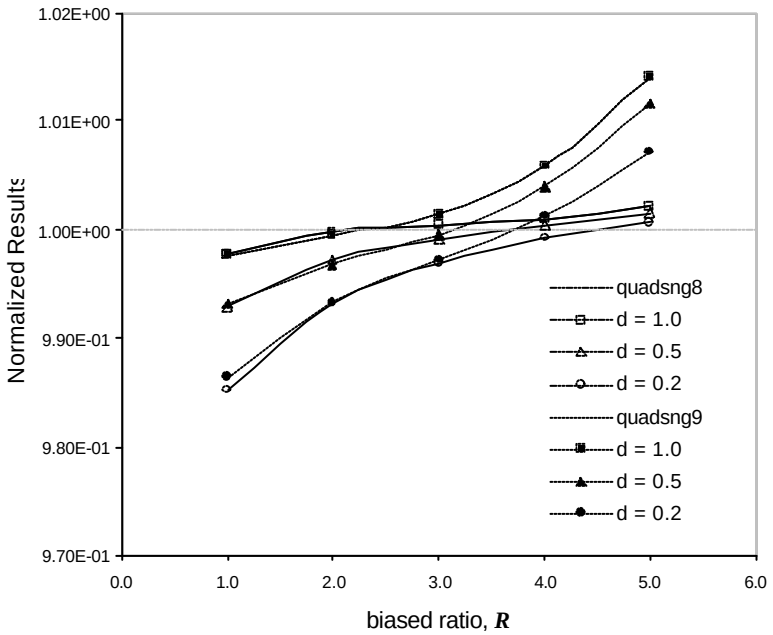


Figure 5.24. The normalized results for the biased elements study for different distance d from the ground plane.

From Table 5.1, it is observed that larger bias ratios are required as the distance *d* decreases. This is again consistent with the observations noted earlier. Another interesting observation seen in Figure 5.24 is that the 8-node singular element seems to be less sensitive to over-biasing (when the bias ratio is larger than the optimal value) in this example. The figure shows clearly that the results for the 9-node singular element deteriorate much faster than the results for the 8-node singular elements when the elements are over-biased. Hence, this makes the 8-node singular elements more favourable than 9-node ones, especially considering that their shape functions are simpler, and they are computationally less expensive (with only 8 nodes per element).

Table 5.1. Optimal biased ratios for the singular elements for different distances.

Distance, *d*	**Optimal biased ratio, *R***	
	Quadsng8	**Quadsng9**
1.0	2.2	2.3
0.5	3.7	3.3
0.2	4.5	3.8

5.5.3 Electromechanical coupling analysis

Electrostatics force is one common driving force used to actuate micro-parts in some MEMS devices. Electromechanical coupling arises when electrostatics forces, which are induced by the applied voltages, deform parts of the structures. The deformation, on the other hand, is governed by the stiffness of the structures. In general, the deformed structures may result in further changes in the surface charge distribution, and thus the electrostatics forces acting on the structures. Hence, the coupling analysis requires one to solve for a self-consistent equilibrium state, in which the electrostatics forces are exactly counter-balanced by the mechanical forces due to the stiffness of the structure.

The multilevel Newton method [15] is employed to solve for the self-consistent equilibrium state. This method requires an electrostatics solver that computes the surface charges, which are then used to compute the electrostatics forces for an applied voltage. It also requires a mechanical solver that calculates the structural deformation when subjected to the electrostatics forces. In this study, an in-house code is used for the electrostatics solver, while a commercial general-purpose finite element software package ABAQUS† is used for the mechanical analysis. The electromechanical coupling analysis is more clearly described in Appendix E.

† ABAQUS. HKS Hibbitt, Karlssoon & Sorensen, Inc. 1080 Main Street, Pawtucket, USA.

In this section, three typical micro-devices, namely, the micro-beam switch, the comb-drive actuator and the micro-mirror are analyzed. In the micro-beam examples, which include a cantilever beam and a doubly-clamped beam, the effects of the force distribution on the pull-in voltage are examined. In the comb-drive example, the comb-finger levitation problem is addressed, in which the deflection profile of the comb-finger is important. Finally, in the micro-mirror example, the tilting angles of the mirror are studied. These examples are have been used earlier in [12-15].

Four different types of boundary elements are used here, namely, the constant, linear, quadratic and quadratic-singular elements. The first three types are the standard elements, with the names denoting the order of the polynomial of the basis functions used for the surface charge variations on the element. The quadratic-singular element corresponds to the case where standard quadratic elements are used for the non-singular regions and singular elements for the singular regions. It is mentioned here that the problem sizes for the various types of elements are different for the boundary element analysis, where the quadratic/quadratic-singular elements are usually about four times larger than the constant/linear elements. This makes the comparisons less meaningful, since the former types of elements are expected to produce more accurate results simply because of the larger degree of freedoms. Nevertheless, the results for the constant and linear elements are also included for completeness sake. As for the finite elements, the 27 node solid elements are used for all cases.

Micro-beam examples

Micro-beams are often used as on/off switches in micro-devices. Typically, the beam is placed over the substrate (usually grounded and coated with a layer of dielectric to prevent short-circuiting) with a small gap between them. When a voltage is applied on the beam, electrostatics forces are exerted on the beam causing it to bend towards the substrate. As the voltage increases, the forces increase rapidly and deflection of the beam increases non-linearly. This continues until a critical voltage at which the beam collapses abruptly onto the substrate. This critical voltage is known as the pull-in voltage, which is an important parameter that defines the on/off states of the micro-switch.

In this example, the micro-beam has dimensions of 100x2x1 *mm*. The gap between the beam and the substrate is assumed to be 1 *mm*, with a 0.3 *mm* thick dielectric coating. The pull-in voltage is attained when the beam reaches a certain displacement profile and deflects in an unstable manner towards the

substrate. Two situations are considered here, namely, when the beam is fixed only at one end (cantilever beam), and when both ends are fixed (doubly-clamped beam) [13-15].

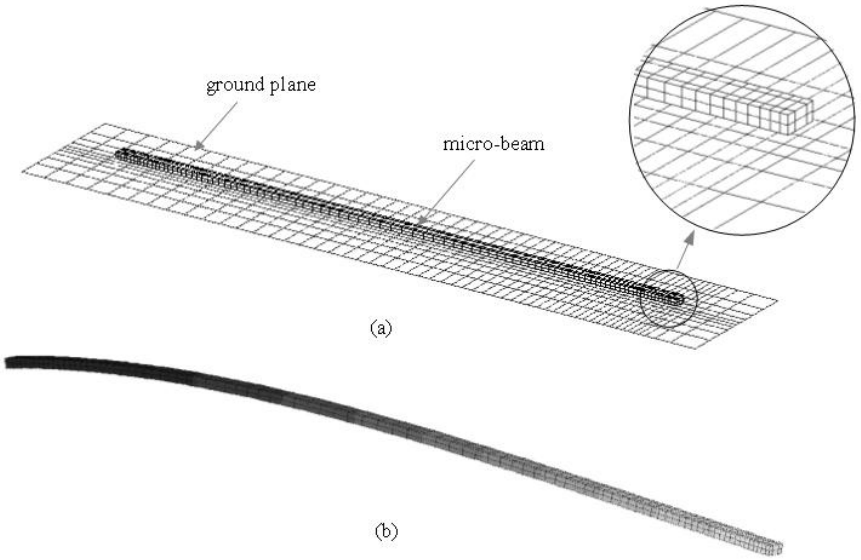


Figure 5.25. (a) Discretization of cantilever micro-beam example, (b) Deflection profile of beam before pull-in voltage, with magnification of 20.

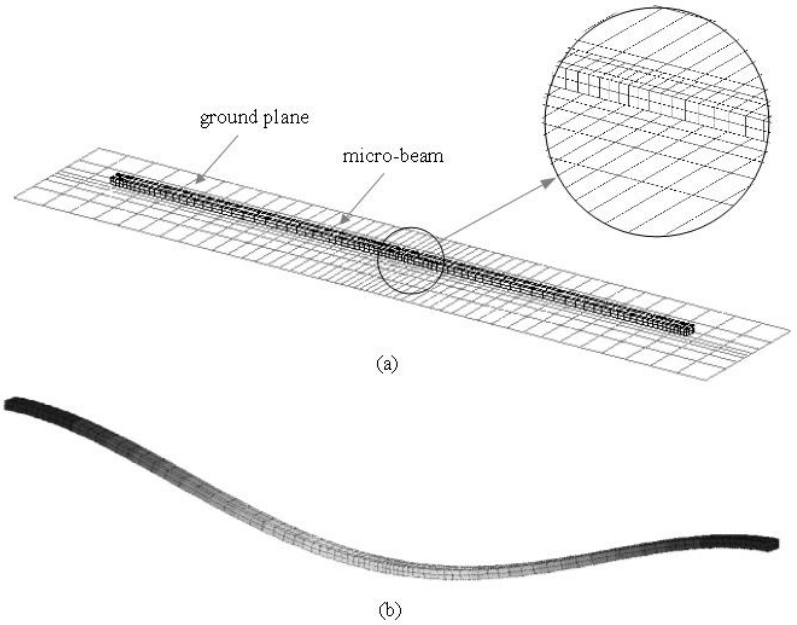


Figure 5.26. Discretization of doubly-clamped micro-beam example, (b) Deflection profile of beam before pull-in voltage, with magnification of 20.

Figures 5.25 and 5.26 show the discretized mesh, and the beam deflection profiles for the cantilever and doubly-clamped beams, respectively. The meshes for the two problems are not identical, as mesh refinements are employed at different regions appropriately. The maximum deflections (at the free end for the cantilever beam and the mid-span for the double-clamped beam) at different applied voltages

for the various elements are plotted in Figures 5.27 and 5.28, respectively. The pull-in voltages are summarized in Table 5.2.

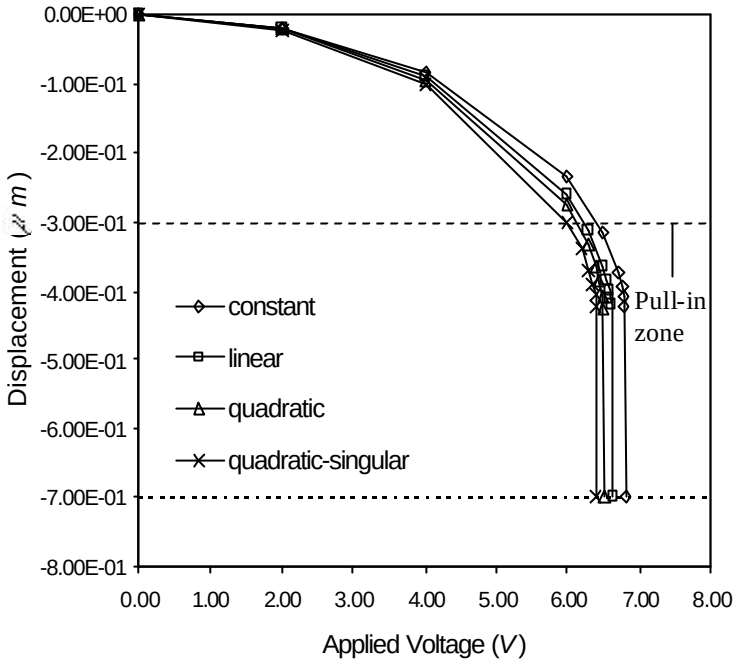


Figure 5.27. Variation of maximum deflection with applied voltage for different elements in cantilever micro-beam.

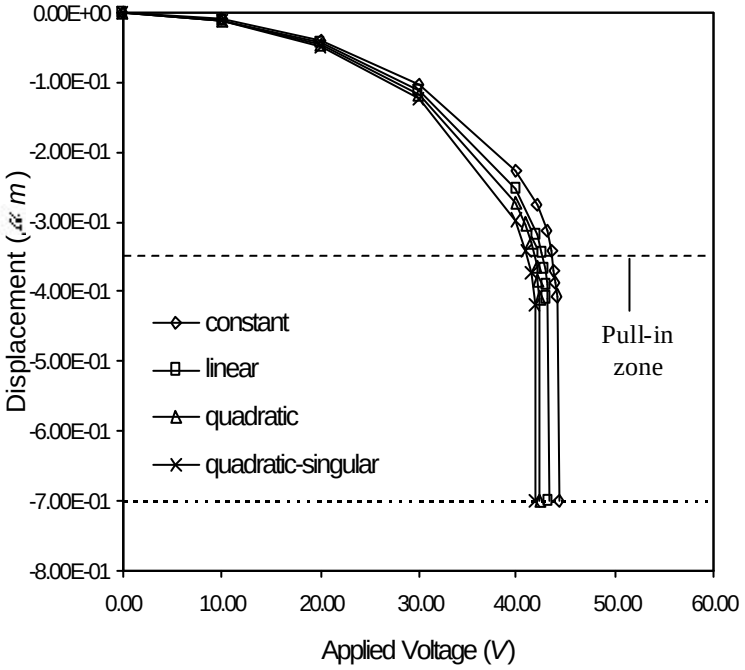


Figure 5.28. Variation of maximum deflection with applied voltage for different elements in doubly-clamped micro-beam.

Table 5.2. Pull-in voltages for the beam examples for different elements.

Element type	Pull-in voltage (V)	
	Cantilever	Double-clamped
Constant	6.80	44.2
Linear	6.63	43.2
Quadratic	6.51	42.4
Quadratic-singular	6.40	41.9

From Table 5.2, it is noted that the pull-in voltages are not significantly different for different elements. Compared to the quadratic-singular, the constant element shows differences of 5-7%, linear elements 3-4%, and quadratic element only 1-2%. The results essentially demonstrate the ability of the various standard elements to capture the actual surface charge distributions, where the higher-order elements obviously perform better. It seems that the corner and edges singularity effects, through the surface force distributions, are not significant as far as the pull-in voltage is concerned. One of the possible reasons is the rapidly changing gradient of the curve near the pull-in voltage. This effect is due to the highly nonlinear relationship between the induced electrostatics force and the gap between the beam and the substrate. From Figures 5.27 and 5.28, it is first noted that the differences in the curves are diverging before the pull-in phenomenon sets in. Specifically, the differences in the maximum deflection build up gradually to more than 20 % for the constant element, and about 14 % and 8 % for the linear and quadratic elements, respectively. However, in the pull-in zone, the curves are parallel to each other because the beam deflects rapidly with small increments in the applied voltage. This makes the choice of elements less important once the beam is in the pull-in zone.

Comb-finger levitation problem

Levitation [93] is a parasitic phenomenon that is often encountered in comb-drive designs. Generally, this effect appears in problems that contain more than two conductors. Consider the simplified model shown in Figure 5.29, which consists of only three comb fingers, each of dimensions 20x1x1 *mm* and suspended at 1 *mm* over a ground plane. The central finger, which is grounded and fixed at its left end, bends upward under the levitation force, when the two outer fingers (assumed to be fixed) are connected to an external voltage source. Figure 5.30 shows the maximum deflection of the central finger versus the applied voltages for the various elements used.

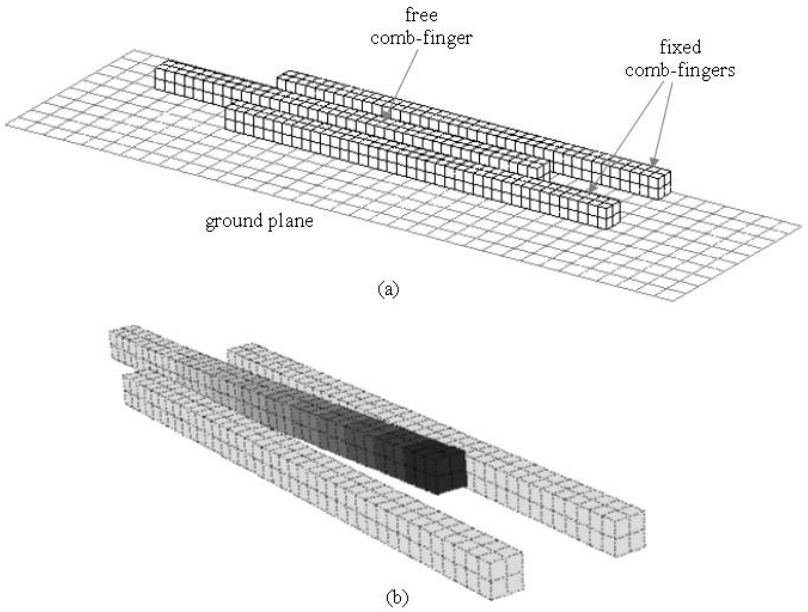


Figure 5.29. (a) Discretization of comb-finger levitation example, (b) Deflection profile of comb-finger at 200 V, with magnification of 10.

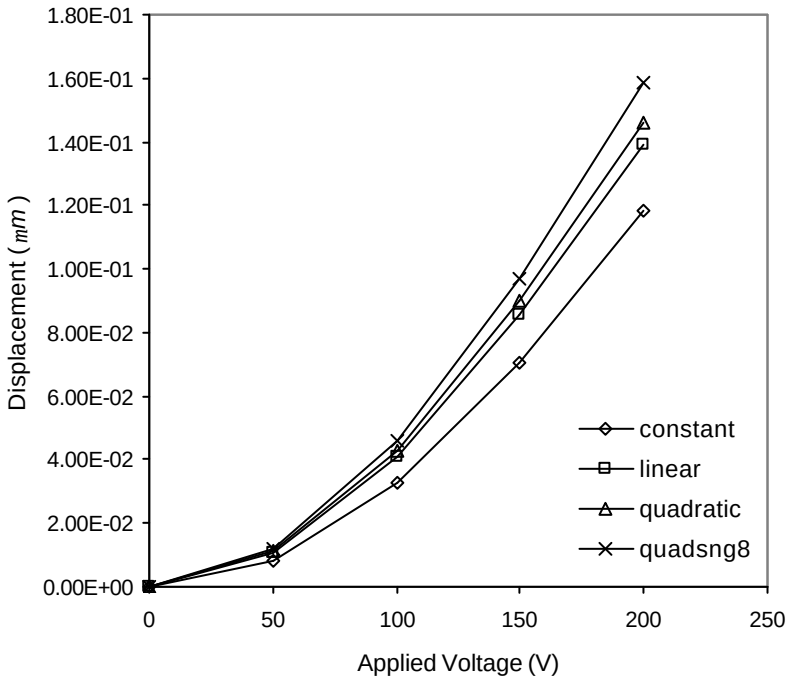


Figure 5.30. Comb-finger maximum deflections versus applied voltages for the various elements.

From Figure 5.30, it is noted that the maximum deflections of the finger are quite different for the different types of elements. Again, with reference to the results of the quadratic-singular element, the differences in the deflections at the tip of the central finger for the various elements at different applied voltages are tabulated in Table 5.3. It is observed that the differences in terms of deflections are more significant than that for the pull-in voltages for the micro-beam examples. Hence, this example reveals that the distribution of the forces can have significant effect on the deflection profile, especially on slender beam-like structures.

Table 5.3. Percentage differences in the deflections at the tip of the central finger, with respect to the results of quadratic-singular, for the various standard elements.

Applied voltage (*V*)	**Differences in deflections of central finger (%)**		
	Constant	**Linear**	**Quadratic**
50	30.4	10.8	6.02
100	28.8	11.0	7.02
150	27.4	11.7	7.12
200	25.4	12.5	8.18

Micro-mirror example

This example is similar to the problem discussed in [12]. The geometry of the problem together with the discretization is depicted in Figure 5.31(a). The mirror plate (10 *mm* in the diagonal length and 0.4 *mm* thick) is fixed at the ends of two torsional arms. Two electrodes (6 x 3 x 1 *mm*) slightly above the ground plane are placed at 1.5 *mm* underneath the mirror. A voltage is applied on one of the electrodes (driving source) with all the other conductors held at zero volts. This has the effect of tilting the mirror towards the driving electrode. In this example, the important results are the tilting angles of the mirror, which are obtained from the displacements at the tips of the mirror. The general deflection profile of the mirror is shown in Figure 5.31(b). Figure 5.32 shows the tilting angles of the mirror under different applied voltages for the various elements. The percentage differences with respect to the quadratic-singular solutions for the standard elements are computed and tabulated in Table 5.4.

This example also shows some differences in the tilting angles of the mirror obtained by using different types of elements. It is noted that the differences grow quickly as the applied voltages increase. This is again believed to be due to the highly nonlinear relationship between the induced electrostatics force and the gap between the conductors. This observation is consistent with the results for the micro-beam examples, before the pull-in zone.

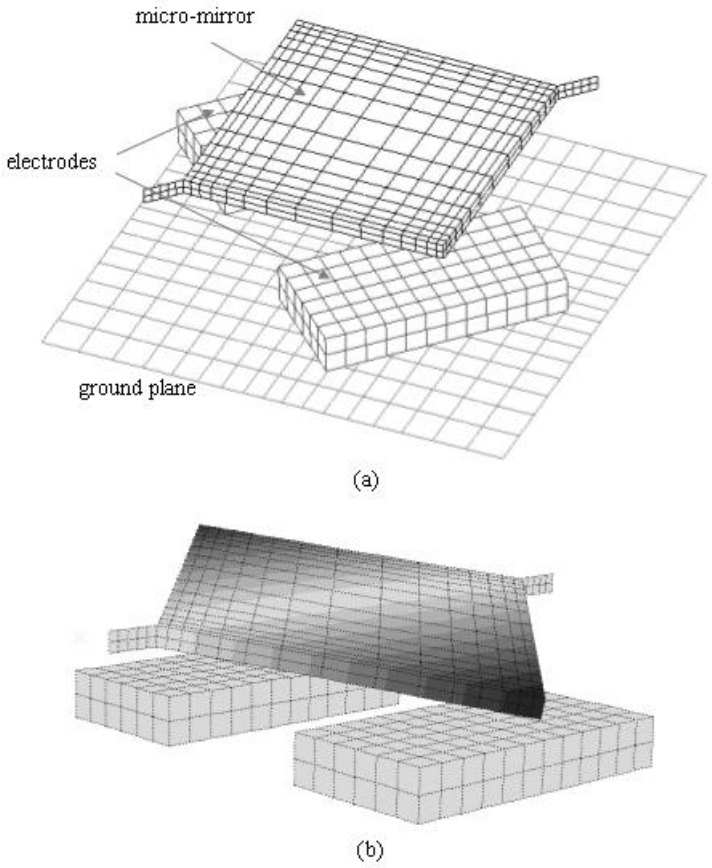


Figure 5.31. (a) Discretization of micro-mirror, (b) Deflection profile of micro-mirror at 350 *V*, with magnification of 5.

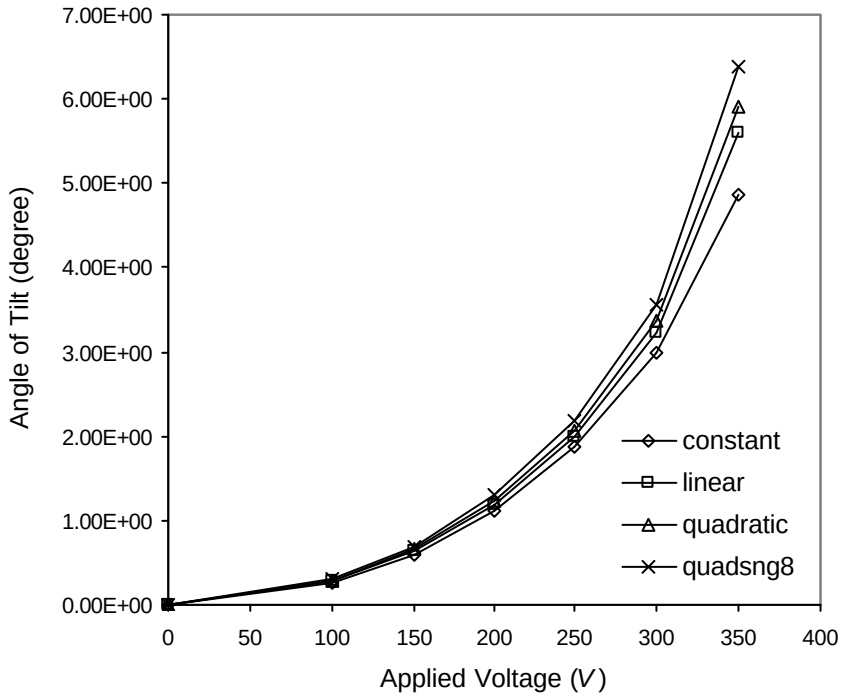


Figure 5.32. Mirror tilting angles versus applied voltages for the various elements.

Table 5.4. Percentage differences in tilting angles of micro-mirror for different elements.

Applied voltage (*V*)	Differences in tilting angles of the mirror (%)		
	Constant	Linear	Quadratic
100	12.4	8.05	5.03
150	12.8	8.07	5.04
200	13.3	8.17	5.09
250	14.3	8.41	5.26
300	16.2	9.49	5.56
350	23.7	12.1	7.32

5.6 Conclusion for Three-Dimensional Singular Elements

New singular elements have been developed for three-dimensional boundary element analysis of corner and edge singularities in potential problems. The singular elements can represent the correct singularity behaviours in the vicinity of the edges and corners, because the singular features are incorporated in their shape functions. Two sets of singular elements are formulated, namely the 8-node and 9-node elements.

It is demonstrated that the singular boundary element can produce more accurate results for the capacitance calculations than the standard elements (shape functions of low order polynomials), and the *h*- mesh refinement method [92]. For the two examples studied here, the accuracy is shown to be better by more than one order of magnitude. In terms of electrostatics forces, the singular elements are also more accurate than the standard elements, though less significant compare to capacitance calculations. However, in this case, it is important to note that the singular elements give a much faster convergence rate with increasing number of elements than the standard elements.

Numerical solutions can often be improved, without increasing the problem size, by using biased elements, that is the *r*- mesh refinement method. This technique works more consistently for the standard elements, as the solutions are observed to improve when smaller elements are used near the singular regions. However, this is generally not the case when singular elements are used. It is observed that there exists an optimum bias ratio that produces possibly the most accurate solution for a given number of elements. This optimal situation is achieved when the size of the singular elements is comparable with the actual size of the singular regions. Hence, it is expected that the optimum bias ratio will vary with the proximity between conductors, as this determines the actual size of the singularity regions. It is also noted that the optimum bias ratio is different for the 8-node and 9-node

singular elements. From the biased element distribution study, it is observed that the 8-node elements tend to be less sensitive to the biasing effects than the 9-node ones. This additional feature, together with the obvious fact that 8-node elements are computationally cheaper than 9-node elements (one degree of freedom less for each element), with comparable accuracy, makes them more favourable than the 9-node elements.

The singular elements are also used in the electrostatics analysis of the electromechanical coupling simulations of some micro-devices. It is observed that the use of the quadratic-singular elements can give better results for the deflection profiles. In general, the standard elements tend to give smaller deformations than the singular elements. This indicates that ignoring the geometrical singularities (as in standard elements) is likely to underestimate the true displacements. However, the differences in the pull-in voltages are relatively small, as demonstrated in the micro-beam example.

6

Reviews of Fast Algorithms for BEM

Recent developments in fast algorithms have rekindled the interests in solving large problems using BEM, because of the linear growth in the computational complexities. These fast algorithms work by approximating the dense matrix-vector product, which is the key step in the projection type of iterative methods for solving the linear. The coefficient matrix is usually not formed explicitly, but approximated by sparse representation. This means that it is also cheaper in terms of computational storage. However, it is important to realize that the improvement in the computational efficiency is achieved by compromising the accuracy. This chapter reviews some of the existing fast algorithms.

6.1 Fast Multipole Method

Fast Multipole Method (FMM) was developed by Greengard and Rohklin [39, 44] for solving potential fields in particle systems in astronomy studies. Nabor and White [41-43, 45] then implemented the method in electrostatics analysis, mainly to calculate the capacitance of three-dimensional structures. The efficiency of FMM relies on the effective usage of the multipole and local expansions, which are employed repeatedly in a hierarchical manner through a series of translation operations. The following sub-sections briefly describe the essence of the algorithm. Readers are referred to [44, 45] for the detailed implementation of the method.

6.1.1 Multipole Expansion

Given a localized distribution of charges $s(\boldsymbol{x}')$, which is bounded within a sphere S_a of radius a, the potential it generates outside the sphere can be approximated by the following multipole expansion

$$\tilde{f}(\boldsymbol{x}) \approx \sum_{n=0}^{p} \sum_{m=-n}^{n} \frac{Y_n^m(q,f)}{R^{n+1}} M_n^m \tag{6.1}$$

where M_n^m are multipole moments, which are associated with their corresponding spherical harmonics $Y_n^m(q,f)$ and radial distant R from the centre of S_a to the potential point $\boldsymbol{x}$, and p is the order of the multipole expansion.

The multipole moments are defined as

$$M_n^m = \int s(\boldsymbol{x}')Y_n^{-m}(\theta', \phi')r'^n d^3x' \tag{6.2}$$

and $Y_n^m(\theta, \phi)$ is given by

$$Y_n^m(\theta, \phi) = \sqrt{\frac{(n-|m|)!}{(n+|m|)!}} P_n^{|m|}(\cos\theta)e^{im\phi} \tag{6.3}$$

where $P_n^m(\cos\theta)$ is the associated Legendre function of the first kind with degree *n* and order *m*, which is defined only when *n* is a nonnegative integer, and for $-n \le m \le n$. The error incurred by truncating the multipole expansion in (6.1) to order of *p* is bounded by [44, 45]

$$\left| f_{exact} - \tilde{f} \right| \le \left(\frac{Q_1}{R-a} \right) \left(\frac{a}{R} \right)^{p+1} \tag{6.4}$$

where $Q_1 = \int s(x')d^3x'$.

Beside multipole expansion, there also exist other expressions that can approximate far field potential due to some localized charges, such as the Poisson's formula (see for example [94], [95).

6.1.2 Local Expansion

Suppose there are some charges $q(x')$ distributed outside a sphere S_a of radius *a* centred at the origin, the potential at any point $\boldsymbol{x}$ within S_a due to $q(x')$ can be approximated by the local expansion as,

$$\tilde{f}(\boldsymbol{x}) \approx \sum_{j=0}^{p} \sum_{k=-j}^{j} L_j^k Y_j^k(\theta, \phi) r^j \tag{6.5}$$

where L_j^k are the local expansion coefficients, which are defined as

$$L_j^k = \int q(\boldsymbol{x}') \frac{Y_k^{-j}(\theta', \phi')}{R^{j+1}} d^3x' \tag{6.6}$$

and the error is similarly bounded, as in (6.4), by

$$\left| f_{exact} - \tilde{f} \right| \le \left(\frac{Q_2}{a-r} \right) \left(\frac{r}{a} \right)^{p+1} \tag{6.7}$$

Mathematically, local expansion is the Taylor series of the potential function generated by $q(x')$ in spherical coordinates, and L_j^k correspond to the potential and its gradients at the centre of S_a.

6.1.3 Translation Operators

Multipole and local expansions form the basis of FMM. However, the method relies on the ability to translate between the two expansions. Basically, there are three translation operators, namely:

(i) Multipole to Multipole (***M2M***) translation, which converts multipole moments M_n^m defined at (r, a, b) into multipole moments $\tilde{M}_n^m$ centred at the origin.

(ii) Multipole to Local (***M2L***) translation, which computes the local expansion coefficients L_j^k, that is, the potential and its gradients, due to the multipole moments M_n^m.

(iii) Local to Local (***L2L***) translation, which converts local expansion coefficients L_j^k defined at the origin into local expansion coefficients $\tilde{L}_j^k$ at some other point.

The translation formulas for these operators can be found in [38, 40, 41, 44, 45]. It is remarked that these operations scale with $O(p^4)$.

6.1.4 FMM algorithm

The algorithm begins with a hierarchical spatial decomposition of a computational cube that bounds the problem domain into successively smaller cells, where each cell is subdivided into 8 child cells. This results in a hierarchical oct-tree representation of the simulation domain, where level 0 is the root cube, and level L consists of 8^L cells.

At the lowest level, the distributed charges within each cell are converted to multipole moments located at the centre of the cell using (6.2). The multipole moments for all cells higher up the tree are then derived from the multipole moments of their child cells by using the ***M2M*** translation operator.

Next, at all levels, the local expansion coefficients, due to the multipole moments in the "interaction cells", are computed for all the cells through the ***M2L*** translation operator. In general, there are exactly at most 189 "interaction cells" for a given cell, which gives roughly $189p^4$ operations per cell. This translation process is the most expensive part of the algorithm, and hence different techniques were implemented to improve it. Greengard and Rokhlin [40] developed the new version of FMM, which scales approximately like $20p^3 + 189p^2$, by using the diagonal forms of translation operators with

exponential expansions. Further improvement was made by Cheng *et al.* [38] using the adaptive spatial decomposition algorithm. On the other hand, Elliott and Board [96] reduced the $O(p^4)$ scaling factor to $O(p^2 \log p)$ by performing FFT on the translation operators. However, it is noted this approach becomes numerically unstable for large p.

The local expansion coefficients for all the cells at the lowest level are then obtained by summing the local expansion coefficients from the higher level cells, which are translated down the hierarchy through the ***L2L*** translation operator. Finally, using the local expansion in (6.5) gives the potential at point $\boldsymbol{x}$, which only accounts for the "distant" charges effects. The "near" charge contributions are then added directly onto the potential point.

Alternatively, using multipole expansion alone can give rise to a simple fast algorithm, generally known as the tree algorithm [97, 98], which is $O(n \log n)$. The basic idea is very similar to FMM algorithm, except that local expansion is not used. Instead, the multipole expansion is evaluated directly on the potential point. Hence, to a certain extent, FMM can be seen as an enhancement of the tree algorithm.

6.2 Precorrected-FFT Approach

This method was developed by Phillips and White [46-48] for solving complicated three-dimensional electrostatics problems. It is motivated by the approximation scheme that enables one to represent an arbitrary distribution of charges by a small number of weighted point charges, which all lie on a uniform grid. Evaluating the potentials at the grid points due to the grid charges can then be seen as a discrete convolution, which can be performed efficiently using FFT algorithms. In general, it comprises the following steps:

(1) projecting the panel charges onto a uniform grid of point charges,

(2) computing the grid potentials due to the grid charges via FFT,

(3) interpolating the grid potentials onto the panels collocation points, and

(4) pre-correcting the interpolated potentials by replacing the inaccurate "near" charges contributions obtained via FFT with the ones that are computed directly.

6.2.1 Projecting arbitrary charge distribution onto a grid

The first step of this algorithm is to represent the panel charges $s(x)$ by a set of N_G point charges q_j, $j = 1 \ldots N_G$ that are positioned on a uniform grid. This can be done by matching potentials of the grid charges and distributed charges $s(x)$ at some pre-selected points. Another possible approach is based on matching the multipole moments directly [99]. Nevertheless, both the techniques result in performing singular value decomposition (SVD) on an over-determined system, whose pseudoinverse gives the linear transformation for mapping arbitrary charges $s(x)$ to grid point charges q_j.

6.2.2 Computing grid potentials by discrete convolution via FFT

Once the panel charges are projected to a grid, computing the potentials at the grid points due to the grid charges is a three-dimensional convolution. That is,

$$\hat{f}(i, j, k) = \boldsymbol{G}\hat{q} = \sum_{i', j', k'} g(i - i', j - j', k - k')\hat{q}(i', j', k') \tag{6.8}$$

where $\hat{f}(i, j, k)$ and $\hat{q}(i', j', k')$ are the grid point potential and grid point charge at position (i, j, k) and (i', j', k'), respectively, and $g(i - i', j - j', k - k')$ is the Green's function, which is essentially the inverse distance between the grid points (i, j, k) and (i', j', k'). The key to the efficiency of precorrected-FFT lies on the fact that the discrete convolution in (6.8) can be computed rapidly by using FFT algorithms [49].

6.2.3 Approximating potentials by interpolating grid potentials

Grid potentials are then interpolated onto all the panel's collocation points. One simple approach is to use a polynomial interpolation over several grid points. Alternatively, an operator that interpolates potentials at grid points onto panel's collocation points can be obtained by following the similar idea of representing charges on the grid.

6.2.4 Precorrecting the approximated potentials

This step is required because the grid charges do not accurately approximate the "near" interactions. Hence, the task here is to replace the inaccurate contributions from the grid charges that were included through the convolution process by the ones that are computed accurately.

In summary, precorrected-FFT algorithm approximates the dense matrix-vector product, as given by

$$f = \boldsymbol{P}q \quad \Rightarrow \quad f \approx \left[\tilde{\boldsymbol{P}} + \boldsymbol{V}^T \boldsymbol{G} \boldsymbol{W}\right] q \tag{6.9}$$

where $\tilde{\boldsymbol{P}}$ is from the precorrecting step, $\boldsymbol{V}$ and $\boldsymbol{W}$ are the potential interpolation and grid charge projection operators, respectively, and $\boldsymbol{G}$ is discrete Fourier transform matrix. And all the matrices possess sparse representations.

6.3 Matrix Sparsification Techniques

6.3.1 Wavelet based method

The idea of employing wavelet bases to build sparse versions of discretized boundary integral operators was introduced by Beylkin *et al*. [100]. It was later used by Spasojevic *et al*. [101, 102] to solve two-dimensional electrostatics problems. In their works, the orthogonal [101] and bi-orthogonal [102] Haar wavelets were used as the basis for representing the charge distributions, that is, the surface charges on each boundary element is approximated by

$$s(x) \approx \sum_{l=-1}^{L} \sum_{k=0}^{K} s_{lk} v_{lk}(x) \tag{6.11}$$

where s_{lk} and $v_{lk}(x)$ are the wavelet coefficient and wavelet basis, respectively, and L is order of resolution, with K = max(0, 2^l-1) translated functions at level l resolution. Sparsifying the fully populated matrix can be achieved simply by ignoring the "small" entries in the matrix. This elimination process is often based on the distance criterion, that is, the distance between the source and field points. Other works that were based on the wavelet approach can be found in [103-105].

6.3.2 Singular Value Decomposition

Another approach of sparsifying a dense matrix is through the use of Singular Value Decomposition (SVD) methods [106, 107], which exploits the fact that a large part of the dense matrix is numerically low rank. The algorithm first adaptively partitions the matrix into low rank submatrices, via divide and conquer, and then applies SVD on these submatrices to obtain a sparse representation of the original dense matrix.

7

Fast Fourier Transform on Multipoles (FFTM)

In this part of the thesis, we propose a fast algorithm for the rapid solution of the integral equation in BEM. This method arises from an important observation that potential evaluation using multipole expansion can be expressed as a series discrete convolution of the multipole moments with their associated spherical harmonics functions. FFT algorithms can be employed to evaluate the discrete convolutions rapidly, and this essentially provides the efficiency of this approach. We refer it to as the Fast Fourier Transform on Multipoles (FFTM) method.

This chapter is organized as follows. The FFTM algorithm is first presented in Section 7.1. Some important issues regarding its implementation are also adressed. This is then followed by its complexity analysis in Section 7.2. In Section 7.3, some numerical examples are presented to illustrate the performance of the method, in terms of its accuracy and efficiency. It is demonstrated that FFTM is an accurate method, and has only linear growth in the computational complexity, which implies that it can as efficient as other fast algorithms, such as FMM and precorrected FFT.

7.1 FFTM Algorithm

This algorithm generates a number of transformation matrices that are denoted by three-letter abbreviations, following the works of Nabors and White in [45]. The letter notations have the following meanings: ***M*** = Multipole moments, ***Q*** = Charge, ***P*** = Potential and ***2*** = To. Basically, the algorithm comprises the following five steps:

(1) discretization of spatial domain into many smaller cells,

(2) representation of the panel charges by multipole moments for all cells (***Q2M***),

(3) evaluation of the potentials at cell centers due to the multipole moments, through discrete convolutions that is accelerated by FFT algorithms,

(4) interpolation of the cell potentials (due to “distant” charges) onto collocation points on the panels (***P2P***),

(5) inclusion of potential contributions from the "near" charges directly onto the panels (***Q2P***).

This process is summarized in Figure 7.1. The following sub-sections elaborate on each of the steps.

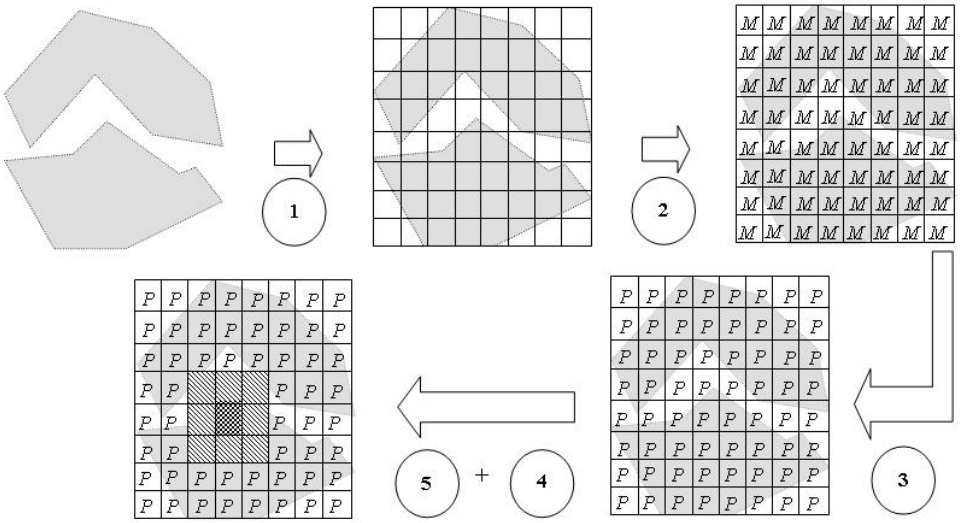


Figure 7.1. 2D pictorial representation of FFTM algorithm. Step (1): Division of problem domain into many smaller cells. Step (2): Computation of multipole moments for all cells. Step (3): Evaluation of potentials at cell centers by convolutions via FFT. Step (4): Interpolation of cell potentials onto panel collocation points. Step (5): Inclusion of near charges contributions (panels within the shaded region) directly onto panels.

7.1.1 Spatial discretization

This step divides the problem domain into many smaller cells, and allocates the panels among them. The aim is to identify closely pack panels that can be approximated by simpler representations, such as multipole moments in FMM or grid point charges in precorrected-FFT. It also helps to separate the "near" panels and the "distant" ones. Unlike FMM, the initial volume that bounds the problem domain need not be a cube, since the hierarchical partitioning of the cells is not needed. The dimensions of the volume only need to satisfy the ratio required by the FFT solvers, which is usually in powers of two. Otherwise, dummy layers of empty cells have to be added to meet the requirement. This process is commonly known as zero padding. Nowadays, it is possible to perform FFT on any arbitrary size with the help of the freeware FFTW (Fastest Fourier Transform in the West), provided by Frigo and Johnson[†]. This improves the efficiency of FFTM by minimizing the number of zero padding, and hence avoiding the unnecessary increase in size of the FFT array.

† ***FFTW***, *C subroutines library for computing Discrete Fourier Transform (DFT).* *The* freeware can be downloaded from http://www.fftw.org.

7.1.2 Transformation of panels charges to multipole moments

In this step, the arbitrary distribution of charges $s(x')$ within a given cell is represented by an equivalent system of sources that are positioned at the centre of the cell. These sources include a point charge, an electric dipole, and other higher multipole moments, which can be obtained using (6.2), and rewritten here for convenience,

$$M_n^m = \int s(\boldsymbol{x}') Y_n^{-m}(\theta', \phi') r'^l d^3 x' \tag{7.1}$$

Equation (7.1) is essentially the ***Q2M*** transformation function. Applying ***Q2M*** to all the cells transforms the boundary element discretized problem to one that contains point sources that are regularly spaced, as depicted in Figure 7.1 after Step 2.

7.1.3 Evaluation of potentials at cells centres using FFT

This step is to evaluate the potentials at the cells' centres due to the effects of the multipole moments in all the cells. The regular spacing of the cell centres enables this potential evaluation process to be expressed as discrete convolutions, which can be done rapidly using FFT. Mathematically, the potential calculations using multipole expansions can be written as a series of three-dimensional discrete convolutions as

$$\tilde{f}(i, j, k) \approx \sum_{n=0}^{p} \sum_{m=-n}^{n} \left[\sum_{i'} \sum_{j'} \sum_{k'} M_n^m(i', j', k') \frac{Y_n^m}{R^{n+1}}(i - i', j - j', k - k') \right] \tag{7.2}$$

where the indices (i, j, k), and (i', j', k') denote the discrete locations of the field points and multipole moments, respectively, and there are exactly $(p+1)^2$ discrete convolutions. In order to eliminate the aliasing effects completely, the convolution size needs to be increased by eight times with zero padding.

7.1.4 Evaluation of potentials at panels' collocation points

Once the potentials at the multipole cell centers are determined, they must be interpolated onto the actual collocation nodes on the panels. Consider a collocation point $\boldsymbol{x}$ that falls in cell k, its potential is the sum of the contributions from the "distant" and "near" charges, that is,

$$\tilde{f}_k(\boldsymbol{x}) \approx \sum_{i=1}^{N_d} \tilde{f}_k^{\,i}(\boldsymbol{x}) + \sum_{j=1}^{N_n} \tilde{f}_k^{\,j}(\boldsymbol{x}) \tag{7.3}$$

where N_d denotes the cells that are considered far away from cell k, whereas N_n corresponds to the near-neighbors of cell k, which are usually defined to be either the first nearest neighboring cells (27 cells), or first and second nearest neighboring cells (125 cells). The rest of the cells in the domains are considered "distant". Hence, the two sets of cells are mutually exclusively, that is, $N_d + N_n = N_c$, where N_c is the total number of cells.

The "distant" charges contributions, which are approximated by multipole expansion, can be determined by interpolating the cells' potentials (obtained in Step 3) onto the collocation point $\boldsymbol{x}$. In this study, the simple quadratic interpolation method is adopted. Basically, the idea is to form a potential interpolation element with the nearest neighboring cells. Hence, a three-dimensional quadratic interpolating function has 27 potential points, as given by

$$\tilde{f}(x, y, z) = \sum_{i=1}^{27} N_i(x, y, z)\tilde{f}_i \tag{7.4}$$

where $N_i(x, y, z)$ is the set of quadratic Lagrange interpolation functions, and $\tilde{f}_i$ is cell's potential. Equation (7.4) is the transformation function for ***P2P*** matrices.

However, prior to that, we need to perform a potential correction step, which is essential because the cell potentials obtained through the discrete convolutions have inevitably included the "near" charges contributions that are inaccurately represented by multipole expansions. The potential correction scheme adopted here is rather straightforward, as illustrated in Figure 7.2 for the two-dimensional case.

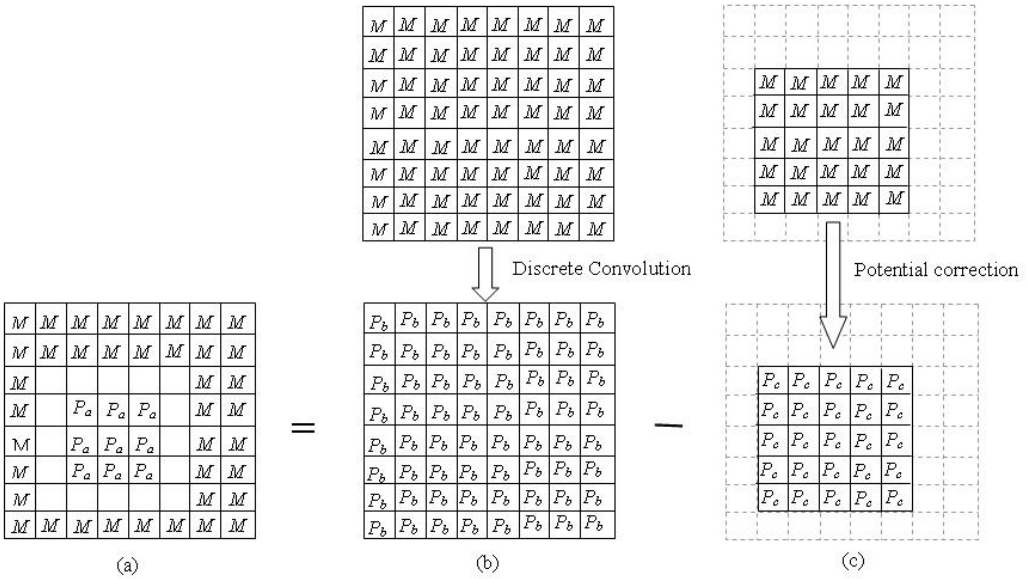


Figure 7.2. (a) Potentials at nine interpolation cells, which account for effects of distant charges only. This is given by the difference of potential due to (b) convolution corresponding to set N_c and, (c) convolution corresponding to set N_n.

Figure 7.2(a) shows the desired situation where the potentials at the nine interpolation cells include the effects of the "distant" charges only. This is achieved by taking the difference of the potentials calculated from the two discrete convolutions, as depicted in Figures 7.2(b) and 7.2(c), respectively. The former convolution is that described in Section 7.1.3. On the other hand, the second convolution evaluates the potentials at the same interpolation points, but due to the charges in the near-neighboring cells only. We refer to this convolution as the potential correction step, which will be discussed further in Section 7.1.5. Hence, by using the corrected interpolation potentials in (7.4), the "distant" charges contributions can be computed.

The second component of (7.3), which accounts for the "near" charges effects, is identical to the direct pass in FMM [41, 45]. In this case, the potential contributions from the panels that falls within N_n cells are evaluated directly onto the collocation point. The transformation matrices that perform this task are denoted by ***Q2P***.

7.1.5 Potential correction step

This step calculates the potentials at the 27 interpolation points due to the multipole moments in the N_n cells. Although this can again be done rapidly by discrete convolution using FFT, it will be shown shortly that the direct approach is more efficient. The following discussion compares the efficiency of the two approaches (in terms of real multiplication operations), for situation where N_n = 125.

Suppose this task is done by discrete convolution using FFT, the operation counts can be approximated as follows. First, one needs to perform two FFTs and a complex multiplication of Fourier coefficients of size m, where the minimum size of m is 7x7x7 = 343, since at least 2 zero paddings are required in each direction. However, it is more efficient to let m to be 8x8x8 = 512, because of its high efficiency with FFT algorithms. A detail study of the FFT algorithms reveals that this FFT can be done with only 128 real multiplication operations, by fully utilizing the twiddling factors [49]. As for the complex multiplication of the Fourier coefficients, there are exactly 256 of them. However, due to the symmetry of the response functions, whose Fourier transforms are either purely real or imaginary, each complex multiplication is reduced from four real multiplication operations to two. Hence, the total number of real multiplication operations needed to perform this potential correction step through discrete convolution is 2x128 + 2x256 = 768.

On the other hand, evaluating this convolution directly requires 27x124 = 3348 operation counts, since there are 27 interpolation points, and for each point there are 124 cells contributions to be considered. The singular contributions due to the self multipole moments are set to zero, because they are to be accounted for exactly in the "near" contributions. It appears that the direct approach is computationally more expensive than the convolution approach. However, two simple techniques can be applied to reduce the direct approach cost significantly.

The first technique involves setting the response functions at the nearest neighbours to zero, that is,

$$\frac{Y_n^m}{R^{n+1}}(i-i', j-j', k-k') = 0, \quad \text{for } |i-i'|, |j-j'| \text{ and } |k-k'| \le 1 \tag{7.5}$$

Using (7.5) naturally excludes the effects of the multipole moments of the nearest neighbouring cells, whose contributions are to be computed exactly by using ***Q2P***. This reduces the number of cells to be considered for each interpolation point to 125-27 = 98. Hence, the total operation counts reduces to 27x98 = 2646.

The second technique makes use of the symmetry of the response functions. This allows us to exchange many of the multiplication operations with additions. It is noted that the gain from this technique hinges on the number of distinct response function values that are associated with each interpolation point. This is summarized in Table 7.1 for the various response functions (up to p = 2) and the 27 interpolation points. On average, applying this technique reduces the number of multiplication operations to about 20 % of the original counts, that is, from 2646 to about 500. Hence, applying these techniques make the direct approach more favourable in performing the potential correction step.

7.1.6 Remarks on the use of local expansion

The use of local expansion to compute potentials at the panel collocation points, as used in FMM, may be a more intuitive and desirable approach of implementing FFTM. This approach would enable the method to attain arbitrary high order of accuracy, which is not possible with the quadratic interpolation method.

However, there are some practical issues that hinder the implementation of local expansions. First, the complicated multipole to local expansion transformation makes it difficult to implement. More

importantly, the number of discrete convolutions scales with $O(p+1)^4$, which means that the computational cost (both in the time and memory storage requirements) increases dramatically with increasing value of p. In other words, the improvement in the accuracy is achieved at a very high computational cost. Hence, further investigations are needed to study the efficiency of this approach. For now, the simple quadratic interpolation scheme is used.

Table 7.1. Number of distinct values for different response functions at different potential interpolation points.

Interpolation points	**Number of distinct values for response functions, $\frac{Y_n^m}{R^{n+1}}$ (n, m)**								
	(0,0)	**(1,0)**	**(1,1)**	**(1,-1)**	**(2,0)**	**(2,1)**	**(2,-1)**	**(2,2)**	**(2,-2)**
1	15	27	27	27	32	22	22	22	22
2	15	27	27	27	32	22	22	22	22
3	15	27	27	27	32	22	22	22	22
4	15	27	27	27	32	22	22	22	22
5	15	27	27	27	32	22	22	22	22
6	15	27	27	27	32	22	22	22	22
7	15	27	27	27	32	22	22	22	22
8	15	27	27	27	32	22	22	22	22
9	14	24	17	24	29	18	16	22	18
10	14	24	24	17	29	16	18	22	18
11	14	24	17	24	29	18	16	22	18
12	14	24	24	17	29	16	18	22	18
13	14	24	17	24	29	18	16	22	18
14	14	24	24	17	29	16	18	22	18
15	14	24	17	24	29	18	16	22	18
16	14	24	24	17	29	16	18	22	18
17	14	17	24	24	23	18	18	16	16
18	14	17	24	24	23	18	18	16	16
19	14	17	24	24	23	18	18	16	16
20	14	17	24	24	23	18	18	16	16
21	6	9	9	9	11	7	7	7	7
22	11	15	15	15	17	13	13	10	10
23	11	15	15	15	17	13	13	10	10
24	11	15	15	15	20	10	13	16	13
25	11	15	15	15	20	13	10	16	13
26	11	15	15	15	20	10	13	16	13
27	11	15	15	15	20	13	10	16	13
Total	**360**	**575**	**575**	**575**	**705**	**463**	**463**	**507**	**463**
(Total/2646) %	**13.6**	**21.7**	**21.7**	**21.7**	**26.6**	**17.5**	**17.5**	**19.2**	**17.5**

7.2 Algorithmic Complexity Analysis

This section gives estimates on the time and memory complexity for the FFTM algorithm. The analysis looks at two parts, namely at the initialization and iteration stages. At initialization stage, we are mainly concerned with the memory complexity required to store the various transformation

matrices, whereas at the iteration stage, we are interested in the time complexity. The overall computational complexities of FFTM algorithm are summarized in Table 7.2.

Table 7.2. Time and memory complexities of FFTM algorithm.

STAGES	OPERATION	COMPUTATIONAL TIME	MEMORY STORAGE
Initialization	Computing ***Q2M, P2P*** and ***Q2P*** matrices	$O\left((p+1)^2 n\right) + O(27n) + O(N_n k_c n)$	$O\left((p+1)^2 n\right) + O(27n) + O(N_n k_c n)$
	Computing response functions and their FFTs	$O\left((p+1)^2 N_c\right) + O\left((p+1)^2 N_c \log N_c\right)$	$O\left((p+1)^2 N_c\right)$
Total cost at Initialization stage		$O\left((p+1)^2\left[n + N_c + N_c \log N_c\right] + (27 + N_n k_c)n\right)$	$O\left((p+1)^2(n+N_c) + (27+N_n k_c)n\right)$
Iteration	Computing multipole moments via ***Q2M***	$O\left((p+1)^2 n\right)$	
	Computing cells' potentials by convolutions	$O\left((p+1)^2[(8N_c) + (16N_c)\log(8N_c)]\right)$	$O(17N_c)$
	Potential correction and interpolation via ***P2P***	$O\left((p+1)^2 5.4 N_n N_c F_c + 27n\right)$	
	Computing "near" interactions via ***Q2P***	$O(N_n k_c n)$	
Total cost at Iteration stage	Perform matrix - vector products for K_{iters} times	$K_{iters} \times O\left((p+1)^2\left[n + (16N_c)\log(8N_c) + (8N_c) + 5.4 N_n F_c N_c\right] + (27 + N_n k_c)n\right)$	$O(17N_c + n(1+K_{iters}))$

Definitions:
n is the problem size.
p is the multipole expansion order.
N_c is the total number of mutipole cells after the spatial discretization step.
N_n is the number of cells that are in the direct interaction list, either 27 or 125.
k_c is the average number of panels in a cell.
F_c is a sparsity factor that defines the ratio of the non-empty cells to N_c.
K_{iters} is the number of iterations required to achieve the desire accuracy.

7.2.1 Complexity at Initialization stage

The main computational cost of this initialization stage is due to the formations of the various transformation matrices, which include ***Q2M***, ***P2P*** and ***Q2P***, and also computation of the response functions $\frac{Y_n^m}{R^{n+1}}$ and their Fourier transforms.

The complexities for computing and storing ***Q2M***, ***P2P*** and ***Q2P*** are $O\left((p+1)^2 n\right)$, $O(27n)$ and $O(N_n k_c n)$, respectively, where k_c denotes the average number panels in one cell. The constant factor of 27 in the complexity of ***P2P*** is due to the quadratic interpolation scheme used for the interpolating

potential. Computing the response functions and their Fourier transforms have complexities $O\left((p+1)^2 N_c\right)$ and $O\left((p+1)^2 N_c \log N_c\right)$, respectively. Note that zero padding is avoided here due to the symmetries of the response functions. Finally, the total time and memory complexities at the initialization stage are

$$\text{Time} = O\left((p+1)^2\left[n + N_c + N_c \log N_c\right] + (27 + N_n k_c)n\right)$$

$$\text{Memory} = O\left((p+1)^2 (n + N_c) + (27 + N_n k_c)n\right)$$

It is important to note that the dominant cost at this stage is likely to be due to the computing and storing of ***Q2P*** matrices, since $N_n k_c$, which is equal to $27k_c$ or $125k_c$, is usually larger than $(p+1)^2$.

7.2.2 Complexity at iteration stage

At the iteration stage, the main concern is the time complexity as it is the primary factor that determines the efficiency of this algorithm. The memory complexity is considerably less as compared to that at the initialization stage. The major memory requirements are:

(i) Two matrices of sizes $8N_c$ and N_c. The former is used for the Fourier Transform in the discrete convolution step, while the later one stores the approximated potentials obtained.

(ii) One matrix for storing the basis vectors generated by the GMRES at each iteration. Normally, the memory required is $O(nK_{iters})$, where K_{iters} is the number of iterations for the solution to converge to the desire accuracy. However, this can be constrained by using the restart GMRES [37].

Time complexity at the iteration stage comprises of the following components:

(i) $O\left((p+1)^2 n\right)$ operations to compute $(p+1)^2$ multipole moments using ***Q2M***, $O(27n)$ operations to interpolate potentials using ***P2P***, and $O(N_n k_c n)$ operations to compute the "near" charges interactions using ***Q2P***.

(ii) $O\left((p+1)^2\left[2(8N_c \log 8N_c) + 8N_c\right]\right)$ operations to compute $(p+1)^2$ discrete convolutions, each requiring two FFTs and one complex multiplication of size $8N_c$.

(iii) $O\left((p+1)^2\left[0.2(27N_n)F_c N_c\right]\right)$ operations to perform the potential correction step, where F_c is a sparsity factor that defines the ratio of the non-empty cells to N_c (since this step is only applied

for non-empty cells), and the 0.2 factor accounts for the speedup due to the two techniques described in Section 7.1.5.

Hence, the total time and memory complexities for one iteration is given by

$$\text{Time} = O\left(\begin{matrix}(p+1)^2\left[n+(16N_c)\log(8N_c)+(8N_c)+5.4N_nF_cN_c\right]\\+(27+N_nk_c)n\end{matrix}\right)$$

$$\text{Memory} = O(17N_c + nK_{iters})$$

7.3 Numerical Examples

In this section, some numerical examples are used to study the performance of FFTM. The study comprises of an accuracy analysis and an efficiency analysis (in terms of computational speed and memory requirements). Different FFTM schemes are characterized by two parameters, namely, (i) the direct interaction list D_{list}, and (ii) the multipole expansion order p. D_{list} = 1 when only the nearest neighbors are used for the direct interactions, and D_{list} = 2 when the first and second nearest neighbors are used for the direct interactions. The parameter p takes value of 0, 1 or 2. The combinations of the two parameters give a total of six FFTM schemes. All the analyses are done on a HP C3600 workstation with 1 *GB* of RAM.

7.3.1 Accuracy analysis of FFTM

The accuracy of FFTM is gauged against the solutions that are obtained using the GMRES explicit method, where the full coefficient matrix is formed explicitly. Four examples are used in this accuracy analysis. These include, (i) the self-capacitance extraction of a cube, (ii) the electrostatics force analysis of a cube that is placed over a ground plane, (iii) the electromechanical coupling analysis of the comb-levitation problem, and (iv) the 4x4bus-crossing example [41]. The first three examples are also used in the singular elements analysis in Chapter 5.

Four different types of boundary elements are used here, namely the constant, linear, quadratic and quadratic-singular, where their names denote the order of the polynomials that represent the surface charge density on the elements. For the quadratic-singular element type, singular elements are used in place of the standard quadratic elements at sharp corners and edges.

Effects of spatial discretization on accuracy of FFTM

It is believed that the spatial discretization can have significant effects on the accuracy of the solution. By using finer cell discretization, the accuracy of the "distant" charges contributions improves, because the separation distance ratios $\left(\frac{a}{R}\right)$ for the multipole expansion are smaller now. However, the number of panel charges that are to be treated exactly in the "near" charges component is reduced, which results in a loss of accuracy. Hence, it is desirable to study this effect on the different FFTM schemes.

The cube self-capacitance extraction problem is used for this study, and the cube is meshed with 64 uniform constant boundary elements on each face. The "exact" solution is 73.033 *pF*, and the results are tabulated in Table 7.3.

Table 7.3. Capacitance of cube example using different cell discretizations for different FFTM schemes.

Cells discretization (n_x x n_y x n_z)	**D_{list} = 1**			**D_{list} = 2**		
	p = 0	**p = 1**	**p = 2**	**p = 0**	**p = 1**	**p = 2**
5 x 5 x 5	74.963	73.101	72.968	73.990	73.048	73.031
8 x 8 x 8	71.679	73.995	73.049	71.894	73.014	73.035
12 x 12 x 12	73.661	73.143	73.005	73.382	73.100	73.038

The results are generally quite accurate, giving errors less than 5 %, for all the schemes. However, it is noted that for the lower order schemes, the results fluctuate quite significantly with different kind of spatial discretizations. However, the degree of fluctuation decreases when higher order multipole expansion is used. This behavior is probably due to the fact that the higher order multipole moments can represent the charge distributions within the cells more accurately. Monopole (p = 0) moment simply approximates the charge distributions within each cells by a point charge at the cell's center, which has magnitude equal to the sum of the charges within the cell. This means that it does not account for the charge distributions within the cell. On the other hand, the dipoles (p = 1) and the quadrupoles (p = 2) moments can model the first and second derivatives of the charge distributions, respectively. Hence, this makes the multipole expansion less sensitive to the cell discretization step. As the lower order FFTM schemes (p = 0) are too sensitive to the cell discretization, they are not used in the subsequent analyses.

Self-capacitance extraction of cube example

This example is identical to the one used above, with spatial discretization fixed at 8x8x8. The capacitance results are tabulated in Table 7.4.

Table 7.4. Capacitance extraction of cube example, for different FFTM schemes and different types of elements.

Element type	GMRES explicit	FFTM			
		D_{list} = 1		D_{list} = 2	
		p = 1	p = 2	p = 1	p = 2
Constant	73.033	72.995	73.049	73.014	73.035
Linear	74.194	73.916	73.945	73.976	73.981
Quadratic	73.716	73.715	73.762	73.726	73.734
Quadratic-singular	73.503	73.495	73.541	73.508	73.520

Generally, the results are very accurate for all the schemes, with errors less than 1 %. An important observation is noted by comparing the results of the FFTM schemes that have the identical p value but different D_{list} value. For example, consider the results in column 4 (D_{list} = 1 and p = 2) and 6 (D_{list} = 2 and p = 2) of Table 7.4. In this case, one expects the first column of results to be less accurate, because the second nearest neighboring cells are approximated by the multipole expansions that tend to be less accurate. However, the two sets of results are not significantly different. In fact, they differ by less than 0.1 %. Similar observations can be seen for the results between column 3 and 5. This observation may be due to the following reason.

First, the difference in the two situations is the treatment of the second nearest neighboring cells. For schemes with D_{list} = 1, their effects are approximated by multipole expansions, while those with D_{list} = 2 accounts for them exactly in the direct interaction list. Although the multipole approximations are less accurate, their effects on the overall solution may not be significant. This is simply because their potential contributions may only be a small portion of the total potential contributions from all the cells, since this layer of neighbouring cells (98 of them) usually forms only a small fraction of the total number of cells. Hence, their error contributions are also expected to be small. This suggests that the FFTM schemes with D_{list} = 1 should be used over those with D_{list} = 2, by virtue that they are less expensive, and only slightly less accurate.

It is also observed that FFTM only approximates the corresponding "exact" solutions for the different types of elements used. In other words, this method is capable of retaining the effects of the element shape functions, through the used of ***Q2M*** (multipole moments for "distant" charges contributions) and ***Q2P*** (for "near" charges contributions) transformation matrices. This is especially important for the quadratic-singular element type, because it is desirable that the accuracy of the singular elements be retained when approximating the dense matrix-vector products by these transformations.

Electrostatics force on cube over ground plane

This example is also identical to the one used in Section 5.5.2, except that the ground plane is not infinite. In this case, the ground plane is assumed to be three times larger than the unit cube, and the cube is placed at distance of 0.5 unit above the ground plane. The resultant electrostatics force acting on the cube's face that is just above the ground is computed. The spatial discretization here is 16x16x8, and the results are given in Table 7.5.

Table 7.5. Electrostatics force on cube's surface, for different FFTM schemes and different types of elements.

Element type	**GMRES explicit**	**FFTM**			
		D_{list} **= 1**		D_{list} **= 2**	
		p **= 1**	p **= 2**	p **= 1**	p **= 2**
Constant	3.922	3.901	3.855	3.928	3.916
Linear	4.666	4.691	4.674	4.664	4.665
Quadratic	4.654	4.643	4.672	4.661	4.650
Quadratic-singular	4.753	4.751	4.782	4.762	4.750

In this case, the results are generally less accurate than the previous example. As electrostatics force is proportional to the square of the surface charge density, the error is likely to be twice that in the capacitance calculation.

Comb-finger levitation example

This example is an electromechanical coupling analysis of comb fingers as described in Section 5.5.3. We examine two cases, where the applied voltage is 100 *V* and 200 *V*, respectively. Only the constant and quadratic-singular element types are used here. The cell discretization is 20x10x2, and the results are given in Table 7.6.

Table 7.6. Maximum deflection of central comb-finger, for different FFTM schemes and different types of elements.

Element type	Applied voltage (V)	GMRES explicit	FFTM			
			D_{list} = 1		D_{list} = 2	
			p = 1	p = 2	p = 1	p = 2
Constant	100	0.03266	0.03118	0.03259	0.03139	0.03267
	200	0.1185	0.1136	0.1176	0.1142	0.1179
Quadratic-singular	100	0.04585	0.04525	0.04547	0.04638	0.04560
	200	0.1589	0.1607	0.1577	0.1611	0.1581

The results are again accurate for all the various schemes. For the schemes with p = 1, the error is 1 – 4 %, while for p = 2, it is less than 1 %. The deflection of the comb-finger depends greatly on the computed force distributions, which in turn depends on the surface charge distributions. Hence, it is no surprise that the higher order schemes (p = 2) can produce more accurate solutions, since they can approximate the actual charge distributions more accurately than the lower order schemes (p = 1).

4x4 bus-crossing example

In this example, taken from Nabor and White [41], the capacitance matrix of a 4x4 bus-crossing example, as shown in Figure 7.3, is computed. For consistent comparison with the results in [41], only the FFTM schemes with D_{list} = 2 are used. The cell discretization used is 10x10x3. The results of the FMM from [41] are also duplicated in Table 7.7, followed by the results of the FFTM schemes.

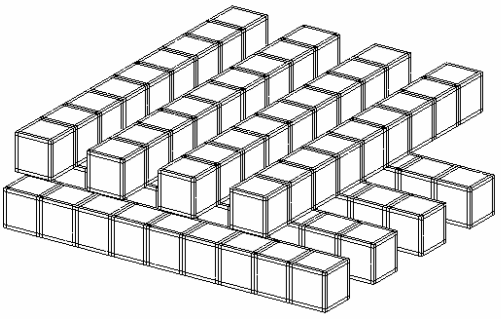


Figure 7.3. 4x4 bus-crossing example from [41]. Conductors are meshed as close to the original work as possible.

It is observed that FFTM is generally more accurate than FMM. This is especially obvious for the lower order schemes (p = 0, 1) and for the off-diagonal capacitance entries. The significant improvement in the accuracy is largely due to the ways the distant potential contributions are computed in the two methods.

Table 7.7. Capacitance extraction of 4x4 bus-crossing example by FMM from [41], and FFTM methods.

Solution Method	Capacitance Matrix Entry (pF)							
	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}
GMRES explicit	402.9	-136.2	-12.00	-7.886	-48.18	-39.90	-39.90	-48.18
FMM (p = 0)	394.5	-124.0	-0.175	-2.471	-52.15	-43.39	-43.08	-52.92
FMM (p = 1)	406.6	-139.7	-12.36	-6.676	-48.48	-40.45	-40.27	-48.46
FMM (p = 2)	405.2	-137.8	-11.91	-8.079	-48.36	-40.09	-40.01	-48.45
FFTM (p = 0)	404.2	-133.1	-13.53	-6.108	-49.14	-41.53	-41.27	-49.85
FFTM (p = 1)	403.4	-136.7	-12.57	-8.014	-48.15	-39.63	-39.62	-48.05
FFTM (p = 2)	403.2	-136.3	-11.49	-7.966	-48.36	-40.05	-40.05	-48.34

In FMM, multipole and local expansions are used in a hierarchical fashion to approximate the distant potential fields. This hierarchical approach tends to introduce more approximations when multipole moments and local coefficients are passed upwards and downwards in the hierarchy in the algorithm. On the other hand, FFTM replaces this hierarchical process by using FFT algorithm to evaluate the discrete convolutions of the multipole expansion. In this case, the cell size to distance ratio $\left(\frac{a}{R}\right)$, which determines the accuracy of the multipole expansion (see equation (6.4)), are smaller than those in FMM. This point is more clearly illustrated in Figure 7.4 for the two-dimensional case.

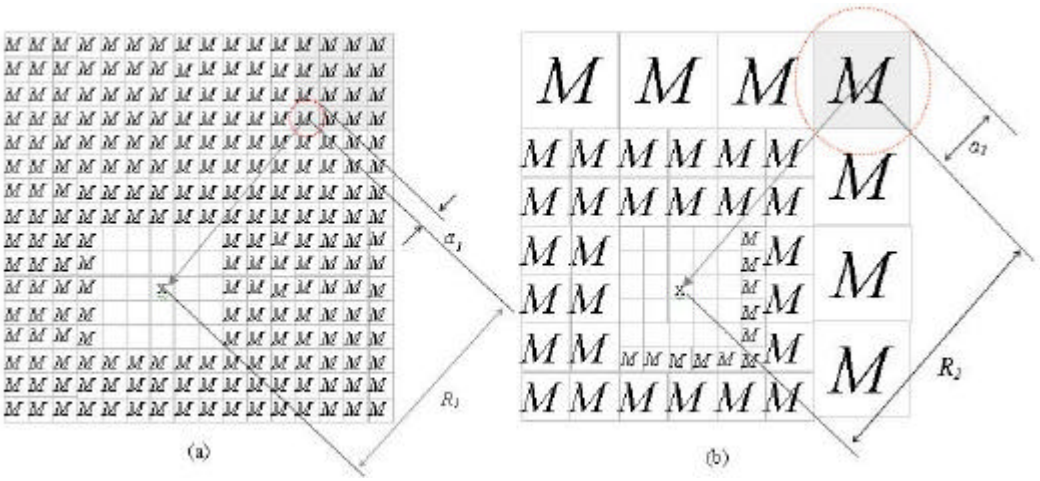


Figure 7.4. Comparison on accuracy of (a) FFTM and (b) FMM, based on cell to distance ratio.

Suppose we want to evaluate the potential at point x, due to the surrounding multipole moments, which corresponds to the "distant" charges contributions in (7.3). The obvious approach is to compute all the multipole moments effects directly, but this may be computationally too expensive. FFTM does it in a more efficient way by recognizing that this potential evaluation task can be seen as discrete

convolutions of the multipole moments with their associated response functions, which can be evaluated rapidly using FFT algorithms. On the other hand, FMM uses a number of transformations to reduce the computational cost, which involve passing multipole and local expansions in a hierarchical manner. This process results in multipole moments representation that contains cells of different sizes, such as the one depicted in Figure 4(b). Each of the coarsest cells in Figure 4(b) corresponds to 16 cells at the finest level from Figure 4(a). As noted in (6.4), the accuracy of the multipole expansion depends on the multipole order p, and the separation distant ratio $\left(a/R\right)$. For a given multipole expansion order, it is easily seen in Figure 4 that FFTM is likely to be more accurate than FMM, since the coarser cells in FMM have larger $\left(a/R\right)$ ratios than the corresponding finer cells they originally represent. Although, this argument would be absolutely true only if the local expansion is used for FFTM, we believe that it still holds, for low order expansion ($p \leq 2$), since the quadratic interpolation functions resemble the second order local expansion. Through this simple error analysis, it is demonstrated that potential evaluation through convolutions in FFTM is likely to be more accurate than that obtained by using the hierarchical approach in FMM.

7.3.2 Efficiency analysis of FFTM

This section studies the efficiency of FFTM, in terms of the computational speed (CPU time) and memory storage requirements. Although it is preferable to compare FFTM with the existing fast methods, it is not done here because we are not familiar with the implementations of FASTCAP (FMM based program by Nabors and White [45]) and FFTCAP (pre-corrected FFT based program by Phillips and White [48]). In this study, the comparisons are made only with respect to the GMRES explicit method. Only the constant element and FFTM schemes with $p > 0$, are used here.

Effects of spatial discretization on efficiency of FFTM

Spatial discretization is also expected to affect the efficiency of the method. The effects come in two ways, namely when evaluating the discrete convolutions, and computing the "near" charges contributions via ***Q2P***. For finer cell discretization, the cost of evaluating the discrete convolutions obviously increases, but computing the "near" charges' effects become less expensive now, since the number of panels in the direct interaction lists decrease.

The cube example is used here for two kinds of cell discretization, namely 8x8x8 and 12x12x12. Only the schemes (D_{list} = 1, p = 1) and (D_{list} = 2, p = 2) are used to investigate the spatial discretization effects. The cube is meshed with uniform elements, and larger problems are generated by using a finer element mesh. Plots of the CPU time and memory storage requirements are shown in Figure 7.5.

The two sets of plots are observed to be similar. The 12x12x12 spatial discretization seems a little more efficient, because the gradients of the associated curves, as shown in Figure 7.5, are slightly gentler than that for the 8x8x8 case. This means that its computational costs grow slower with increasing size of the problem. However, for smaller problems, the 8x8x8 spatial discretization is noted to be more efficient.

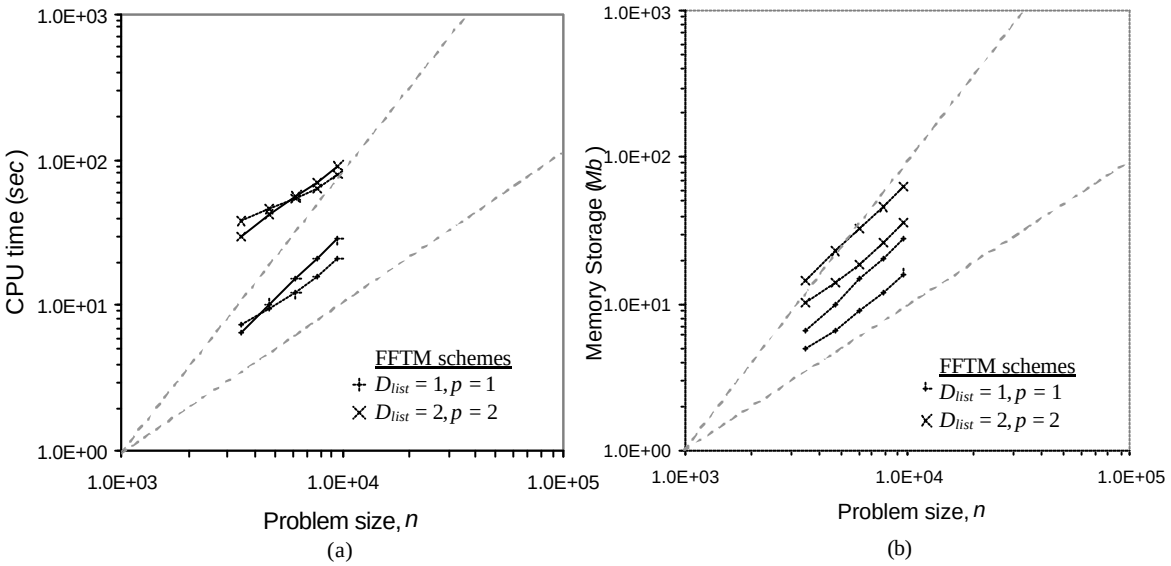
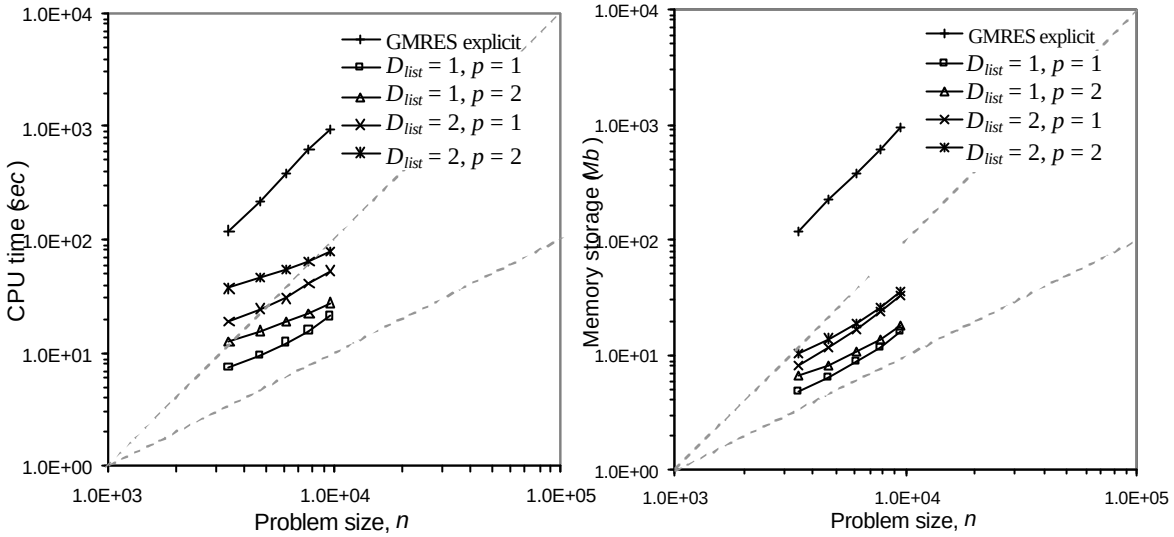


Figure 7.5. (a) CPU time and (b) memory storage requirements for FFTM schemes using different spatial discretization. Solid and dashed lines correspond to 8x8x8 and 12x12x12 cell discretizations, respectively.

Test examples

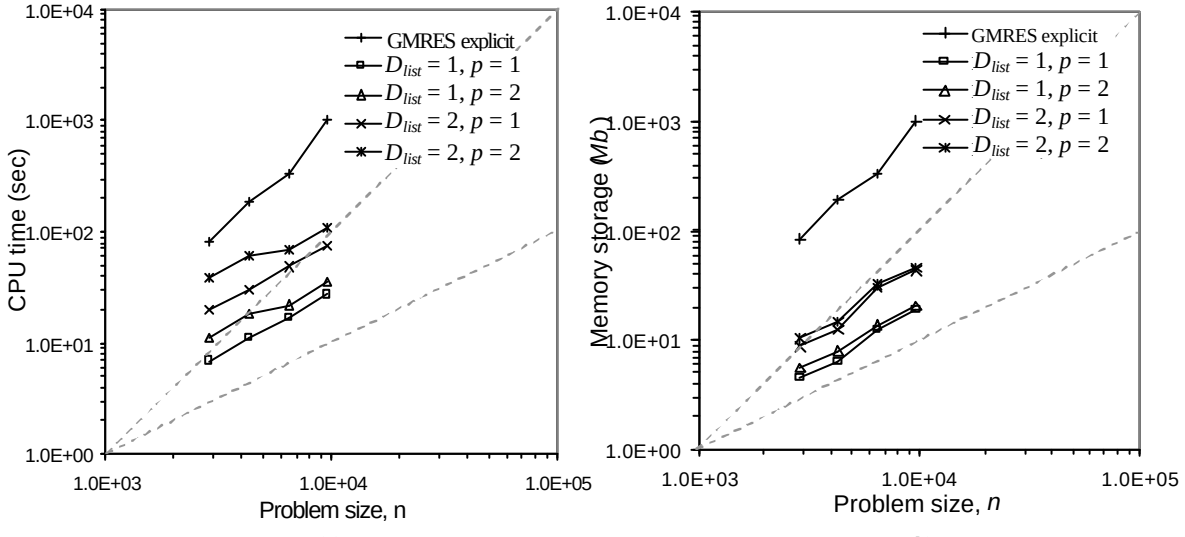
The efficiency of the various FFTM schemes is now compared against the GMRES explicit method. Two examples are considered here, namely the capacitance calculations of a cube (identical to the one used above with cell discretization fixed at 12x12x12), and the bus-crossing example (as in Figure 7.3). For the bus-crossing example, the problem size is increased by using finer mesh and/or adding two more layers of conductors. The cell discretization is either 14x14x6 or 14x14x12, depending on the

layers of the conductors. The efficiency plots of the CPU time and memory storage requirements for the two examples are given in Figures 7.6 and 7.7, respectively.



(a) (b)

Figure 7.6. Plots of (a) CPU time and (b) memory storage versus problem sizes for cube example.



(a) (b)

Figure 7.7. Plots of (a) CPU time and (b) memory storage versus problem sizes for bus-crossing example.

All the problems are solved more rapidly with FFTM. It is also observed that both sets of results show similar behaviour. As expected, the GMRES explicit method grows quadratically with increasing problem size. On the other hand, the gradients of the FFTM curves, for both time and memory requirement, are approximately near unity for all the different schemes. In general, if one desires better accuracy over the low order scheme (p = 1 and D_{list} = 1), it is usually more efficient to increase the multipole order (from p = 1 to p = 2 with D_{list} = 1) rather than to use D_{list} = 2 (that is, using D_{list} = 2 instead of D_{list} = 1, and keeping p = 1),. This is because the computational cost for storing and evaluating the "near" contributions via the ***Q2P*** matrices increases by about 5 times (from 27 cells to 125 cells), if one were to used D_{list} = 2 instead of D_{list} = 1. On the other hand, the computational cost only increases by about 2 times (from 4 to 9) if we use a higher multipole order, since they grow like order $(p+1)^2$.

From the test examples, it is demonstrated that FFTM is obviously more efficient than the GMRES explicit approach. More importantly, the method has only linear complexity growth for both the computational time and memory storage requirements. This means that FFTM can be as efficient as the existing fast methods, such as the FMM and precorrected FFT method.

Larger realistic problems

Finally, we employ FFTM to solve some larger and more realistic problems. They include the micro-mirror, 5x5 woven, bus-crossing, comb-drive and 10x10 woven, in ascending order of problem size and are depicted in Figure 7.8.

The CPU times and memory storage requirements are summarized in Table 7.8. There are two rows of results for each problem. The upper one is the CPU time and the lower one is the memory storage requirement. The computational costs for GMRES explicit approach are extrapolated from the results in the efficiency study, since these problems are too large to be simulated with our workstation. In Table 7.9, we also calculated the ratios of these costs with respect to the estimated cost of the GMRES explicit method.

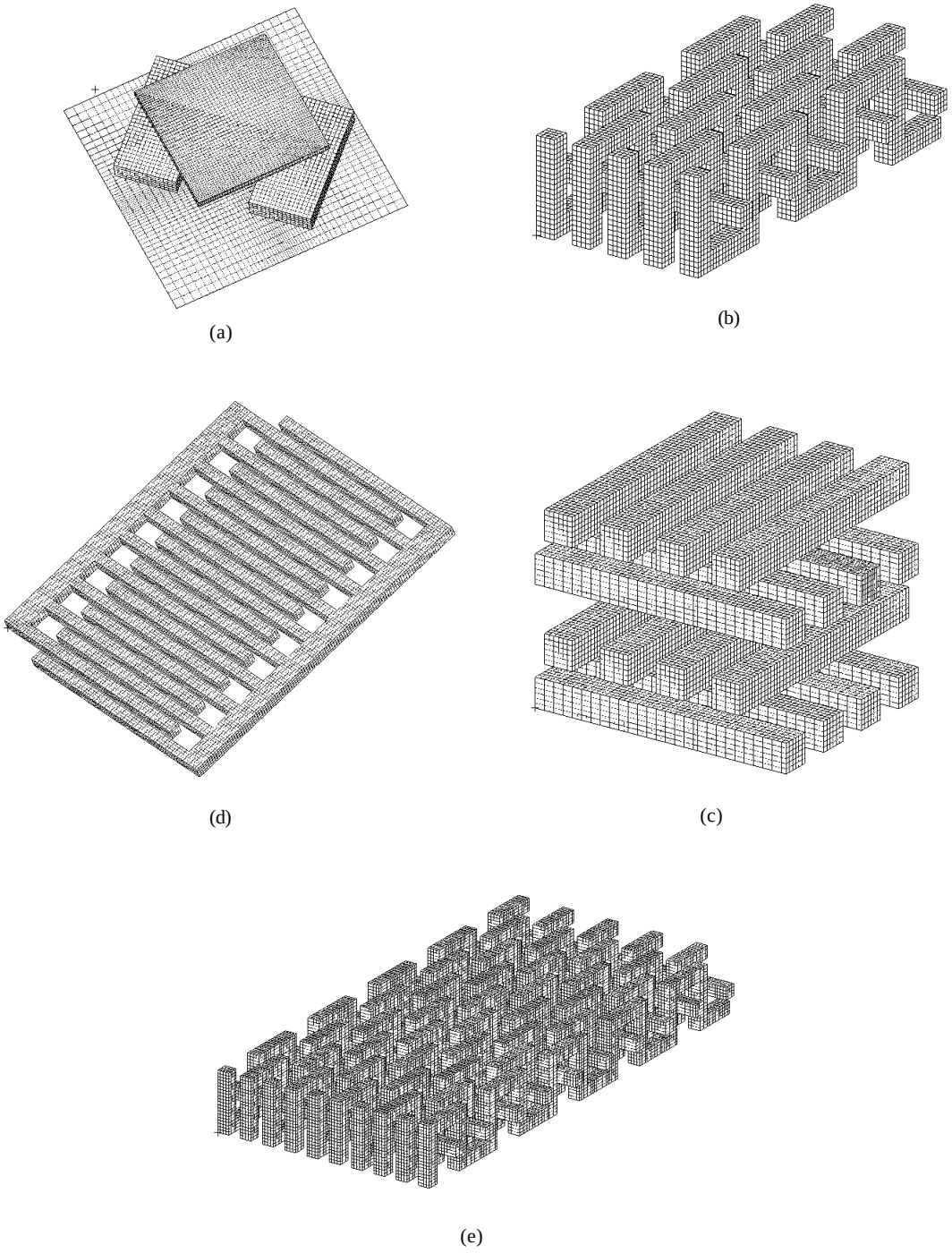


Figure 7.8. (a) micro-mirror, (b) 5x5woven, (c) bus-crossing, (d) comb-drive, and (e) 10x10woven. Cell discretizations used are (24x24x8), (16x32x8), (24x24x24), (50x60x2), and (32x64x8), respectively.

Table 7.8. CPU time and memory storage requirements for some large realistic problems.

Example (DOF)	**Solution method**				
	GMRES explicit	**FFTM, D_{list} = 1**		**FFTM, D_{list} = 2**	
		***p* = 1**	***p* = 2**	***p* = 1**	***p* = 2**
Micro-mirror (10590)	19.3 mins	0.884 mins	1.10 mins	2.39 mins	3.04 mins
	897 Mb	33.54 Mb	37.26 Mb	85.70 Mb	89.71 Mb
5x5woven (16640)	47.0 mins	0.972 mins	1.38 mins	2.71 mins	4.23 mins
	2.22 Gb	30.22 Mb	34.16 Mb	73.22 Mb	77.28 Mb
Bus-crossing (22368)	1.41 hrs	1.88 mins	2.89 mins	4.23 mins	7.79 mins
	4.00 Gb	32.89 Mb	42.33 Mb	65.23 Mb	74.78 Mb
Comb-drive (31328)	2.78 hrs	2.20 mins	3.44 mins	5.39 mins	9.77 mins
	7.85 Gb	45.59 Mb	56.81 Mb	92.50 Mb	101 Mb
10x10woven (65280)	12.05 hrs	7.02 mins	9.68 mins	16.07 mins	25.56 mins
	34.09 Gb	110 Mb	126 Mb	172 Mb	183 Mb

Table 7.9. Ratio of CPU time and memory storage with respect to GMRES explicit method.

Example (DOF)	**FFTM, D_{list} = 1**		**FFTM, D_{list} = 2**	
	***p* = 1**	***p* = 2**	***p* = 1**	***p* = 2**
Micro-mirror (10590)	0.0457	0.0569	0.123	0.157
	0.0374	0.0415	0.0955	0.100
5x5woven (16640)	0.0207	0.0294	0.0578	0.0901
	0.0136	0.0154	0.0331	0.0349
Bus-crossing (22368)	0.0222	0.0342	0.0501	0.0922
	0.00822	0.0106	0.0163	0.0187
Comb-drive (31328)	0.0132	0.0207	0.0324	0.0586
	0.00622	0.00734	0.0118	0.0129
10x10woven (65280)	0.00971	0.0134	0.0222	0.0353
	0.00323	0.00370	0.00505	0.00537

From Table 7.9, it is observed that the FFTM schemes are about one to two orders more efficient than the explicit GMRES method. It is also noted that the savings are usually more significant in terms of the memory storage requirements. For the largest problem considered here, the saving can be more than 100 times for the CPU time, and 300 times for the memory storage.

7.4 Conclusion for FFTM Method

In this part of the thesis, we developed an alternate fast algorithm that can evaluate the dense matrix-vector products rapidly. We referred it to as the Fast Fourier Transform on Multipoles (FFTM) method. The speedup in the algorithm is achieved by: (i) using the multipole expansion to approximate "distant" potential fields, and (ii) evaluating the approximate potential fields by discrete convolution via FFT algorithms.

It is demonstrated that FFTM provides relatively good accuracy, and is likely to be more accurate than FMM for the same order of multipole expansion, at least up to $p = 2$. Generally, it is more efficient to increase p rather than to use larger D_{list}, in order to obtain more accurate solution.

FFTM has approximately linear growth in terms of the computational time and memory storage requirements. Hence, it is as efficient as the existing fast methods, such as FMM and precorrected FFT methods. In fact, for a given order of accuracy, we believe that FFTM is likely to be more efficient than FMM, since the latter method would need a higher order expansions in order to achieve the desire accuracy.

8

Conclusion and Future Works

There are two main contributions in this thesis, namely: (i) improving the accuracy of the BEM analysis of electrostatic problems by using singular boundary elements, and (ii) developing a fast algorithm (FFTM) for solving the dense linear system of equations generated by BEM rapidly.

The first part of the thesis is concerned with improving the accuracy of the electrostatics analysis of corner and edge singularities in potential problems. This is achieved by developing new singular boundary elements, which correctly represent the singularity behaviour in the vicinity of the edges and corners. These singular elements have incorporated the singularity features, specifically the order of singularity, in the formulations of the shape functions. Chapter 4 is a preliminarily study on the two-dimensional singular elements analysis, and Chapter 5 extends this approach to three-dimensional problems. In both studies, it is demonstrated that the use of singular elements can produce more accurate results, both in the capacitance and electrostatics force calculations, than the standard elements (shape functions of low order polynomials). Furthermore, it is also shown that this singular element approach is more accurate than some existing methods, such as the "regularized function method" by Igarashi and Honma [25] (for two-dimensional analysis) and *h*- mesh refinement method [92] (for three-dimensional analysis).

For the three-dimensional study, the singular elements are also used in the electrostatics analysis of the electromechanical coupling simulations of some micro-devices. It is observed that using the singular elements give rise to larger deformations in comparison to the standard elements. This indicates that ignoring the corners and edges singularities (as in standard elements) in the electrostatic analysis is likely to underestimate the true deformations of the micro-structures in the simulations. However, the differences in the pull-in voltages are relatively smaller due to the pull-in phenomenon.

The second part of the thesis aims to improve the efficiency of solving the integral equation in the BEM. In Chapter 7, we proposed and implemented an alternate fast algorithm, which we referred to as the Fast Fourier Transform on Multipoles (FFTM) method. The speedup in the algorithm is achieved

by: (i) using the multipole expansion to approximate "distant" potential fields, and (ii) evaluating the approximate potential fields by discrete convolution via FFT.

It is demonstrated that FFTM provides relatively good accuracy, and is likely to be more accurate than FMM for the same order of multipole expansion (up to the second order). It is also shown that FFTM has approximately linear growth in terms of the computational time and memory storage requirements. Hence, it is as efficient as the existing fast methods, such as FMM and precorrected FFT methods. In fact, we believe that FFTM is likely to be more efficient than FMM, since FFTM needs lower order of expansion to achieve the same order of accuracy.

Several extensions of this piece of work can be identified. For the singular boundary elements, one obvious extension is to employ it in fracture mechanics, specifically for three-dimensional problems. To our best knowledge, three-dimensional fracture mechanics analysis of sharp corners is still considerably rare up to date. In this case, one would have to determine the order of singularities for different configurations of geometrical corners and material properties. Once this information is obtained, the general methodology for formulating the singular elements, as presented in Section 5.3, can be used to derive the shape functions for the singular elements.

As mentioned in Section 7.1.6, the use of local expansion, in conjunction with multipole expansion, is the most natural approach to devise the FFTM algorithm. However, the $O(p+1)^4$ growth in the number of discrete convolutions hinders the practicability of this approach. Fortunately, this scaling factor can be reduced quite significantly by applying the first technique that is used to reduce the computational cost of the potential correction step (see Section 7.1.5). In this case, besides the "near" cells, the response functions of the higher multipole moments for the "very faraway" cells are also set to zero. Physically, this means that the potential contributions from the higher multipole moments that are located "very faraway" from the potential point are simply ignored. This can be done because the higher multipole moments potential effects die down rapidly with increasing distance between the source and field points. By doing so, the number of zero paddings required to eliminate the aliasing effects can be greatly reduced. To be more explicit, instead of $2^3 = 8$ times of zero padding, it is now Z^3 times, where $1.0 < Z < 2.0$, and it depends on the order of accuracy required. Suppose $Z = 1.5$ ($Z^3 =$

3.375), then the computational cost for evaluating the discrete convolutions is reduced by approximately two times.

Another way to enhance the performance of FFTM is to employ parallel computation. An obvious part of the algorithm that we believe will gain significant speedup is the evaluation of the numerous discrete convolutions, because they are independent of one another. In other words, the task of computing the $O(p+1)^4$ discrete convolutions can be distributed to many processors simultaneously. Hence, the computational time is expected to scale like $O((p+1)^4/m)$, where m is the number of processors available.

Besides improving the FFTM algorithm, it can also be extended to other areas, such as in particle simulations and solving Helmholtz problems. Although, both the problems have already being solved efficiently using FMM, we believe that FFTM can perform better because of its superior accuracy over FMM. In fact, any problems that have being solved by FMM, can also be solve as efficiently by FFTM. To a greater extent, we believe that this new fast algorithm can be applied to any problems that are solved by BEM, since the problems are ultimately reduced to solving dense linear systems of equations.

Bibliography

[1] S.D. Senturia. CAD for microelectromechanical system. 8th International Conference on Solid-State Sensors and Actuators (TRANSDUCERS '95), pp. 5-8.

[2] S.D. Senturia, N. Aluru and J. White. Simulating the Behavior of MEMS Devices: Computational Methods and Needs. *IEEE Computational Science and Engineering*, vol. 4, no. 1, pp. 30-43, 1997.

[3] S.D. Senturia. CAD Challenges for Microsensors, Microactuators, and Microsystems. *Proceeding of the IEEE*, vol. 86. no. 8, pp. 1611-1626, Aug 1998.

[4] S.D. Senturia, R.M. Harris, B.P. Johnson, S. Kim, K. Nabors, M.A. Shulman and J. White. A computer-aided design system for microelectromechanical system (MEMCAD). *Journal of Microelectromechanical Systems*, vol. 1, no. 1, pp. 3-13, 1992.

[5] J.R. Gilbert, P.M. Osterberg, R.M. Harris, D.O. Ouma, X. Cai, A. Pfajfer, J. White, and S.D. Senturia. Implementation of MEMCAD system for electrostatic and mechanical analysis for complex structures from mask descriptions. *IEEE Proceedings on Micro Electro Mechanical Systems 1993, (MEMS '93)*, pp. 207-212.

[6] B.P. Johnson, S. Kim, S.D. Senturia, J. White. MEMCAD capacitance calculations for mechanically deformed square diaphragm and beam microstructures. *6th International Conference on Solid-State Sensors and Actuators (TRANDUCERS '91)*, pp. 494 –497.

[7] F. Maseeh. IntelliCAD : The CAD for MEMS. *Proceedings of Western Electronic Show and Convection 1995 (WESCON '95)*, pp. 320-324.

[8] J.M. Funk, J.G. Korvink, J. Buhler, M. Bachtold, and H. Baltes. SOLIDIS: A Tool for Microactuator Simulation in 3D. *Journal of Microelectromechanical Systems*, vol. 6, no. 1, pp. 70-82, 1997.

[9] X. Cai, H. Yie, P. Osterberg, J. Gilbert, S.D. Senturia and J. White. A relaxation/multipole-accelerated scheme for self-consistent electromechanical analysis of complex 3-D microelectromechanical structures. *International Conference on Computer-Aided Design, 1993 (ICCAD-93)*, pp. 283 –286.

[10] H. Yie, X. Cai and J. White. Convergence properties of relaxation versus the surface-Newton generalized-conjugate residual algorithm for self-consistent electromechanical analysis of 3-D micro-electro-mechanical structures. *International Workshop on Numerical Modeling of Processes and Devices for Integrated Circuits 1994 (NUPAD V)*, pp. 137 –140.

[11] M. Bachtold, J.G. Korvink, J. Funk and H. Baltes. New convergence scheme for self-consistent electromechanical analysis of MEMS. *International Electron Devices Meeting, 1995*, pp. 605 – 608.

[12] J.R. Gilbert, R. Legtenberg and S.D. Senturia. 3D coupled electromechanics for MEMS: applications of Cosolve-EM. *IEEE Proceedings of Micro Electro Mechanical Systems 1995 (MEMS '95)*, pp. 122-127.

[13] N.R. Aluru, J. White. An efficient numerical technique for electromechanical simulation of complicated microelectromechanical structures. *Sensors and Actuators A*, vol. 58, pp. 1-11, 1997.

[14] N.R. Aluru and J. White. Algorithms for Coupled Domain MEMS Simulation. *Proceedings of the 34th on Design Automation Conference 1997 (DAC '97)*, pp. 686 –690.

[15] N.R. Aluru and J. White. A Multilevel Newton Method for Mixed-Energy Domain Simulation of MEMS. *Journal of Microelectromechanical Systems*, vol. 8, no. 3, pp. 299-307, 1999.

[16] G.E. Blandford, A.R. Ingraffea and J.A. Liggett. Two-dimensional stress intensity factor computations using the boundary element method. *International Journal for Numerical Methods in Engineering*, vol. 17, pp. 387-404, 1981.

[17] J. Martinez and J. Dominguez. On the use of quarter-point boundary elements for stress intensity factor computations. *International Journal for Numerical Methods in Engineering*, vol. 17, pp. 1941-1950, 1984.

[18] M.P. Aiza, A. Saez and J. Dominguez. A singular element for three-dimensional fracture mechanics analysis. *Engineering Analysis with Boundary Elements*, vol. 20, pp. 275-285, 1997.

[19] M. Rezayat, F.J Rizzo and D.J. Shippy. A unified boundary integral equation method for a class of second order elliptic boundary value problems. *Journal of Australia Mathematics Society Series B*, vol. 25, pp. 501-517, 1984.

[20] Z.H. Jia and D.J. Shippy. On the computation of two-dimensional stress intensity using the boundary element method. *International Journal for Numerical Methods in Engineering*, vol. 26, pp. 2739-2753, 1988.

[21] Z.H. Jia and D.J. Shippy. Three-dimensional crack analysis using singular boundary elements. *International Journal for Numerical Methods in Engineering*, vol. 28, pp. 2257-2273, 1989.

[22] El Abdi R and G. Valentin. Isoparametric elements for a crack normal to the interface between two bonded layers. *Computers & Structures*, vol. 33, 241-248, 1989.

[23] M.L. Luchi and S. Rizzuti. Boundary elements for three-dimensional elastic crack analysis. *International Journal for Numerical Methods in Engineering*, vol. 24, pp. 2253-2271, 1987.

[24] H. Igarashi, Y. Mizuyama and T. Homna. A boundary element analysis of transmission-line parameters using singular elements. *IEEE Transactions on Magnetics*, vol. 32, no. 3, pp. 686-689, 1996.

[25] H. Igarashi and T. Honma. A boundary element method for potential fields with corner singularities. *Applied Mathematical Modelling*, vol. 20, pp. 847-852, 1996.

[26] G.C. Georgiou, L.G. Olson and Y.S. Smyrlis. A singular function boundary integral method for Laplace equation. *Communications in Numerical Methods in Engineering*, vol. 12, pp. 127-134, 1996.

[27] L.G. Olson, G.C. Georgiou and W.W. Schultz. An efficient finite element method for treating singularities in Laplace's equation. *Journal of Computational Physics*, vol. 96, pp. 391-410, 1991.

[28] N.M. Wigley. An efficient method for subtracting off singularities at corners for Laplace's equation. *Journal of Computational Physics*, vol. 78, pp. 369-377, 1988.

[29] Z.C. Li, R. Mathon and P. Sermer. Boundary elements for solving elliptic problems with singularities and interfaces. *SIAM Journal of Numerical Analysis*, vol. 24, no. 3, pp. 487-498, 1987.

[30] Z.C. Li and T.D. Bui. Generalized hybrid-combined methods for the singularity problems of homogenous elliptic equations. *International Journal for Numerical Methods in Engineering*, vol. 26, 785-803, 1988.

[31] Z.C. Li and T.D. Bui. The simplified hybrid-combined methods for Laplace's equation with singularities. *Journal of Computational and Applied Mathematics*, vol. 29, pp. 171-193, 1990.

[32] Z.C. Li and T.D. Bui. Penalty-combined method and applications in solving elliptic problems with singularities. *Computer Methods in Applied Mechanics and Engineering*, vol. 97, pp. 291-316, 1992.

[33] Z.C. Li and T.D. Bui. Coupling techniques in boundary-combined methods for solving elliptic problems with singularities. *Engineering Analysis with Boundary Elements*, vol. 12, pp. 305-320, 1993.

[34] G. Fichera. Asymptotic behavior of the electric field and density of the electric charge in the neighourhood of singular points of a conducting surface. *Russian Mathematical Surveys*, vol. 30, no. 3, pp. 107-127, 1975.

[35] A. E. Beagles and J. R. Whiteman. General conical singularities in three-dimensional Poisson problems. *Mathematical Methods in the Applied Science*, vol. 11, pp. 215-235, 1989.

[36] Z. P. Bazant. Three-dimensional harmonic functions near termination or intersection of gradient singularity lines: A general numerical method. *International Journal of Engineering Science*, vol. 12, pp. 221-243, 1974.

[37] Y. Saad and M.H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific Statistical and Computing*, vol. 7, no. 3, pp. 856-869, 1986.

[38] H. Cheng, L. Greengard and V. Rokhlin. A fast adaptive multipole algorithm in three dimensions. *Journal of Computational Physics*, vol. 155, pp. 468-498, 1999.

[39] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, vol. 73, no. 2, pp. 325-348, 1987.

[40] L. Greengard and V. Rokhlin. A new version of the Fast Multipole Method for the Laplace equation in three dimensions. *Acta Numerica*, vol. 6, pp. 229-269, 1997.

[41] K. Nabors and J. White. Fastcap: A multipole accelerated 3-D capacitance extraction program. *IEEE Transaction on Computer-Aided Design Integrated Circuits and Systems*, vol. 11, no. 10, pp. 1447-1459, 1991.

[42] K. Nabors, S. Kim, and J. White. Fast capacitance extraction of general three-dimensional structures. *IEEE Transaction on Microwave Theory and Techniques*, vol. 40, no. 7, pp. 1496-1506, 1992.

[43] K. Nabors, F.T. Korsmeyer, F.T. Leighton and J. White. Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional first-kind integral equations of potential theory *SIAM Journal on Scientific Statistical and Computing*, vol. 15, no. 3, pp. 713-735, 1994.

[44] L. Greengard and V. Rokhlin. *Rapid evaluation of potential fields in three dimensions*, in Vortex Methods, C.R. Anderson and C. Greengard, eds., vol. 1360 of Lecture Notes in Mathematics, Springer-Verlag, New York, 1988.

[45] K.S. Nabors. *Efficient Three-dimensional Capacitance Calculation*. Ph. D. dissertation, MIT, Boston, 1993.

[46] J.R. Phillips. Error and complexity analysis for a collocation-grid-projection plus precorrected-FFT algorithm for solving potential integral equations with Laplace or Helmholtz kernels. *Proceedings of the 1995 Copper Mountain Conference on Multi-grid Methods*, April, 1995.

[47] J.R. Phillips and J. White. A precorrected-FFT method for electrostatic analysis of complicated 3-D structures. *IEEE Transaction Computer-Aided Design Integrated Circuits and Systems*, vol. 16, no. 10, pp. 1059-1072, 1997.

[48] J.R. Phillips. *Rapid solution of potential integral equations in complicated 3-dimensional geometries*. Ph. D. dissertation, MIT, Boston, 1997.

[49] E.O. Brigham. *The Fast Fourier Transform and its Applications*. Prentice-Hall, Englewood Cliffs, 1988.

[50] C.A. Brebbia and J. Dominguez. *Boundary Elements: An Introductory Course*. Southhampton, Boston : Computational Mechanics Publications; New York : Co-published with McGraw-Hill, 1992.

[51] Marc Bonnet. *Boundary Integral Equation Methods for Solids and Fluids*. Chichester, West Sussex : John Wiley & Sons, 1995.

[52] J. D. Jackson. *Classical Electrodynamics, 2nd edition*, John Wiley, New York, 1975.

[53] C.T. Kelley. *Iterative methods for linear and nonlinear equations*. SIAM, Philadelphia, 1995.

[54] Y. Saad. *Iterative methods for sparse linear systems*. PWS Pub. Co., Boston, 1996.

[55] E. Kita and N. Kamiya. Recent studies on adaptive boundary element methods. *Advances in Engineering Software*, vol. 19, pp. 21-32, 1994.

[56] S. Liapis. A review of error estimation and adaptivity in the boundary element method. *Engineering Analysis with Boundary Elements*, vol. 14, pp. 315-323, 1994.

[57] N. Kamiya, E. Kita and M. Koide. Error estimation and adaptive mesh refinement in boundary element methods. *Boundary Element Technology VIII (BETCH '93)*, Computational Mechanics Publications, Southampton, and Elsevier Applied Science, London, pp. 343-352, 1993.

[58] E. Rank. Adaptive *h*-, *p*- and *hp*- versions for boundary integral element methods. *International Journal for Numerical Methods in Engineering*, vol. 28, pp. 1335-1349, 1989.

[59] P. Parreira. Further developments on error indicators and estimators for adaptive hierarchical boundary elements. *11th International Conference on BEM (Advances in Boundary Elements)*, ed. C.A. Brebbia and J.J. Connor, Computational Mechanics Publication, Southampton, and Springer Verlag, Berlin, vol. 1: Computational and Fundamentals, pp. 105-122, 1989.

[60] W. Sun and N.G. Zamani. Adaptive mesh redistribution for the boundary element in elastostatics. *Computers and Structures*, vol. 36, no. 6. pp. 1081-1088, 1990.

[61] W. Sun and N.G. Zamani. Adaptive mesh refinement/redistribution for the equations of linear elasticity, boundary element formulation. *Computers and Structures*, vol. 44, no. 3. pp. 627-637, 1992.

[62] W. Sun and N.G. Zamani. An adaptive *h-r* boundary element algorithm for the Laplace equation. *International Journal for Numerical Methods in Engineering*, vol. 33, pp. 537-552, 1992.

[63] Y.F. Dong and P. Parreira. Quadratic *h*-hierarchical adaptive boundary elements. *Boundary Element Technology VII (BETCH '92)*, ed. C.A. Brebbia and M.S. Ingber. Computational Mechanics Publications, Southampton, and Elsevier Applied Science, London, pp. 835-850, 1992.

[64] Y.F. Dong and P. Parreira. *H*- adaptive BEM based on linear hierarchical functions. *14th International Conference on BEM (Boundary Elements XIV)*, ed. C.A. Brebbia, J. Dominquez and F. Paris, Computational Mechanics Publication, Southampton, Elsevier Applied Science, London, pp. 653-665, 1992.

[65] A. Charafi, A.C. Neves and L.C. Wrobel. Use of local reanalysis and quadratic *h*-hierarchical functions in adaptive boundary element models. *Boundary Element Technology VIII (BETCH '93)*, ed. H. Pina and C.A. Brebbia. Computational Mechanics Publications, Southampton, pp. 353-368, 1993.

[66] A. Charafi, A.C. Neves and L.C. Wrobel. *h*- Hierarchical adaptive boundary element method using local reanalysis. *International Journal for Numerical Methods in Engineering*, vol. 38, pp. 2185-2207, 1995.

[67] A. Charafi and L.C. Wrobel. *h*- Hierarchical functions for 2D and 3D BEM. *Engineering Analysis with Boundary Elements*, vol. 16, pp. 341-349, 1995.

[68] K. Kawaguchi and N. Kamiya. An adaptive BEM by sample point error analysis. *Engineering Analysis with Boundary Elements*, vol. 9, pp. 255-262, 1992.

[69] Z. Zhao and X. Wang. Error estimation and *h* adaptive boundary elements. *Engineering Analysis with Boundary Elements*, vol. 23, pp. 793-803, 1999.

[70] E. Alarcon and A. Reverter. *p*- adaptive boundary elements. *International Journal for Numerical Methods in Engineering*, vol. 23, pp. 801-829, 1987.

[71] M. Cerrolaza and E. Alarcon. *p*- adaptive boundary elements for three-dimensional potential problems. *Communications in Applied Numerical Methods*, vol. 3, pp. 335-344, 1987.

[72] M. Cerrolaza and E. Alarcon. Further applications of *p*- adaptive boundary elements. *10th International Conference on BEM (Boundary Element X)*, ed. C.A. Brebbia, Computational Mechanics Publication, Southampton, and Springer Verlag, Berlin, vol. 1: Mathematical and Computational Aspects, pp. 123-135, 1988.

[73] E. Kita, K. Higuchi and N. Kamiya. Adaptive mesh redistribution of boundary element method. *19th International Conference on BEM (Boundary Elements XIX)*, ed. M. Marchetti, C.A. Brebbia and M.H. Aliabadi, Computational Mechanics Publication, Southampton, pp. 739-748, 1997.

[74] G.F. Carey and S. Kennon. Adaptive mesh redistribution for a boundary element (panel) method. *International Journal for Numerical Methods in Engineering*, vol. 24, pp. 2315-2325, 1987.

[75] M.S. Ingber and A.K. Mitra. Grid optimization for the boundary element method. *International Journal for Numerical Methods in Engineering*, vol. 23, pp. 2121-2136, 1986.

[76] M. Bactold, M. Emmenegger, J.G. Korvink and H. Baltes. An error indicator and automatic adaptive meshing for electrostatic boundary element simulations. *IEEE transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 12, 1997.

[77] E. Kita, K. Higuchi and N. Kamiya. Application of ꟴ and hr- adaptive BEM to two-dimensional elastic problem. *Engineering Analysis with Boundary Elements*, vol. 24, pp. 317-324, 2000.

[78] AD. Nurse. New superelements for singular derivative problems of arbitrary order. *International Journal for Numerical Methods in Engineering* , vol. 50, pp. 135-146, 2001.

[79] J. Qian and N. Hasebe. On the technique of shifting side-nodes in isoparametric elements to impose arbitrary singularity. *Computers & Structures*, vol. 66, pp. 841-846, 1998.

[80] A.H. Stroud and D. Secrest. *Gaussian Quadrature Formulas*. Englewood Cliffs, NJ: Prentice-Hall, New York.

[81] A.H. Stroud. *Approximate Calculation of Multiple Integrals*. Englewood Cliffs, NJ: Prentice-Hall, New York.

[82] H.L.G. Pina, J.L.M. Fernandes and C.A. Brebbia. Some numerical integration formulae over triangles and squares with 1/R singularities. *Applied Mathematical Modelling*, vol. 5, pp. 209-211, 1981.

[83] K. Hayami and C.A. Brebbia. Quadrature methods for singular and nearly singular integrals in 3-D boundary element method. *Boundary Elements X*, vol. 1, pp. 237-264, 1988.

[84] M. Critescu and G. Loubignac. Gaussian quadrature formula for functions with singularities in 1/R over triangles and quadrangles. *Recent Advances in Boundary Element Method*, pp. 375-390, Pentech Press, New York.

[85] J.C. Lachat and J.O. Watson. Effective numerical treatment of boundary integral equations: a formulation fro three-dimensional elastostaics. *International Journal for Numerical Methods in Engineering*, vol. 10, pp. 991-1005, 1976.

[86] M.H. Aliabadi and W.S. Hall. The regularizing transformation integration method for boundary element kernels. Comparison with series expansion and weighted Gaussian integration methods. *Engineering Analysis with Boundary Elements*, vol. 6, no. 2, pp. 66-71, 1989.

[87] K. Hayami and C.A. Brebbia. A new coodinate transformation method for singular and nearly singular integrals over general curved boundary elements. Boundary Elements IX, vol. 1, pp. 375-399, Proceedings of the Ninth Int. Conf., Stuttgart.

[88] J.C.F. Telles. A self-adaptive coordinate transformation for efficient numerical evaluations of general boundary element integrals. *International Journal for Numerical Methods in Engineering*, vol. 24, pp. 959-973, 1987.

[89] J.C.F. Telles and R.F. Oliveira. Third degree polynomial transformation for boundary element integrals: Further improvements. *Engineering Analysis with Boundary Elements*, vol. 13, pp. 135-141, 1994.

[90] V. Sladek, J. Sladek and M. Tanaka. Evaluation of 1/r integrals in BEM formulations for 3-D problems using coordinate multi-transformations. *Engineering Analysis with Boundary Elements*, vol. 20, pp. 229-244, 1997.

[91] V. Sladek, J. Sladek and M. Tanaka. Optimal transformations of the integration variables in computation of singular integrals in BEM. *International Journal for Numerical Methods in Engineering*, vol. 47, pp. 1263-1283, 2000.

[92] J. Tausch and J. White. Mesh refinement strategies for capacitance extraction based on residual errors. *IEEE 5th Topical Meeting on Electrical Performance of Electronic Packaging 1996*, pp. 236-237.

[93] W.C. Tang, M.G. Lim and R.T. Howe. Electrostatic comb drive levitation and control method. *Journal of Microelectromechanical Systems*, vol. 1, pp. 170-178, 1992.

[94] C.R. Anderson. An implementation of the fast multipole method without multipoles. *SIAM Journal on Scientific Statistical and Computing*, vol. 14, no. 4, pp. 923-947, 1992.

[95] J. Makino. Yet another fast multipole without multipoles-pseudoparticle multipole method. *Journal of Computational Physics*, vol. 151, pp. 910-920, 1999.

[96] W.D. Elliott and J.A. Board, Jr. Fast Fourier transform accelerated fast multipole algorithm. *SIAM Journal on Scientific Computing*, vol. 17, no. 2, pp. 398-415, 1996.

[97] A.W. Appel. An efficient program for many-body simulations. *SIAM Journal on Scientific Statistical and Computing*, vol. 6, no. 6, pp. 85-103, 1985.

[98] J. Barnes and P. Hut. A hierarchical O(*N* log *N*) force calculation algorithm. *Nature*, vol. 324, pp. 446-449, 1986.

[99] C.L. Berman. Grid-multipole calculations. *SIAM Journal on Scientific Computing*, vol. 16, no. 5, pp. 1082-1091, 1995.

[100] G. Beylkin, R. Coifman and V. Rokhlin. Fast wavelet transform and numerical algorithms I. *Communications on Pure ans Applied Mathematics*, vol. 44. , pp. 141-183, 1991.

[101] M. Spasojevic, R. Schenider and P.L. Levin. On the creation of sparse boundary element mattrices for two-dimensional electrostatics problems using the orthogonal Haar wavelet. *IEEE Transactions on Dielectric and Electric Insulation*, vol. 4, pp. 249-258, 1997.

[102] P.L. Levin, M. Spasojevic and R. Schenider. Creation of sparse boundary element matrices for 2-D and axi-symmetric electrostatics problems using the bi-orthogonal Haar wavelet. *IEEE Transactions on Dielectric and Electric Insulation*, vol. 5, pp. 469-484, 1998.

[103] T. vonPetersdorff and C. Schwab. Wavelet approximations for first kind boundary integral equations on polygons. *NUMERISCHE MATHEMATIK*, vol. 74, pp. 479-516, 1996.

[104] T. vonPetersdorff, C. Schwab and R. Schneider. Multiwavelets for second-kind integral equations. *SIAM Journal on Numerical Analysis*, vol. 34, pp. 2212-2227, 1997.

[105] C. Lage, C. Schwab. Wavelet Galerkin algorithms for boundary integral equations. *SIAM Journal on Scientific Computing*, vol. 20, pp. 2195-2222, 1999.

[106] S. Kapur and J. Zhao. A fast method of moments solver for efficient parameter extraction of MCM's. *Proc. Design Automation Conference, CA, June, 1997*, pp. 141-146.

[107] S. Kapur, and D. E. Long. IES3: Efficient electrostatic and electromagnetic simulation. *IEEE Computational Science and Engineering*., vol. 5, pp. 60-67, 1998.

Appendix A

Generalized Minimum RESidual (GMRES)

Solving linear system of equations is one fundamental task in many numerical methods, for example solving boundary value problems (BVP) using BEM, where the problems are reduced to a dense linear system of equations, as given by

$$\boldsymbol{A}\vec{x} = \vec{b} \tag{A.1}$$

where $\boldsymbol{A}$ is a fully populated coefficient matrix.

Solving (A.1) using direct methods, such as Gaussian Elimination, require $O(n^3)$ operations, which becomes computationally intractable if the problem size n exceeds several hundreds. On the other hand, using projection iterative methods can reduce the operation counts to $O(n^2)$. This is because the main computational cost of these iterative solvers is due to the generation the orthonormal basis vectors, which are defined by dense matrix-vector products. In general, they only aim to solve (A.1) approximately by minimizing the residual norm. Generalized Minimum RESidual (GMRES) is one such iterative method that is especially effective in solving dense linear systems generated by BEM. The remaining of this appendix will describe the method in more details.

A.1 Basic Concepts of Projection Iterative Methods

Suppose $\boldsymbol{K}_m$ is the chosen subspace and m is its dimension, then m constraints are imposed in order to extract an approximation $\hat{x}_m$. A typical approach is to impose m orthogonality conditions. If the residual vector, $\vec{r} = \vec{b} - \boldsymbol{A}\vec{x}$ is constrained to be orthogonal to m linearly independent vectors, then another subspace $\boldsymbol{L}_m$ of dimension m is generated, which is called the subspace of constraints, and these orthogonality conditions are known as the Petrov-Galerkin conditions. To summarize, a projection iterative method seeks an approximate solution $\hat{x}_m$ from an affine subspace $\vec{x}_0 + \boldsymbol{K}_m$ of dimension m by imposing the Petrov-Galerkin conditions, that is,

$$\vec{b} - \boldsymbol{A}\hat{x}_m \perp \boldsymbol{L}_m \tag{A.2}$$

where $\vec{x}_0$ is an arbitrary initial guess of the solution.

A.2 Krylov Subspace Methods

One important class of $\boldsymbol{K}_m$ is the Krylov subspace, which is defined by

$$\boldsymbol{K}_m(\boldsymbol{A}, \vec{r}_0) = span\{\vec{r}_0, \boldsymbol{A}\vec{r}_0, \boldsymbol{A}^2\vec{r}_0, ..., \boldsymbol{A}^{m-1}\vec{r}_0\} \tag{A.3}$$

where $\vec{r}_0 = \vec{b} - \boldsymbol{A}\vec{x}_0$ is the initial residual.

Different Krylov subspace methods arise from the different choices of the subspace $\boldsymbol{L}_m$. One possible choice is $\boldsymbol{L}_m = \boldsymbol{K}_m$ and also its variant $\boldsymbol{L}_m = \boldsymbol{A}\boldsymbol{K}_m$, in which GMRES belongs. The approximated solution $\hat{x}_m$ is generally be expressed as

$$\hat{x}_m = \vec{x}_0 + \sum_{i=0}^{m-1} g_i \boldsymbol{A}^i \vec{r}_0 = \vec{x}_0 + p_{m-1}(\boldsymbol{A})\vec{r}_0 \tag{A.4}$$

where $p_{m-1}(\boldsymbol{A})$ corresponds to a polynomial of degree *m*-1.

A.3 GMRES: Basic Concepts and Theorems

GMRES was proposed by Saad and Schultz [37] as a Krylov subspace method for solving non-symmetric systems, where the constraint subspace $\boldsymbol{L}_m$ is chosen to be $\boldsymbol{A}\boldsymbol{K}_m$. The *mth* iteration of GMRES is the solution to the least square problem

$$\text{minimize } \left\|\vec{b} - \boldsymbol{A}\hat{x}_m\right\|_2, \quad \hat{x}_m \in \vec{x}_0 + \boldsymbol{K}_m \tag{A.5}$$

where $\vec{r}_m = \vec{b} - \boldsymbol{A}\hat{x}_m$ is the GMRES residual at the *mth* iterations. Substituting (A.4) into the residual equation gives,

$$\begin{aligned} \vec{b} - \boldsymbol{A}\hat{x}_m &= \vec{b} - \boldsymbol{A}(\vec{x}_0 + p_{m-1}(\boldsymbol{A})\vec{r}_0) \\ &= (1 - p_{m-1}(\boldsymbol{A}))\vec{r}_0 = \overline{p}_m(\boldsymbol{A})\vec{r}_0 \end{aligned} \tag{A.6}$$

where $\overline{p}_m \in \boldsymbol{P}_m$ is a residual polynomial of degree *m* that satisfy the condition $\overline{p}_m(\boldsymbol{0}) = 1$. Using (A.5) and (A.6), the following theorem is derived.

THEOREM A.1. Let $\boldsymbol{A}$ be nonsingular and $\hat{x}_m$ be the approximated solution at the *mth* GMRES iterations. Then for all $\overline{p}_m \in \boldsymbol{P}_m$

$$\left\|\vec{r}_m\right\|_2 = \min_{p \in \boldsymbol{P}_m} \left\|\overline{p}(\boldsymbol{A})\vec{r}_0\right\|_2 \le \left\|\overline{p}_m(\boldsymbol{A})\vec{r}_0\right\|_2 \tag{A.7}$$

which leads to the following corollary,

$$\frac{\|\vec{r}_m\|_2}{\|\vec{r}_0\|_2} \le \|\vec{p}_m(\boldsymbol{A})\|_2 \tag{A.8}$$

From (A.8), it is easy to see that GMRES algorithm will find the exact solution in at most *n* iterations (assuming infinite precision for the arithmetic operations). It is also interesting to note two other theorems, which show finite termination of the GMRES algorithm under certain circumstances.

THEOREM A.2. Let $\boldsymbol{A}$ be a nonsingular and diagonalizable matrix. Suppose $\boldsymbol{A}$ has only *k* distinct eigenvalues, then GMRES will terminate in at most *k* iterations.

THEOREM A.3. Let $\boldsymbol{A}$ be a nonsingular normal matrix. Let $\vec{b}$ be linearly spanned by the *k* eigenvectors of $\boldsymbol{A}$, that is, $\vec{b} = \sum_{i=1}^{k} g_i \vec{v}_i$, where $\vec{v}_i$ is the *ith* eigenvector of $\boldsymbol{A}$ and g_i is the corresponding coefficient. Then GMRES will also terminate in at most *k* iterations.

A.4 GMRES : Implementation and Algorithms

Suppose $\boldsymbol{V}_m$ is an orthogonal projector onto $\boldsymbol{K}_m$. Then (A.4) can be written as

$$\hat{x}_m = \vec{x}_0 + \boldsymbol{V}_m \vec{y}_m \tag{A.9}$$

where $\vec{y}_m \in R^m$ is the coefficient vector to be determined. The least squares problem in (A.5) becomes

$$\text{minimize } \|\vec{r}_0 - \boldsymbol{A}\boldsymbol{V}_m \vec{y}_m\|_2, \quad \vec{y}_m \in R^m \tag{A.10}$$

Suppose Gram-Schmidt or modified Gram-Schmidt is used to form the orthonormal basis of $\boldsymbol{K}_m$, which in this case is called the Arnoldi process (ALGORITHM A.1), then

$$\boldsymbol{A}\boldsymbol{V}_m = \boldsymbol{V}_{m+1}\boldsymbol{H}_m \tag{A.11}$$

where $\boldsymbol{H}_m$ is an upper Hessenberg matrix whose entries h_{ij} satisfy the condition, $h_{ij} = 0$ if $i > j-1$. Substituting (A.11) into the residual vector gives

$$\vec{r}_m = \vec{r}_0 - \boldsymbol{V}_{m+1}\boldsymbol{H}_m \vec{y}_m = \boldsymbol{V}_{m+1}(b\vec{e}_1 - \boldsymbol{H}_m \vec{y}_m) \tag{A.12}$$

where $b = \|\vec{r}_0\|_2$, $\vec{e}_1 = (1, 0, \ldots, 0)^T \in R^m$, and the residual norm is

$$\|\vec{r}_m\|_2 = \|(b\vec{e}_1 - \boldsymbol{H}_m \vec{y}_m)\|_2 \tag{A.13}$$

since the column vectors of $\boldsymbol{V}_{m+1}$ are orthonormal.

ALGORITHM A.1: Arnoldi process.

1. Let $\vec{r}_0 = \vec{b} - \boldsymbol{A}\vec{x}_0$ and $\vec{v}_1 = \dfrac{\vec{r}_0}{\left\|\vec{r}_0\right\|_2}$.
2. For i = 1,…, m-1,

$$\vec{v}_{i+1} = A\vec{v}_i - \sum_{j=1}^{i} \left(\left(A\vec{v}_i\right)^T \vec{v}_j \right) \vec{v}_j$$

$$\vec{v}_{i+1} = \frac{\vec{v}_{i+1}}{\left\|\vec{v}_{i+1}\right\|_2}$$

In summary, the GMRES approximation is a unique solution of (A.9), which minimized (A.10), that is, $\vec{y}_m = \text{minimize } \left\|\left(b\vec{e}_1 - \boldsymbol{H}_m \vec{y}_m\right)\right\|_2 \in R^m$.

To solve the least squares problem defined in (A.13) efficiently, the upper Hessenberg matrix is transformed into upper triangular form by using the Givens rotations. A Givens rotation is defined as

$$\boldsymbol{G}_i = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 & 0 \\ 0 & \ddots & \ddots & & & 0 \\ \vdots & \ddots & c_i & s_i & & \vdots \\ \vdots & & -s_i & c_i & \ddots & \vdots \\ 0 & & & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & \cdots & 0 & 1 \end{bmatrix} \tag{A.14}$$

which is a (m+1) square matrix, where m is the number of iterations performed. The entries c_i and s_i are located in the i and i+1 rows and columns, which are defined as

$$c_i = \frac{h_{ii}^{(i-1)}}{\sqrt{\left(h_{ii}^{(i-1)}\right)^2 + h_{i+1,i}^2}}, \quad s_i = \frac{h_{i+1,i}}{\sqrt{\left(h_{ii}^{(i-1)}\right)^2 + h_{i+1,i}^2}} \tag{A.15}$$

Applying (A.16) repeatedly will ultimately reduce $\boldsymbol{H}_m$ to its triangular form

$$\boldsymbol{Q}_m \boldsymbol{H}_m = \boldsymbol{R}_m \tag{A.16}$$

where $\boldsymbol{Q}_m = \boldsymbol{G}_m \ldots \boldsymbol{G}_2 \boldsymbol{G}_1$, and $\boldsymbol{R}_m$ is a (m+1)xm upper triangular matrix. Finally, substituting (A.16) into (A.13) gives

$$\begin{aligned} \left\|\left(b\vec{e}_1 - \boldsymbol{H}_m \vec{y}_m\right)\right\|_2 &= \left\|\boldsymbol{Q}_m\left(b\vec{e}_1 - \boldsymbol{H}_m \vec{y}_m\right)\right\|_2 \\ &= \left\|\left(\vec{g}_m - \boldsymbol{R}_m \vec{y}_m\right)\right\|_2, \quad \text{since } \boldsymbol{Q}_m \text{ is unitary.} \end{aligned} \tag{A.17}$$

The solution of (A.17) is obtained by solving the triangular system, with the last row of $\boldsymbol{R}_m$ and last term of $\vec{g}_m$ removed. Note that the last entry of $\vec{g}_m$ corresponds to the residual of the least square problem that is used as the convergence indicator.

The Arnoldi algorithm and Givens rotations are the two important frameworks that GMRES algorithm builds on. With these in hand, the GMRES algorithm is derived in ALGORITHM A.2.

ALGORITHM A.2: GMRES algorithm.

Suppose the following information is given:
$\boldsymbol{A}$ = coefficient matrix, $\vec{b}$ = right hand side, $\vec{x}_0$ = an initial guess,
$\in$ = convergence tolerance, and $kmax$ = maximum iterations allowed.

1. $\vec{r}_0 = \vec{b} - \boldsymbol{A}\vec{x}_0$, $\vec{v}_1 = \dfrac{\vec{r}_0}{\|\vec{r}_0\|_2}$, $r = b = \|\vec{r}_0\|_2$,
   $\vec{g} = (b, 0, ..., 0)^T \in R^{k\max+1}$, $k = 0$.
2. While $r > \in \|\vec{b}\|$ and $k < k\max$
   (a) $k = k + 1$
   (b) Orthogonalization by Arnoldi process
   (c) Transforming $\boldsymbol{H}_k$ into triangular form.
   if $k > 1$, apply $\boldsymbol{Q}_{k-1}$ to kth column of $\boldsymbol{H}_k$.
   (d) Compute residual as, $r = |g_{k+1}|$.
3. Solving the triangular system.
   Set $r_{ij} = h_{ij}$, and $w_i = g_i$ for $1 \le i,j \le k$.
   Solve $\boldsymbol{R}_k \vec{y}_k = \vec{w}_k$.
4. Return solution as, $\vec{x}_k = \vec{x}_0 + \boldsymbol{V}_k \vec{y}_k$.

One disadvantage of the method is that it requires all the basis vectors of the Krylov subspace to be stored as the iteration progress. This means that performing k iterations require storing k vectors of size n, which is undesirable for large problems. To overcome this problem, one can used a restarted version of the GMRES algorithm as given in ALGORITHM A.3.

ALGORITHM A.3: Restarted GMRES algorithm.

1. Define $\in$ and $k\max$. And set $i = 0$ and $\vec{x}_0 = \boldsymbol{0}$.
2. While convergence is not attained,
   $i = i + 1$.
   Use ALGORITHM A.2, with $\begin{cases} \vec{x}_0 = \boldsymbol{0}, & \text{if } i = 1. \\ \vec{x}_0 = \hat{x}_{i-1}, & \text{otherwise.} \end{cases}$
   where $\hat{x}_{i-1}$ is the solution return by previous GMRES solve.
3. Return $\hat{x}_{i+1}$ as the approximate solution by GMRES.

Appendix B

Extracting Order of Singularity for Two and Three- Dimensional Corners and Edges

The singularity behavior of the potential gradients or surface charge distributions in the vicinity of sharp corners and edges is strongly determined by the order of singularity, which is essentially the eigenvalue of the associated eigen-problem of the given geometry. This appendix presents the techniques used to extract the order of singularities for the edges and corners in two and three-dimensional problems.

B.1 Potential Fields in the Vicinity of Two-Dimensional Corner

In the two-dimensional context, a corner is referred to as the intersection point of two planes, which in this case are the adjacent surfaces of a conductor. The corner is placed at the origin *O* and the conductor is assumed to be at a constant potential f_0. Figure B.1 shows a general corner in two-dimensional space.

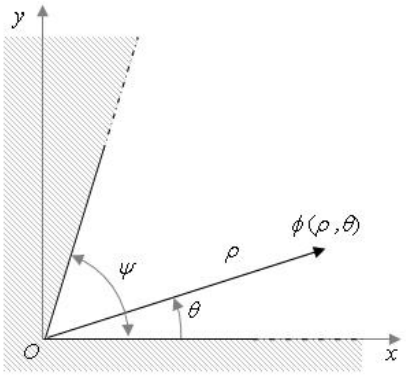


Figure B.1. Two-dimensional corner with opening angle y.

The governing equation for the potential field, in the polar coordinates (r, q) in two-dimensional space, is given by

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial f}{\partial r}\right)+\frac{1}{r^2}\frac{\partial^2 f}{\partial q^2}=0 \qquad \text{(B.1)}$$

By using separation of variables [52], the general solution of (B.1) is

$$f(r,q) = (a_0 + b_0 \ln r)(A_0 + B_0 q) + \sum_{i=1}^{\infty} \left[(a_i r^{a_i} + b_i r^{-a_i})(A_i \cos(a_i q) + B_i \sin(a_i q)) \right] \tag{B.2}$$

where $a's$ are the eigenvalues, which are constrained by the boundary conditions applied at the corner. For uniform Dirchlet boundary condition, (B.2) is reduced to

$$f(r,q) = f_0 + \sum_{i=1}^{\infty} c_i r^{\left(\frac{ip}{\Psi}\right)} \sin\left(\frac{ipq}{\Psi}\right) \tag{B.3}$$

where c_i are the unknown coefficients that depend on the boundary conditions remote from the corner point.

Finally, the normal potential gradients, that is, $\frac{\partial f}{\partial n} = \frac{1}{r}\frac{\partial f}{\partial q}$ at the two adjacent surfaces of the conductor are given by

$$\begin{aligned} \frac{\partial f}{\partial n} &= \sum_{i=1}^{\infty} c_i \left(\frac{ip}{\Psi}\right) r^{\left(\frac{ip}{\Psi}-1\right)}, \quad \text{for } q = 0 \\ \frac{\partial f}{\partial n} &= \sum_{i=1}^{\infty} c_i \left(\frac{ip}{\Psi}\right) r^{\left(\frac{ip}{\Psi}-1\right)} \cos(ip), \quad \text{for } q = \Psi \end{aligned} \tag{B.4}$$

and in the vicinity of the re-entrant corner can be approximated by

$$\frac{\partial f}{\partial n} \approx c_1 \left(\frac{p}{\Psi}\right) r^{\left(\frac{p}{\Psi}-1\right)} \tag{B.5}$$

which may be singular with the order of singularity given by $\left(\frac{p}{\Psi} - 1\right)$.

B.2 Extracting Order of Singularity for Three-Dimensional Corners

Analytical singularity solutions are almost impossible for the three-dimensional corners. Numerical techniques have to be used to determine the order of singularity for arbitrary corners. One such technique is presented as follows.

The singularity solution for a corner can be constructed from the bounded solution of the three-dimensional Laplace problem defined by the intersection of a sphere of radius r_s, with surface ¶S and centered at the corner, with the region around the corner Ω, satisfying the boundary conditions on the appropriate parts of the corner, as depicted in Figure B.2.

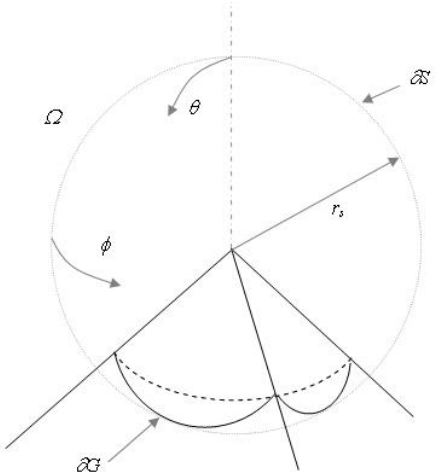


Figure B.2. A corner with apex at the centre of a sphere.

Consider the three-dimensional Laplace equation in the spherical coordinates system,

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial f}{\partial r}\right)+\frac{1}{r^2\sin^2 q}\frac{\partial^2 f}{\partial f^2}+\frac{1}{r^2\sin q}\frac{\partial}{\partial q}\left(\sin q\frac{\partial f}{\partial q}\right)=0 \tag{B.6}$$

Applying the following separation of variables

$$f(r,q,f)=R(r)U(q,f) \tag{B.7}$$

gives,

$$r\frac{d^2}{dr^2}(rR)-a(a+1)R=0,\quad 0<r<r_s \tag{B.8}$$

$$\begin{aligned}[\Delta_q+a(a+1)\boldsymbol{I}]U&=0,\quad (q,f)\in G\\ U&=0,\quad (q,f)\in\partial G_D\\ \frac{\partial U}{\partial n}&=0,\quad (q,f)\in\partial G_N\end{aligned} \tag{B.9}$$

where $G=\P S\Ç W$, $\P G=\P S\Ç\P W$, $\boldsymbol{I}$ is the identity operator. Δ_θ is the Laplace-Beltrami operator given as

$$\Delta_q=\frac{1}{\sin q}\frac{\partial}{\partial q}\left(\sin q\frac{\partial}{\partial q}\right)+\frac{1}{\sin^2 q}\frac{\partial^2}{\partial f^2} \tag{B.10}$$

The solution of (B.8) is straightforward, which is given by

$$R=Ar^a+Br^{-(a+1)} \tag{B.11}$$

where A and B are constants, and a is yet to be determined by solving the eigenvalue problem defined in (B.9). In the following sub-section, we present a numerical technique for solving the eigenvalue problem of a general corner.

B.2.1 Solving the Laplace-Beltrami eigenvalue problem

Finite difference method (FDM) can be used to solve the eigenvalue problem in (B.9). However, from Figure B.2, it can be seen that the problem domain *G* may contains singular solution due to the singularity rays that intersect at the corner at *O*. These singular points have the order of singularity identical to their corresponding rays. Although *U* is not singular, its derivatives are definitely singular. This deteriorates the accuracy of the numerical results.

To alleviate the situation, Bazant [36] suggested removing all the point singularities by expressing $U(q,f)$ (assuming only one singularity point exists) as

$$U(q,f) = [t(q,f)]^p u(q,f) \tag{B.12}$$

where $t(q,f)$ is a chosen function that is nonzero everywhere except on the singularity point, and *p* is the correct singular exponent of the singularity ray. In this case, the function $u(q,f)$ is smooth, and hence FDM can then be used to solve the modified eigen-problem more accurately.

Suppose the singularity ray lies along the pole $q = 0$, the obvious choice for *t* is

$$t = q, \quad \text{or} \quad t = \sin q \tag{B.13}$$

For the general case where the singular ray is located at (q_1, f_1), we can choose

$$t = \sqrt{(q - q_1)^2 + [(f - f_1)\sin q_1]^2} \tag{B.14}$$

Equation (B.14) can be extended to general case where *n* singularity rays exist within the domain,i.e.

$$U(q,f) = \left(t_1^{p_1} t_2^{p_2} .. t_n^{p_n}\right) u(q,f) \tag{B.15}$$

where *p's* are the known singular exponents of the singularity rays, and *t's* are the appropriately chosen functions.

Applying the FDM to the modified eigen-problem leads to the following eigen-matrix problem,

$$\boldsymbol{A}(l)\boldsymbol{U} = \boldsymbol{0}, \quad \text{with } l = a(1+a) \tag{B.16}$$

where $\boldsymbol{A}$ is the coefficient matrix, $\boldsymbol{U}$ is the vector of nodal unknowns and l is the eigenvalue of matrix $\boldsymbol{A}$. Two methods are presented in the following sub-section that can be used to solve the eigen-matrix problem in (B.16).

B.2.2 Solution methods for the eigen-matrix problem

Method A: Reduction to a matrix eigenvalue problem

For FDM, it is noted that l only exists in the diagonal entries of $\boldsymbol{A}$. Hence, (B.16) can be rewritten as

$$(\boldsymbol{A}' - l\boldsymbol{I})\boldsymbol{U} = \boldsymbol{0} \tag{B.17}$$

where $\boldsymbol{A}'$ is a matrix independent of l, and $\boldsymbol{I}$ is the identity matrix. This is a standard eigenvalue problem that can be solved using QR factorization. The two corresponding roots a for each l are

$$a = -\frac{1}{2} \pm \sqrt{\frac{1}{4} + l} \tag{B.18}$$

If l are all real, then the smallest positive l also corresponds to the smallest a, which is the order of singular for the potential field.

Method B: Conversion to non-homogenous equations

Generally, the eigen-problem can be written as

$$\boldsymbol{M}(l)\boldsymbol{U} = \boldsymbol{0} \tag{B.19}$$

This method begins by making an initial guess for l to compute the corresponding matrix $\boldsymbol{M}$, which is then modified by replacing one of the equation, for example the kth equation, with $U_k = 1$. The modified problem, which is now non-homogenous, is then solved, and the solution is substituted back into the original kth equation, that is, $\sum_{i=1}^{n} m_{ki} U_i = Q$. Generally, Q is not equal to zero, unless the assumed l is an eigenvalue of (B.19). Hence, the aim is to find the smallest value of l that makes Q = 0 or near zero.

However, it is noted that the radius of convergence for this approach can be quite small. This means that a good initial guess is required to ensure the method to convergence to the correct eigenvalue. Otherwise, the result has to be scanned in small steps for a large interval, which can be computationally expensive. Therefore, a more efficient approach is to use *method A* to obtain a good initial guess of l, and then use *method B* to refine the solution.

Appendix C

Numerical Integration of Singular Integrals in Three-Dimensional BEM

This appendix summarizes the regularization transformations for the various cases where the collocation point falls on nodes of the element over which integration is performed. The first part deals with cases in which only the fundamental solution is singular. In the second part, both the fundamental solution and the shape functions are singular. In this case, the regularization transformations are first applied to remove the more strongly singularity due to the fundamental solution. The weakly singular shape functions, after the transformations, can be cast into forms that can be effectively treated by the Gauss-Jacobi formulas† [98]. Hence, the second part summarizes these expressions, which helps to determine the appropriate Gauss-Jacobi formulas required.

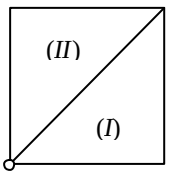
C.1 Regularization Transformations for Treating the Singularity due to Fundamental Solution

The integral concern here is generally of the form

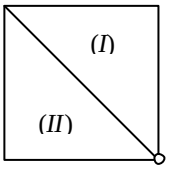
$$I = \sum_{k=1}^{T} \left\{ \int_{-1}^{1}\int_{-1}^{1} f(x_1, x_2) \frac{c(1+h_2)}{\|x - x'\|} dh_1 dh_2 \right\} \tag{5.45}$$

where $f(x_1, x_2)$ is a nonsingular function, T is the number of sub-triangles depending on the collocation point, and c is the constant associated with the Jacobian of transformation that maps x onto h, and also depends on the location of the collocation point. The underlying principle and general mapping functions of this technique are given in Section 5.4.2. In this section, the actual mapping functions are explicitly presented for the various cases, where collocation points falls on different nodes.

(i) Collocation point at corner nodes. c = 0.5 for $T = I, II$.

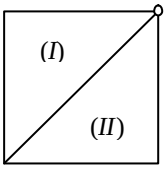


Node 1:
(I): $x_1 = f_1$; $x_2 = f_2$
(II): $x_1 = f_2$; $x_2 = f_1$



Node 2:
(I): $x_1 = -f_2$; $x_2 = f_1$
(II): $x_1 = -f_1$; $x_2 = -f_2$

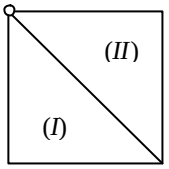
† Gauss-Jacobi formula is given by: $\int_{-1}^{1} (1-x)^a (1+x)^b f(x) dx \approx \sum_{i=1}^{n} f(z_i).v$,
where z_i and v_i are the abscissas and weighs, and a and b are the singular exponents.



Node 3:

(I): $x_1 = -f_1;\quad x_2 = -f_2$

(II): $x_1 = -f_2;\quad x_2 = -f_1$

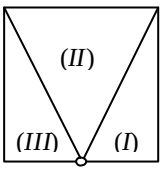


Node 4:

(I): $x_1 = f_2;\quad x_2 = -f_1$

(II): $x_1 = f_1;\quad x_2 = -f_2$

(ii) Collocation point at mid-side nodes. c = 0.25, for T =*I, III,* and c = 0.5 for T = *II.*

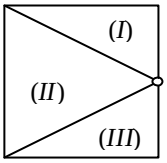


Node 5:

(I): $x_1 = f_3,\quad x_2 = f_2$

(II): $x_1 = f_4,\quad x_2 = f_1$

(III): $x_1 = -f_3,\quad x_2 = f_2$

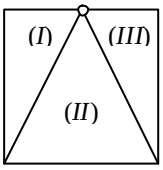


Node 6:

(I): $x_1 = -f_2;\quad x_2 = f_3$

(II): $x_1 = -f_1;\quad x_2 = f_4$

(III): $x_1 = -f_2;\quad x_2 = -f_3$

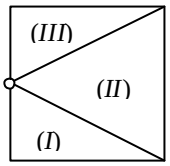


Node 7:

(I): $x_1 = -f_3,\quad x_2 = -f_2$

(II): $x_1 = f_4,\quad x_2 = -f_1$

(III): $x_1 = f_3,\quad x_2 = -f_2$



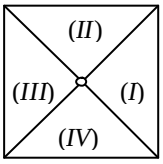
Node 8:

(I): $x_1 = f_2;\quad x_2 = -f_3$

(II): $x_1 = f_1;\quad x_2 = f_4$

(III): $x_1 = f_2;\quad x_2 = f_3$

(iii) Collocation point at center node (9). c = 0.25 for all sub-domains.



Node 9:

(I): $x_1 = f_3;\quad x_2 = f_4$ $\quad (II)$: $x_1 = f_4;\quad x_2 = f_3$

(III): $x_1 = -f_3;\quad x_2 = f_4$ $\quad (IV)$: $x_1 = f_4;\quad x_2 = -f_3$

where $f_1 = h_2$, $f_2 = -1 + \frac{1}{2}(1+h_1)(1+h_2)$, $f_3 = \frac{1}{2}(1+h_2)$, and $f_4 = \frac{1}{2}h_1(1+h_2)$.

C.2 Singularity Expressions for the Singular Shape Functions After the Regularization Transformations

Boundary element integrals , after the regularization transformations, have the following general form

$$I = \int_{-1}^{1}\int_{-1}^{1} f(x_1, x_2) h_i(x_1, x_2)\, dh_1 dh_2 \tag{C.2}$$

where $f(x_1, x_2) = g(x_1, x_2)\dfrac{c(1+h_2)}{\|\boldsymbol{x} - \boldsymbol{x}\|}$ is a nonsingular function. However, $h_i(x_1, x_2)$ may still be singular due to the singular shape functions. There are four possible types of singularity forms, as mentioned in Section 5.4.3:

$$h_1 = \frac{1}{(1+x_i)^{l_E}} \quad \text{for } i = 1 \text{ or } 2. \tag{C.3a}$$

$$h_2 = \frac{r^{a_1}}{(1+x_1)^{l_E}(1+x_2)^{l_E}} \tag{C.3b}$$

$$h_3 = \frac{1}{r^{l_{C2}}} \tag{C.3c}$$

$$h_4 = \frac{r^{a_2}}{(1+x_i)^{l_E}} \quad \text{for } i = 1 \text{ or } 2. \tag{C.3d}$$

where $r = \frac{1}{2}\sqrt{(1+x_1)^2 + (1+x_2)^2}$.

The objectives in this section are to determine: (i) the final expressions of the kernel $h_i(x_1, x_2)$ after the regularization transformations, and (ii) the Gauss-Jacobi formulas needed to evaluate the resulting integrals. Now, consider the different singularity functions given in (C.3a) to (C.3d) separately.

$h_1 = \frac{1}{(1+x_i)^{l_E}}$ for $i = 1$ or 2.

For this form, only one of the two cases is presented (i.e. i = 1 or 2). Without loss in generality, let $h_1 = \frac{1}{(1+x_1)^{l_E}}$. Hence, considering the various cases where the collocation point is located at different nodes.

Node 1:

(I) $h_1 = \frac{1}{(1+h_2)^{l_E}}$

(II) $h_1 = \left[(2)^{l_E}\right]\frac{1}{(1+h_1)^{l_E}}\frac{1}{(1+h_2)^{l_E}}$

Node 2:

(I) $h_1 = \left[\frac{(1-h_1)^b}{(1+x_1)^{l_E}}\right]\frac{1}{(1-h_1)^b}$

(II) $h_1 = \frac{1}{(1-h_2)^{l_E}}$

Node 3:

(I) $h_1 = \frac{1}{(1-h_2)^{l_E}}$

(II) $h_1 = \left[\frac{(1-h_1)^b}{(1+x_1)^{l_E}}\right]\frac{1}{(1-h_1)^b}$

Node 4:

(I) $h_1 = \left[(2)^{l_E}\right]\frac{1}{(1+h_1)^{l_E}}\frac{1}{(1+h_2)^{l_E}}$

(II) $h_1 = \frac{1}{(1+h_2)^{l_E}}$

Node 5:

(I) $h_1 = \frac{1}{(1+x_1)^{l_E}}$

(II) $h_1 = \left[\frac{(1+h_1)^b}{(1+x_1)^{l_E}}\right]\frac{1}{(1+h_1)^b}$

(III) $h_1 = \left[(2)^{l_E}\right]\frac{1}{(1-h_2)^{l_E}}$

Node 6:

$(I),(III)$ $h_1 = \left[\frac{(1-h_1)^b}{(1+x_1)^{l_E}}\right]\frac{1}{(1-h_1)^b}$

(II) $h_1 = \frac{1}{(1-h_2)^{l_E}}$

Node 7:

$$(I)\quad h_1=\left[(2)^{l_E}\right]\frac{1}{(1-h_2)^{l_E}}$$

$$(II)\quad h_1=\left[\frac{(1+h_1)^b}{(1+x_1)^{l_E}}\right]\frac{1}{(1+h_1)^b}$$

$$(III)\quad h_1=\frac{1}{(1+x_1)^{l_E}}$$

Node 8:

$$(I),(III)\quad h_1=\left[(2)^{l_E}\right]\frac{1}{(1+h_1)^{l_E}}\frac{1}{(1+h_2)^{l_E}}$$

$$(II)\quad h_1=\frac{1}{(1+h_2)^{l_E}}$$

Node 9:

$$(I)\quad h_1=\frac{1}{(1+x_1)^{l_E}}$$

$$(II),(IV)\quad h_1=\left[\frac{(1+h_1)^b}{(1+x_1)^{l_E}}\right]\frac{1}{(1+h_1)^b}$$

$$(III)\quad h_1=\left[(2)^{l_E}\right]\frac{1}{(1-h_2)^{l_E}}$$

$$h_2=\frac{r^{a_1}}{(1+x_1)^{l_E}(1+x_2)^{l_E}}$$

This form of singularity exists only in the shape functions of *Corner1* singular element, which is associated with strongly singular corners.

Node 1:

$$(I),(II)\quad h_2=\left\{(2)^{(l_E-a_1)}\left[1+\left(\frac{1+h_1}{2}\right)^2\right]^{\frac{a_1}{2}}\right\}\frac{1}{(1+h_1)^{l_E}}\frac{1}{(1+h_2)^{l_{C1}}}$$

Node 2:

$$(I)\quad h_2=\left[\frac{r^{a_1}(1-h_1)^b}{(1+x_1)^{l_E}}\right]\frac{1}{(1-h_1)^b}\frac{1}{(1+h_2)^{l_E}}$$

$$(II)\quad h_2=\left[(2)^{l_E}r^{a_1}\right]\frac{1}{(1+h_1)^{l_E}}\frac{1}{(1-h_2)^{l_E}(1+h_2)^{l_E}}$$

Node 3:

$$(I)\quad h_2=\left[\frac{r^{a_1}(1-h_1)^b}{(1+x_2)^{l_E}}\right]\frac{1}{(1-h_1)^b}\frac{1}{(1-h_2)^{l_E}}$$

$$(II)\quad h_2=\left[\frac{r^{a_1}(1-h_1)^b}{(1+x_1)^{l_E}}\right]\frac{1}{(1-h_1)^b}\frac{1}{(1-h_2)^{l_E}}$$

Node 4:

$$(I)\quad h_2=\left[(2)^{l_E}r^{a_1}\right]\frac{1}{(1+h_1)^{l_E}}\frac{1}{(1-h_2)^{l_E}(1+h_2)^{l_E}}$$

$$(II)\quad h_2=\left[\frac{r^{a_1}(1-h_1)^b}{(1+x_2)^{l_E}}\right]\frac{1}{(1-h_1)^b}\frac{1}{(1+h_2)^{l_E}}$$

Node 5:

$$(I)\quad h_2=\left[\frac{(2)^{l_E}r^{a_1}}{(1+x_1)^{l_E}}\right]\frac{1}{(1+h_1)^{l_E}}\frac{1}{(1+h_2)^{l_E}}$$

$$(II)\quad h_2=\left[\frac{r^{a_1}(1+h_1)^b}{(1+x_1)^{l_E}}\right]\frac{1}{(1+h_1)^b}\frac{1}{(1+h_2)^{l_E}}$$

$$(III)\quad h_2=\left[(2)^{2l_E}r^{a_1}\right]\frac{1}{(1+h_1)^{l_E}}\frac{1}{(1-h_2)^{l_E}(1+h_2)^{l_E}}$$

Node 6:

$$(I)\quad h_2=\left[\frac{r^{a_1}(1-h_1)^b}{(1+x_1)^{l_E}(1+x_2)^{l_E}}\right]\frac{1}{(1-h_1)^b}$$

$$(II)\quad h_2=\left[\frac{r^{a_1}(1+h_1)^b}{(1+x_2)^{l_E}}\right]\frac{1}{(1+h_1)^b}\frac{1}{(1-h_2)^{l_E}}$$

$$(III)\quad h_2=\left[\frac{(2)^{l_E}r^{a_1}(1-h_1)^b}{(1+x_1)^{l_E}}\right]\frac{1}{(1-h_1)^b}\frac{1}{(1-h_2)^{l_E}}$$

Node 7:

$$(I)\quad h_2=\left[\frac{(2)^{l_E}r^{a_1}(1-h_1)^b}{(1+x_2)^{l_E}}\right]\frac{1}{(1-h_1)^b}\frac{1}{(1-h_2)^{l_E}}$$

$$(II)\quad h_2=\left[\frac{r^{a_1}(1+h_1)^b}{(1+x_1)^{l_E}}\right]\frac{1}{(1+h_1)^b}\frac{1}{(1-h_2)^{l_E}}$$

$$(III)\quad h_2=\left[\frac{r^{a_1}(1-h_1)^b}{(1+x_1)^{l_E}(1+x_2)^{l_E}}\right]\frac{1}{(1-h_1)^b}$$

Node 8:

$$(I)\quad h_2=\left[(2)^{2\,l_E}\,r^{a_1}\right]\frac{1}{(1+h_1)^{l_E}}\frac{1}{(1-h_2)^{l_E}(1+h_2)^{l_E}}$$

$$(II)\quad h_2=\left[\frac{r^{a_1}(1+h_1)^b}{(1+x_2)^{l_E}}\right]\frac{1}{(1+h_1)^b}\frac{1}{(1+h_2)^{l_E}}$$

$$(III)\quad h_2=\left[\frac{(2)^{l_E}\,r^{a_1}}{(1+x_2)^{l_E}}\right]\frac{1}{(1+h_1)^{l_E}}\frac{1}{(1+h_2)^{l_E}}$$

Node 9:

$$(I),(II)\quad h_2=\left[\frac{r^{a_1}(1+h_1)^b}{(1+x_1)^{l_E}(1+x_2)^{l_E}}\right]\frac{1}{(1+h_1)^b}$$

$$(III)\quad h_2=\left[\frac{(2)^{l_E}\,r^{a_1}(1+h_1)^b}{(1+x_2)^{l_E}}\right]\frac{1}{(1+h_1)^b}\frac{1}{(1-h_2)^{l_E}}$$

$$(IV)\quad h_2=\left[\frac{(2)^{l_E}\,r^{a_1}(1+h_1)^b}{(1+x_1)^{l_E}}\right]\frac{1}{(1+h_1)^b}\frac{1}{(1-h_2)^{l_E}}$$

$$h_3=\frac{1}{r^{l_{C2}}}$$

This is a point singularity that exists in *Corner2* singular element.

Node 1:

$$(I),(II)\quad h_3=(2)^{l_{C2}}\left[1+\left(\frac{1+h_1}{2}\right)^2\right]^{-\frac{l_{C2}}{2}}\frac{1}{(1+h_2)^{l_{C2}}}$$

Node 2:

$$(I)\quad h_3=\frac{1}{r^{l_{C2}}}$$

$$(II)\quad h_3=\left[\frac{(1+h_1)^b}{r^{l_{C2}}}\right]\frac{1}{(1+h_1)^b}$$

Node 3:

$$(I),(II)\quad h_3=\left[\frac{(1-h_1)^b}{r^{l_{C2}}}\right]\frac{1}{(1-h_1)^b}$$

Node 4:

$$(I)\quad h_3=\left[\frac{(1+h_1)^b}{r^{l_{C2}}}\right]\frac{1}{(1+h_1)^b}$$

$$(II)\quad h_3=\frac{1}{r^{l_{C2}}}$$

Node 5:

$$(I),(II)\quad h_3=\frac{1}{r^{l_{C2}}}$$

$$(III)\quad h_3=\left[\frac{(1+h_1)^b}{r^{l_{C2}}}\right]\frac{1}{(1+h_1)^b}$$

Node 6:

$$(I)\quad h_3=\frac{1}{r^{l_{C2}}}$$

$$(II)\quad h_3=\left[\frac{(1+h_1)^b}{r^{l_{C2}}}\right]\frac{1}{(1+h_1)^b}$$

$$(III)\quad h_3=\left[\frac{(1-h_1)^b}{r^{l_{C2}}}\right]\frac{1}{(1-h_1)^b}$$

Node 7:

$$(I)\quad h_3=\left[\frac{(1-h_1)^b}{r^{l_{C2}}}\right]\frac{1}{(1-h_1)^b}$$

$$(II)\quad h_3=\left[\frac{(1+h_1)^b}{r^{l_{C2}}}\right]\frac{1}{(1+h_1)^b}$$

$$(III)\quad h_3=\frac{1}{r^{l_{C2}}}$$

Node 8:

$$(I)\quad h_3=\left[\frac{(1+h_1)^b}{r^{l_{C2}}}\right]\frac{1}{(1+h_1)^b}$$

$$(II),(III)\quad h_3=\frac{1}{r^{l_{C2}}}$$

Node 9:

$$(I),(II)\quad h_3=\frac{1}{r^{l_{C2}}}$$

$$(III),(IV)\quad h_3=\left[\frac{(1+h_1)^b}{r^{l_{C2}}}\right]\frac{1}{(1+h_1)^b}$$

$$h_4 = \frac{r^{a_2}}{(1+x_i)^{l_E}} \quad \text{for } i = 1 \text{or } 2.$$

This form of singularity exists in the shape function of *Corner3*. Again without loss in generality, consider $h_4 = \frac{r^{a_2}}{(1+x_1)^{l_E}}$.

Node 1:

$$(I) \quad h_4 = \left(\frac{1}{2}\right)^{a_2}\left[1+\left(\frac{1+h_1}{2}\right)^2\right]^{\frac{a_2}{2}} \frac{1}{(1+h_2)^{l_{C2}}}$$

$$(II) \quad h_4 = \left(\frac{1}{2}\right)^{(a_2-l_E)}\left[1+\left(\frac{1+h_1}{2}\right)^2\right]^{\frac{a_2}{2}} \frac{1}{(1+h_1)^{l_E}} \frac{1}{(1+h_2)^{l_{C2}}}$$

Node 2:

$$(I) \quad h_4 = \left[\frac{r^{a_2}(1-h_1)^b}{(1+x_1)^{l_E}}\right]\frac{1}{(1-h_1)^b}$$

$$(II) \quad h_4 = \left[r^{a_2}\right]\frac{1}{(1-h_2)^{l_E}}$$

Node 3:

$$(I) \quad h_4 = \left[r^{a_2}\right]\frac{1}{(1-h_2)^{l_E}}$$

$$(II) \quad h_4 = \left[\frac{r^{a_2}(1-h_1)^b}{(1+x_1)^{l_E}}\right]\frac{1}{(1-h_1)^b}$$

Node 4:

$$(I) \quad h_4 = \left[(2)^{l_E} r^{a_2}\right]\frac{1}{(1+h_1)^{l_E}} \frac{1}{(1+h_2)^{l_E}}$$

$$(II) \quad h_4 = \left[r^{a_2}\right]\frac{1}{(1+h_2)^{l_E}}$$

Node 5:

$$(I) \quad h_4 = \frac{r^{a_2}}{(1+x_1)^{l_E}}$$

$$(II) \quad h_4 = \left[\frac{r^{a_2}(1+h_1)^b}{(1+x_1)^{l_E}}\right]\frac{1}{(1+h_1)^b}$$

$$(III) \quad h_4 = \left[(2)^{l_E} r^{a_2}\right]\frac{1}{(1-h_2)^{l_E}}$$

Node 6:

$$(I),(III) \quad h_4 = \left[\frac{r^{a_2}(1-h_1)^b}{(1+x_1)^{l_E}}\right]\frac{1}{(1-h_1)^b}$$

$$(II) \quad h_4 = \left[r^{a_2}\right]\frac{1}{(1-h_2)^{l_E}}$$

Node 7:

$$(I) \quad h_4 = \left[(2)^{l_E} r^{a_2}\right]\frac{1}{(1-h_2)^{l_E}}$$

$$(II) \quad h_4 = \left[\frac{r^{a_2}(1+h_1)^b}{(1+x_1)^{l_E}}\right]\frac{1}{(1+h_1)^b}$$

$$(III) \quad h_4 = \frac{r^{a_2}}{(1+x_1)^{l_E}}$$

Node 8:

$$(I),(III) \quad h_4 = \left[(2)^{l_E} r^{a_2}\right]\frac{1}{(1+h_1)^{l_E}} \frac{1}{(1+h_2)^{l_E}}$$

$$(II) \quad h_4 = \frac{r^{a_2}}{(1+h_2)^{l_E}}$$

Node 9:

$$(I) \quad h_4 = \frac{r^{a_2}}{(1+x_1)^{l_E}}$$

$$(II),(IV) \quad h_4 = \left[\frac{r^{a_2}(1+h_1)^b}{(1+x_1)^{l_E}}\right]\frac{1}{(1+h_1)^b}$$

$$(III) \quad h_4 = \left[(2)^{l_E} r^{a_2}\right]\frac{1}{(1-h_2)^{l_E}}$$

Having to explicitly express the kernel h_i's after the regularization transformation, the following Gauss-Jacobi formulas are identified.

(1) $a = l_E$ and $b = 0$

(2) $a = 0$ and $b = l_E$

(3) $a = b$ and $b = 0$

(4) $a = 0$ and $b = b$

(5) $a = 0$ and $b = l_{C1}$

(6) $a = 0$ and $b = l_{C2}$

(7) $a = l_E$ and $b = l_E$

(8) $a = b$ and $b = l_E$

Appendix D

Automatic Identification of Singular Elements in MEMS Device Simulations

In this thesis, MSC/PATRAN (a general pre-processing program) is used to create the input files for the electrostatic analysis. In two-dimensional analysis, the singular nodes and elements can be easily identified and manually selected in the boundary element models. Hence, including the information of the singular elements is rather trivial. However, the situation is not the same in the three-dimensional context, where the complication arises from the extra dimension. Take for example the electrostatic comb drive shown in Figure D.1 (a very common MEMS device that can function as a capacitance sensor or an electrostatic actuator). All the nodes that fall on the sharp edges and corners are singular nodes, and hence the task of collecting them manually is too laborious and almost impossible.

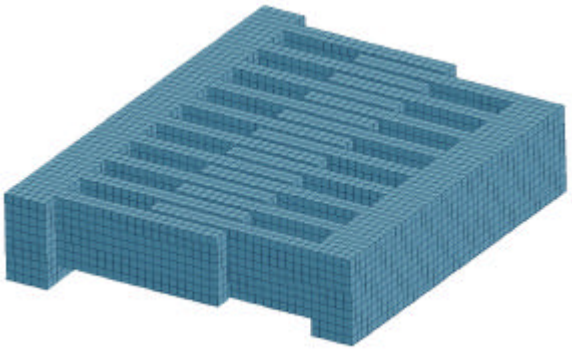


Figure D.1. A three-dimensional model of a comb drive.

This is further complicated by the different types of singular elements that were identified in Section 5.1. To alleviate this problem, a user-defined program, written in PCL (Patran Command Language), is implemented. This program is capable of automatically identifying and classifying the singular elements according to their unique features. The author would like to thank Dr. Su Yi for implementing this pre-processing program. Figure D.2 shows the user-interface of the program that makes it user-friendly.

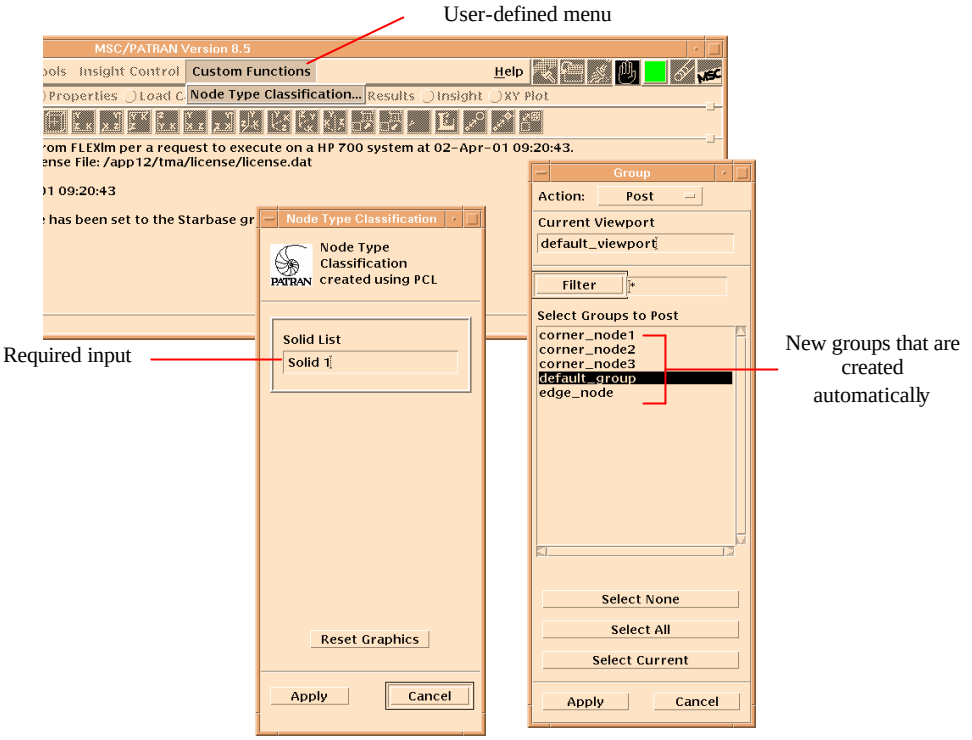


Figure D.2. The user interface created using PCL.

D.1 Classification of Singular Elements

As noted in Section 5.1, there are five different types of singular elements identified for a general rectangular structure. These singular elements possess unique features that allow them to be identified and classified uniquely. The following are some preliminary definitions of singularity geometries in which the classification of singular elements are based on:

(i) A convex edge is singular in nature.

(ii) A concave edge is non-singular in nature.

(iii) A vertex connected to three singular edges is strongly singular.

(iv) A vertex connected to two singular edges is weakly singular.

(v) A vertex connected to one or less singular edge is non-singular.

A summary of the definitions of the singular elements are given as follows:

(1) *Edge*: Contains only one singular edge. The order of singularity (referred to as edge singularity) remains the same along this edge.

(2) *Corner1*: Contains a strongly singular vertex with two adjacent singular edges. The order of singularity increases from edge singularity to the stronger *Corner1* singularity as it approaches the singular corner.

(3) *Corner2*: Contains only a weakly singular vertex and hence, the field is only weakly singular (*Corner2* singularity) at the corner.

(4) *Corner3*: Contains one singular edge and also a weakly singular vertex. In this case, the order of singularity varies from edge singularity to *Corner2* singularity along the singular edge.

(5) *Corner4*: Contains one singular edge and also a non-singular corner. In this case, the singular field would die down at the non-singular corner.

D.2 Automatic Detection of Singular Features of Geometric Model

To identify the singular elements, it is necessary to first efficiently identify the singular features of the geometric model. This involves essentially checking the edges for convexity. As mentioned earlier, a convex edge represents one that is singular in nature.

To determine whether an edge of a model is convex or concave, an understanding of the representation of geometric entities in surface modeling is required. In general, a solid consists of a set of bounding faces with outward directed normal vectors. Each of these faces is formed by one or more closed chain of edges. In the case of a simple trimmed surface, there is only one outer bounding loop of edges. For surfaces with holes, there is an addition of one or more inner bounding loop of edges. Figure D.3 illustrates a simple trimmed surface and one with a inner bounding loop. Also, the ordering of the edges and vertices of a surface follows a standard convention such that the direction of the outer bounding loop of edges is clockwise with reference to the face normal vector $\vec{n}$ while that of the inner bounding loop or loops of edges is anticlockwise.

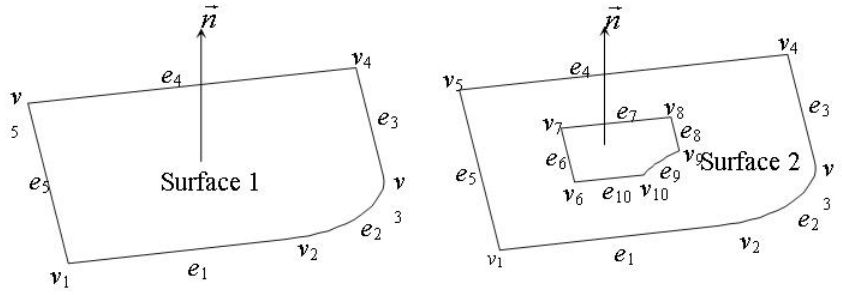


Figure D.3. Trimmed surfaces and their naming convention.

To represent geometric entities in terms of faces, edges and vertices is merely descriptive in nature. To effectively evaluate these entities, some basic concepts of differential geometry are required.

A geometric edge is essentially a 3D curve. The regular parametric representation of the curve is

$$r = r(t) = (x(t), y(t), z(t)) \tag{D.1}$$

The derivative of the vector valued function $r(t)$ is defined as

$$\dot{r}(t) = dr(t)/dt = (dx/dt, dy/dt, dz/dt) \tag{D.2}$$

Higher order derivatives are defined similarly.

An intrinsic property of the curve is the unit tangent vector or gradient of the curve. Suppose s is the natural parameter, that is, the arc length of a curve $r(t)$, then

$$s = \int_0^s |\dot{r}(t)| dt \tag{D.3}$$

It follows that the unit tangent vector of the curve $\boldsymbol{r}(t)$ is defined as

$$T = dr/ds \tag{D.4}$$

By applying a chain rule differentiation, an alternate expression for the unit tangent vector is obtained

$$T = \dot{r}(t)/|\dot{r}(t)| \tag{D.5}$$

In differential geometry, a surface is expressed as

$$r(u,v) = (x(u,v), y(u,v), z(u,v)) \tag{D.6}$$

where u and v are parameters of the surface. A useful property is the surface unit normal vector $\vec{n}$ which is essential for surface interrogation. On differentiating $r(u,v)$ with respect to t gives

$$\dot{r} = \frac{dr}{dt} = \frac{\partial r}{\partial u} \cdot \frac{du}{dt} + \frac{\partial r}{\partial v} \cdot \frac{dv}{dt} = r_u \dot{u} + r_v \dot{v} \tag{D.7}$$

where $\dot{r}$ is the tangent vector of $r(t)$ and r_u and r_v are tangent vectors of isoparametric curves on the domain (u,v-plane) of the parametric surface $\boldsymbol{r}(u,v)$. The three tangent vectors $\dot{r}$, r_u and r_v define a plane called the tangent plane as shown in Figure D.4.

The surface unit normal vector $\vec{n}$ is the unit normal vector to this tangent plane at a particular point, which is obtained by normalizing the vector product of r_u and r_v as

$$\vec{n} = \frac{r_u \times r_v}{|r_u \times r_v|} \tag{D.8}$$

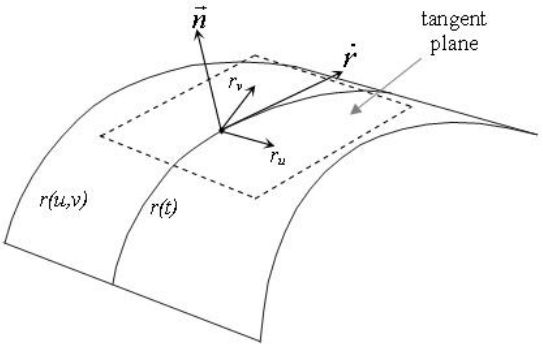


Figure D.4. Illustration of a tangent plane.

Consider the pair of adjacent planar surfaces in Figure D.5 which are orthogonal to each other at an edge $e_{i,j}$ where i signifies the surface index and j the edge index. The edge e_{ij} is convex if the cross product $\vec{n}_i$ and $\vec{n}_k$ is in the same direction as e_{ij}. Consequently, if they are in the opposite direction, then e_{ij} is a concave edge.

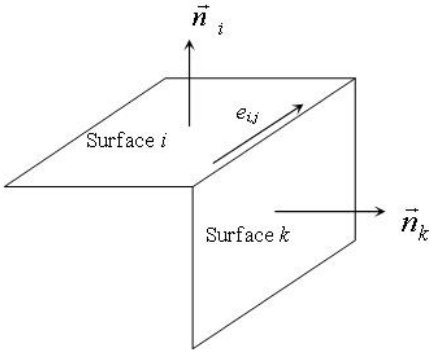


Figure D.5. A pair of orthogonal planar surfaces.

Although this is true for orthogonal planar surface pair with straight edges, such a configuration is very restricted for modeling an object, even though it is observed that many of the MEMS structures are in general 'rectangular'. A method is devised to handle geometric configurations that are not constrained by orthogonal and planar conditions. Consider a pair of general 3D surfaces as shown in Figure D.6 which share a common edge represented by $\gamma(t)$. The unit tangent vector $\boldsymbol{T}$ of $\gamma(t)$ can be evaluated using (D.7) at $t = 0.5$. Next, the surface unit normal vectors $\vec{n}_i$ and $\vec{n}_j$ of surface i and surface j can be evaluated at the parametric values u and v using (D.8) where $r_i(u_i,v_i) = r_j(u_j,v_j) = r(t = 0.5)$.

If the cross product of $\vec{n}_i$ and $\vec{n}_j$ is in the same direction as $\boldsymbol{T}$, then the edge is convex. Consequently, if they are in the opposite direction, then the edge is concave. A special situation arises when the cross product is a null vector. In such a case, the edge is planar. In general, the following criterion apply:

(i) $T \cdot (n_i \times n_j) = +ve$ $\Rightarrow$ edge is convex

(ii) $T \cdot (n_i \times n_j) = -ve$ $\Rightarrow$ edge is concave

(iii) $T \cdot (n_i \times n_j) = 0$ $\Rightarrow$ edge is planar

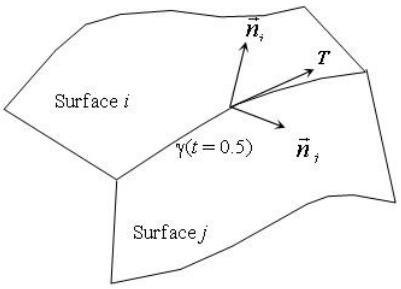


Figure D.6. A pair of non-planar surfaces.

Using these criteria, all the edges of a general solid can be queried for convexity. The flowchart of the algorithm to check the convexity of the edges of a general solid is shown in Figure D.7.

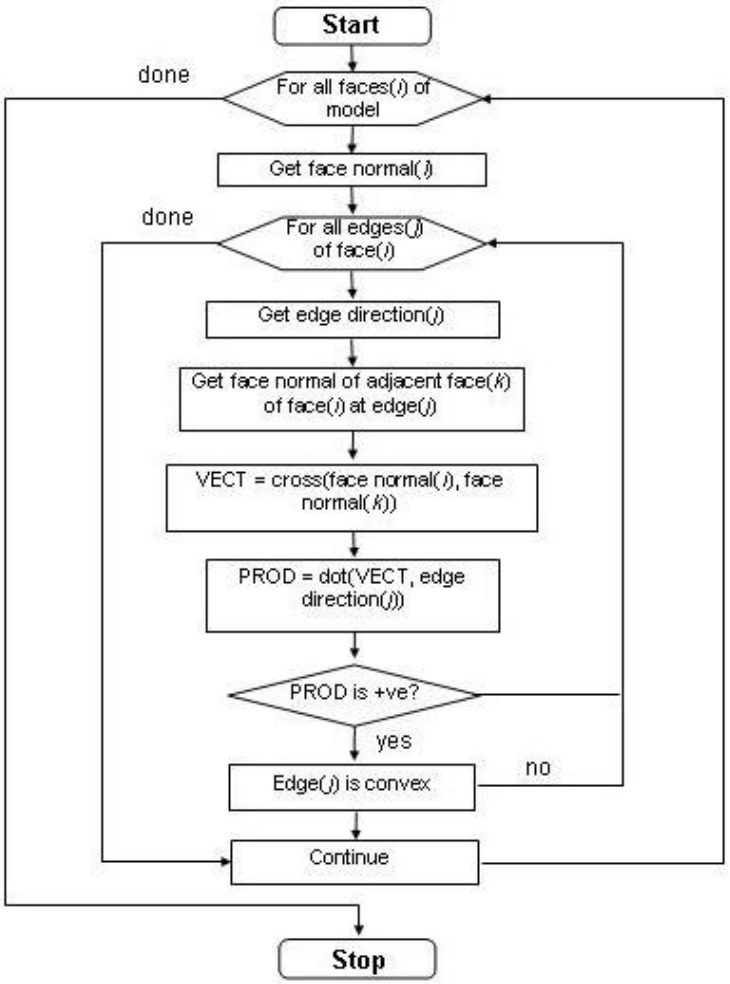


Figure D.7. Flowchart describing the process of checking convexity of edges.

After checking the convexity for all the edges, the nodes of the mesh are classified accordingly. Every type of singular elements described in Section D.1 can be uniquely defined by a combination of these node types. There are altogether four types of different nodes:

(1) A node which lies on a vertex associated with one convex edge.

(2) A node which lies on a vertex associated with two convex edges.

(3) A node which lies on a vertex associated with three convex edge.

(4) A node which lies on a convex edge.

The flowchart of the algorithm to classify the nodes of a mesh according to these four categories is shown in Figure D.8.

D.3 Implementation

The platform used in the implementation of the algorithms described in the previous section is MSC/PATRAN, an industrial standard finite element pre- and post-processor. In particular, the algorithms are coded in the PATRAN Command Language (PCL), which is an integral part of the PATRAN system. Using PCL, access to PATRAN functions and databases is made possible. PCL is also used to create an application user interface, which is depicted in Figure D.2, to enhance the ease of execution of the algorithms. The user is only required to select the solid and activate the 'apply' button. When the execution of the program is completed, four groups are created in the PATRAN database. They are:

(1) `corner_node1` containing nodes lying on vertices associated with one convex edge.

(2) `corner_node2` containing nodes lying on vertices associated with two convex edges.

(3) `corner_node3` containing nodes lying on vertices associated with three convex edges.

(4) `edge_node` containing nodes lying on convex edges.

These groups can then be exported to the required format according to the type of solver used.

To evaluate the performance of the algorithm, the program is run on a HP B200 workstation with 256 MB of RAM. For the comb drive configuration shown in Figure D.1, the program completes the task in only 39.22 seconds.

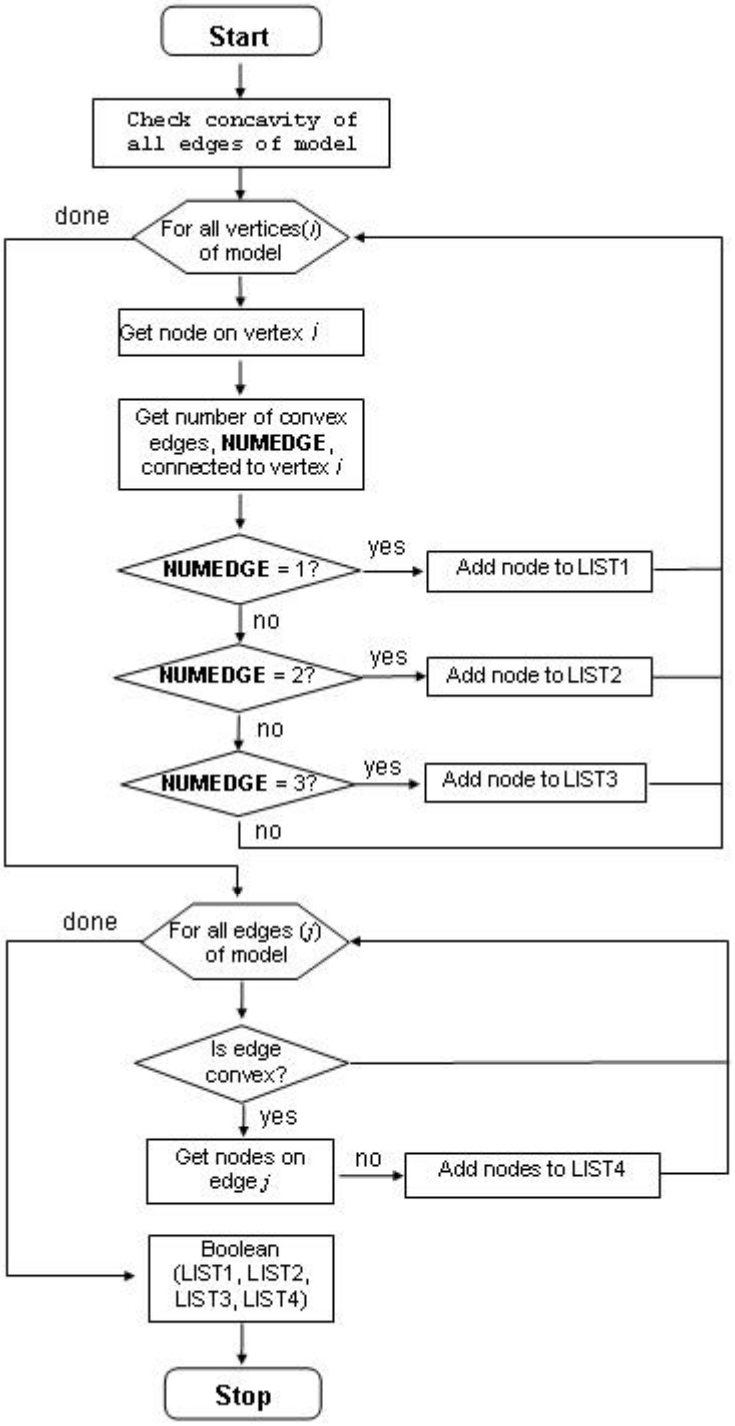


Figure D.8. Flowchart showing the process of classifying singular elements.

Appendix E

Electromechanical Coupling Analysis

To date, many MEMS devices are driven by electrostatic force. The actuation principle can be briefly described as follows. Electrical potentials that are applied on the conductors (actuators) induced electrical charges on their surfaces, which in turns generate electrostatic forces on the conductors. These forces then deform the MEMS structures, which result in mechanical restoring forces in the structures. The deformations of the structures also change the surface charge distributions, and hence the electrostatic forces, which usually further deformed the structures. This process will continue until an equilibrium state is attained, where the electrostatics driving forces are completely balanced by the mechanical restoring forces. This equilibrium state is often referred to as the self-consistent state.

It is obvious that the coupling analysis is nonlinear. Mathematically, the solutions for the two domains can be represented as

$$q = R_E(u, f) \tag{E.1}$$

where $R_E(u, f)$ denotes a linear operator that relates the surface charges density q, for a given conductor geometry u, and the applied electrical potentials f. And,

$$u = R_M(u, P(q)) \tag{E.2}$$

where $R_M(u, P(q))$ represents a linear or nonlinear operator that defines the structural displacements u, for a given the external pressure loading P, which is a function of the surface charge density q.

Note that (E.1) and (E.2) can be solved in a black-box manner. This means that they can be solved individually using different methods as if they are stand-alone problems. One obvious advantage using a black-box approach is the ease of implementation.

In the following section, we briefly outlined a black-box approach, namely the multilevel Newton method [15]. This method is used in this thesis to solve the electromechanical coupling analysis. There also exists other approaches, such as the simple relaxation technique [9], the Surface-Newton Generalized Conjugate Residual (SNGCR) algorithm [10], and the tightly coupled Newton method [13, 14].

E.1 Multilevel Newton Method

In this approach, the coupled equations are solved by employing a nested Multidimensional Newton-Raphson method. The outer-Newton iteration solves the following residual equation:

$$F(u, q) = \begin{Bmatrix} q - R_E(u) \\ u - R_M(q) \end{Bmatrix} = 0 \tag{E.3}$$

where $R_E(u)$ is the charge on the conductors for a given conductors geometry u, and $R_M(q)$ is the structural displacement due to the electrostatic forces generated by the charges q. Hence, the Newton iteration equation is given as

$$-F(u^k, q^k) = \boldsymbol{J}(u^k, q^k) \begin{Bmatrix} d_q \\ d_u \end{Bmatrix}^k \tag{E.4}$$

where d_q and d_u are the variations in the solutions at the k iteration, which can be taken as the convergence indicator, and $\boldsymbol{J}(u, q)$ corresponds to the Jacobian of (E.3) which is given by

$$\boldsymbol{J}(u, q) = \begin{bmatrix} \boldsymbol{I} & -\partial R_E / \partial u \\ -\partial R_M / \partial q & \boldsymbol{I} \end{bmatrix} \tag{E.5}$$

where $\boldsymbol{I}$ is the identity sub-matrix.

Basically, convergence is attained when d_q and d_u are both smaller than a given tolerance. The self-consistent solutions are then computed as,

$$q^* = q^k + d_q^k, \quad \text{and} \quad u^* = u^k + d_u^k \tag{E.6}$$

A summary of the multilevel Newton technique is given in the following algorithm.

ALGORITHM E.1: Multilevel Newton algorithm.

1. Define convergence tolerance, $\in$.
   And set $k = 1$, $u^k = \boldsymbol{0}$ and $q^k = \boldsymbol{0}$.
2. Do,
   Solve (F.4) for d_q and d_u, i.e
   $$-F(u^k, q^k) = \boldsymbol{J}(u^k, q^k) \begin{Bmatrix} d_q \\ d_u \end{Bmatrix}^k.$$
   Compute $q^{k+1} = q^k + d_q^k$.
   Compute $u^{k+1} = u^k + d_u^k$.
   $k = k + 1$.
   while $\|d_q^k\| \geq \in$, or $\|d_u^k\| \geq \in$.
3. Return u^{k+1} and q^{k+1} as the self-consistent solutions.

Notice that the linear system defined by (E.4) in the above algorithm can be solved by using iterative solver, such as the Generalized Minimal RESidual (GMRES) [37]. An important feature of GMRES is that the coefficient matrix, which in this case the Jacobian of residual $\boldsymbol{J}\left(u^k, q^k\right)$, need not be formed explicitly. In other words, the method is matrix-free, and only requires the matrix-vector product $\boldsymbol{J}^k \vec{v}_m$ to be computed, where $\boldsymbol{J}^k$ is the Jacobian of the residual of at the *kth* Newton iteration, and $\vec{v}_m$ is the *mth* basis vector of the Krylov subspace $\boldsymbol{K}_m\left(\boldsymbol{J}^k, \vec{r}_0\right)$ as defined by $span\left\{\vec{r}_0, \left(\boldsymbol{J}^k\right)\vec{r}_0, \left(\boldsymbol{J}^k\right)^2\vec{r}_0, ..., \left(\boldsymbol{J}^k\right)^{m-1}\vec{r}_0\right\}$, with $\vec{r}_0 = -F\left(u^k, q^k\right)$. Hence, using (E.5), the matrix-vector product is explicitly expressed as

$$\boldsymbol{J}(u,q)\vec{v}_m = \begin{bmatrix} \boldsymbol{I} & -\partial R_E/\partial u \\ -\partial R_M/\partial q & \boldsymbol{I} \end{bmatrix} \begin{Bmatrix} \vec{v}_{q,m} \\ \vec{v}_{u,m} \end{Bmatrix} = \begin{bmatrix} \vec{v}_{q,m} - \dfrac{\partial R_E}{\partial u} * \vec{v}_{u,m} \\ \vec{v}_{u,m} - \dfrac{\partial R_M}{\partial q} * \vec{v}_{q,m} \end{bmatrix} \tag{E.7}$$

where $\vec{v}_{q,m}$ and $\vec{v}_{u,m}$ are the components of $\vec{v}_m$ that are associated with the charge q in the electrostatic analysis and the displacement u in the mechanical analysis respectively. The derivative terms in (E.7) can be approximated by finite-difference as follows:

$$\frac{\partial R}{\partial x} * \vec{v}_x \approx \frac{1}{\Delta_x}\left[R\left(x + \Delta_x * \vec{v}_x\right) - R(x)\right], \quad \text{for } x = u \text{ or } q. \tag{E.8}$$

where the matrix-free parameter Δ is a small value, and is suggested to be [15]

$$\Delta_x = \text{sign}\left(x * \vec{v}_x\right) * \min\left(1, \frac{a\|x\|}{\|\vec{v}_x\|}, \frac{b\|R(x)\|}{\|\vec{v}_x\|}\right) \tag{E.9}$$

with $a \in (0.01, 0.5)$ and $b \in (0.1, 1.0)$.

Therefore, (E.7) becomes

$$\boldsymbol{J}(u,q)\vec{v}_m \approx \begin{bmatrix} \vec{v}_{q,m} - \dfrac{1}{\Delta_u}\left[R_E\left(u + \Delta_u * \vec{v}_{u,m}\right) - R_E(u)\right] \\ \vec{v}_{u,m} - \dfrac{1}{\Delta_q}\left[R_M\left(q + \Delta_q * \vec{v}_{q,m}\right) - R_M(q)\right] \end{bmatrix} \tag{E.10}$$

Notice that $R_E\left(u + \Delta_u * \vec{v}_{u,m}\right)$ and $R_M\left(q + \Delta_q * \vec{v}_{q,m}\right)$ are simply the solutions for the charge q and displacement u, when subjected small perturbations of magnitudes $\Delta_u * \vec{v}_{u,m}$ and $\Delta_q * \vec{v}_{q,m}$ respectively. Hence, they can be solved outside the GMRES iteration. The matrix-vector product in (E.10) can be obtained using the following algorithm.

ALGORITHM E.2: Computation of the matrix-vector product.

Given the parameters:
Δ_u, Δ_q using (E.8), and $\vec{v}_{u,m}$, and $\vec{v}_{q,m}$ from the *mth* GMRES iteration.
Compute the following solutions,

$$\left.\begin{aligned} q_1 &= R_E(u, f) \\ q_2 &= R_E\left(u + \Delta_u * \vec{v}_{u,m}, f\right) \end{aligned}\right\} \text{ using electrostatic solver.}$$

$$\left.\begin{aligned} u_1 &= R_M(u, P(q)) \\ u_2 &= R_M\left(u, P\left(q + \Delta_q * \vec{v}_{q,m}\right)\right) \end{aligned}\right\} \text{ using elastomechanics solver.}$$

Finally compute the matrix-vector product as

$$\boldsymbol{J}(u, q)\vec{v}_m = \begin{Bmatrix} \vec{v}_{q,m} - \dfrac{1}{\Delta_q}(q_2 - q_1) \\ \vec{v}_{u,m} - \dfrac{1}{\Delta_u}(u_2 - u_1) \end{Bmatrix}$$

Basically, ALGORITHM E.2 states that at each GMRES iteration, one require to compute two black-box solves, that is, $R_E\left(u + \Delta_u * \vec{v}_{u,m}, f\right)$ and $R_M\left(u, P\left(q + \Delta_q * \vec{v}_{q,m}\right)\right)$. Hence, the efficiency of the individual solvers has great impact on the overall efficiency of this method.

E.2 Finite Element and Boundary Element Meshes

For coupling analysis, two sets of element meshes are generated. There is a finite element volume mesh of the structure that is required by the mechanical solver, and also a boundary element surface mesh used by the electrostatic solver. The two meshes are associated with each other as they share the same set of nodes on the free-surfaces of the structures, where the coupling effects occur. One simple approach is to extract the boundary element mesh from the finite element mesh, that is, the faces of the finite elements that coincide with the free-surfaces of the structures are regarded as boundary elements. However, it is noted that for a given finite element mesh, this way of creating the boundary element mesh results in different problem sizes for the boundary element analysis using different types of boundary elements.

E.3 Equivalent Nodal Forces

Electrostatic analysis computes the surface charge density distributions induced on the surfaces of the structures, which is then used to derive the electrostatic pressure distributions acting on the structure. The pressure loading has to be converted into nodal forces in the mechanical analysis to solve for the

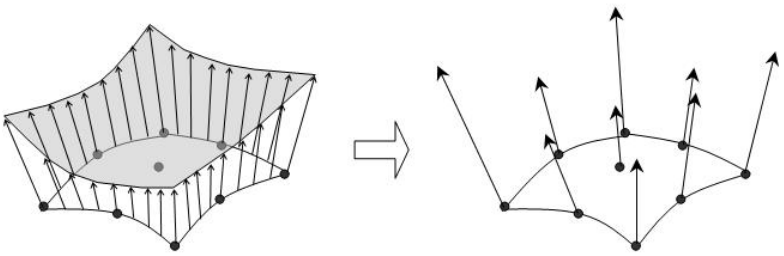
deformation of the structures. The transformation of the distributed pressure loading to its equivalent nodal forces can be done by equating the work done by the two systems of forces, as shown in Figure E.1, that is,

$$\sum_{i=1}^{n} F_i \hat{u}_i = \int_{\Gamma} p(x_1, x_2) u(x_1, x_2) d\Gamma(x_1, x_2) \tag{E.12}$$

where the left hand side of (E.12) corresponds to the work done by the nodal forces F_i, and the right hand side is that due to the pressure loading $p(x_1, x_2)$. By expressing the displacement variations $u(x_1, x_2)$ in terms of the nodal displacement $\hat{u}_i$, that is, $u(x_1, x_2) = \sum_{i=1}^{n} N_i(x_1, x_2)\hat{u}_i$, the equivalent nodal forces are then derived as

$$F_i = \int_{-1}^{1}\int_{-1}^{1} N_i(x_1, x_2) p(x_1, x_2) |J(x_1, x_2)| dx_1 dx_2 \tag{E.13}$$

where $|J(x_1, x_2)|$ is the Jacobian of transformation that maps the element from global coordinates to its intrinsic ones. The equivalent nodal forces computed in (E.13) act in the direction normal to the surface of the structure, but they can be easily resolved into their global coordinate components based on the geometry of the element, namely its surface normal vector.



(a) (b)

Figure E.1. (a) Distributed pressure loading and (b) equivalent nodal forces, acting on an element.

Appendix F

Multipole Expansion Formulas

The multipole expansion given in (6.1) is a complex value function. To avoid complex arithmetic, it is rewritten in the real valued expression, by combining the complex conjugates. This is derived in Section F.1. This appendix also presents the recursive formulas for the associated Legendre functions and trigonometric functions, which is used to accelerate the calculations of the spherical harmonics. It also gives the symmetry properties of these functions that are exploited to avoid computing the response functions $\frac{Y_n^m}{R^{n+1}}$ for the whole problem domain.

F.1. Real Valued Multipole Expansion

Consider the truncated multipole expansion in (6.1), that is,

$$f(\boldsymbol{x}) \approx \sum_{n=0}^{p} \sum_{m=-n}^{n} M_n^m \frac{Y_n^m(q,f)}{R^{n+1}} \tag{F.1}$$

The multipole moments M_n^m and spherical harmonics $Y_n^m(q,f)$ can be explicitly expanded into their real and imaginary components as

$$M_n^m = \sqrt{\frac{(n-|m|)!}{(n+|m|)!}} \left\{ m_{nm}^{(re)} - i m_{nm}^{(im)} \right\} \tag{F.2}$$

where $m_{nm}^{(re)} = \int F(x') \cos(mf') d^3x'$, $m_{nm}^{(im)} = \int F(x') \sin(mf') d^3x'$, and $F(x') = r(x') P_n^{|m|}(\cos q')(r')^n$.

And the s pherical harmonics is defined as

$$Y_n^m = \sqrt{\frac{(n-|m|)!}{(n+|m|)!}} \left\{ y_{nm}^{(re)} + i y_{nm}^{(im)} \right\} \tag{F.3}$$

with $y_{nm}^{(re)} = P_n^{|m|}(\cos q) \cos(mf)$, and $y_{nm}^{(im)} = P_n^{|m|}(\cos q) \sin(mf)$.

Finally, by substituting (F.2) and (F.3) back into (F.1) gives the real valued multipole expansion, as

$$f(\boldsymbol{x}) \approx \sum_{n=0}^{p} \sum_{m=0}^{n} \frac{c_n^m}{R^{n+1}} \frac{(n-|m|)!}{(n+|m|)!} \left[m_{nm}^{(re)} y_{nm}^{(re)} + m_{nm}^{(im)} y_{nm}^{(im)} \right] \tag{F.4}$$

where $c_n^m = \begin{cases} 1, & m = 0; \\ 2, & \text{otherwise.} \end{cases}$

F.2. Recurrence Formulas for Associated Legendre and Trigonometric Functions

To accelerate the computations of the spherical harmonics functions Y_n^m, the following recurrence formulas can be used:

Associated Legendre functions, $P_n^m(\cos\theta)$ *for* $0 \le \theta \le \frac{\pi}{2}$

$$P_n^n(\cos\theta) = \frac{(2n)!}{(2^n)n!}(-\sin\theta)^n, \quad n \ge 0 \tag{F.5a}$$

$$P_n^{n-1}(\cos\theta) = (2n-1)\cos\theta\, P_{n-1}^{n-1}(\cos\theta), \quad n \ge 1 \tag{F.5b}$$

$$P_n^m(\cos\theta) = \frac{1}{(n-m)}\left[(2n-1)\cos\theta\, P_{n-1}^m(\cos\theta) - (n+m-1)P_{n-2}^m(\cos\theta)\right], \quad 0 \le m \le n-2 \tag{F.5c}$$

Trigonometric functions

$$\cos(m\phi) = 2\cos\phi\cos(m-1)\phi - \cos(m-2)\phi \tag{F.6a}$$

$$\sin(m\phi) = 2\cos\phi\sin(m-1)\phi - \sin(m-2)\phi \tag{F.6b}$$

F.3. Symmetry Properties of Associated Legendre and Trigonometric Functions

These symmetry properties are useful when evaluating the spherical harmonics for the full angular ranges, that is, for $0 \le \theta \le \pi$ and $0 \le \phi \le 2\pi$. Consider a point in the first quadrant with the coordinates of (R,θ,ϕ), the following symmetry relation holds for the symmetry points in the other quadrants:

Associated Legendre functions, *for symmetry point at* $(R,\pi-\theta,\phi)$

$$P_n^m(\cos(\pi-\theta)) = \begin{cases} P_n^m\cos(\theta) & \text{if } (n+m) \text{ is even} \\ -P_n^m\cos(\theta) & \text{otherwise.} \end{cases} \tag{F.7}$$

Trigonometric functions

$$\begin{aligned}\cos(m(p-f)) &= (-1)^m \cos(mf)\\ \sin(m(p-f)) &= (-1)^{1+m} \sin(mf)\end{aligned}, \quad \text{for point at } (R,q,p-f) \tag{F.8a}$$

$$\begin{aligned}\cos(m(p+f)) &= (-1)^m \cos(mf)\\ \sin(m(p+f)) &= (-1)^m \sin(mf)\end{aligned}, \quad \text{for point at } (R,q,p+f) \tag{F.8b}$$

$$\begin{aligned}\cos(-mf) &= \cos(mf)\\ \sin(-mf) &= -\sin(mf)\end{aligned}, \quad \text{for point at } (R,q,-f) \tag{F.8c}$$

Hence, by using these symmetry properties, the cost of evaluating the response functions $\frac{Y_n^m}{R^{n+1}}$ is tremendously reduced.