FAST BOUNDARY ELEMENT METHOD SOLUTIONS FOR THREE DIMENSIONAL LARGE SCALE PROBLEMS

A Dissertation
Presented to
The Academic Faculty

By

Jian Ding

In Partial Fulfillment
Of the Requirements for the Degree
Doctor of Philosophy in the
George W. Woodruff School of Mechanical Engineering

Georgia Institute of Technology

December, 2004

FAST BOUNDARY ELEMENT METHOD SOLUTIONS FOR THREE DIMENSIONAL LARGE SCALE PROBLEMS

Approved by:

Assistant Professor Wenjing Ye, Advisor
School of Mechanical Engineering

Professor Peter Hesketh
School of Mechanical Engineering

Professor Jianmin Qu
School of Mechanical Engineering

Dr. Leonard J. Gray
Oak Ridge National Laboratories

Assistant Professor Peter Mucha
School of Mathematics

Date Approved:
December 23, 2004

Dedicated to my dear parents

ACKNOWLEDGEMENT

Until after I finished the study for my Ph.D. degree, did I realize the hard work and relentless effort needed for such a title with honors.

My advisor, Dr. Wenjing Ye, guided me through this path, which is filled with thistles and thorns as well as flowers and happiness, using her patience, love and care for her students, her insight in issues involved in the research. She taught me how to find the right direction when I was standing at the cross-road and wondering. She is the best model for me in my lifelong journey of career development. Thanks.

I would like to acknowledge valuable discussions with all my committee members, Dr. Leonard J. Gray at Oak Ridge National Lab, Dr. Peter Hesketh in MEMS group, Dr. Jianmin Qu in Mechanics of materials group, and Dr. Peter Mucha in Math department.

I also want to thank my parents who were my backbone in finishing this study. They encouraged me whenever I was depressed, supported me by sharing my burdens. They are the most selfless parents in the world. I hope my little progress can reward them to some extent.

My husband had some precious discussions with me about my research. His love and support makes my dream come true. Thank you, Zhiwei.

My colleagues here in the MEMS group in Georgia Tech also helped me to gather information and solve the problems in my research. I would like to thank Nathan Masters especially for his kind help and many discussions with him from which I learned a lot.

TABLE OF CONTENTS

ACKNOWLEDGEMENT iv

LIST OF TABLES vii

LIST OF FIGURES ix

SUMMARY xiii

CHAPTER 1 INTRODUCTION 1

1.1 Motivation 1

1.2 The Boundary Element Method 3

1.3 Precorrected-FFT Technique 5

1.3.1 Projection and interpolation 8
1.3.2 Precorrection 8
1.3.3 Nearby interactions 9
1.3.4 Complexity 9

1.4 Thesis Organization 10

CHAPTER 2 A FAST INTEGRAL APPROACH FOR DRAG FORCE CALCULATION DUE TO OSCILLATORY SLIP STOKES FLOW 12

2.1 Introduction 12

2.2 Governing Equations, Slip Models and the Boundary Integral Formulations 17

2.2.1 Governing equations 17
2.2.2 Slip models 18
2.2.3 The boundary integral formulations 21

2.3 Numerical Methods and Implementation 28

2.3.1 Steady kernels 32
2.3.2 Remaining kernels 35

2.4 Results and Discussion 35

2.4.1 Case study: A sphere oscillating in an unbounded fluid 36

2.4.2 Application 1: Parallel plates 44
2.4.3 Application 2: Micro comb structure 49

2.5 Conclusions 51

CHAPTER 3 A FAST BEM SOLVER FOR NON-HOMOGENEOUS AND NONLINEAR PROBLEMS IN 3-D COMPLEX DOMAINS 52

3.1 Introduction 52

3.1.1 Domain discretization free methods 54
3.1.2 Cell-based direct integration methods 57
3.1.3 The auxiliary domain method 59

3.2 Proposed Volume Integration Approach 62

3.2.1 Evaluation of $\int_{C_i} \widetilde{b}(\mathbf{y}) G(\mathbf{x},\mathbf{y}) dv(\mathbf{y})$ for faraway interactions 64
3.2.2 Implementation schemes for nearby interactions 67
3.2.3 Complexity of volume integration 76

3.3 Results For Volume Integration 77

3.3.1 Singular integrals – convergence study 77
3.3.2 Non-singular integrals – direct method versus accelerated method 78
3.3.3 The complete volume integral 81

3.4 The Iterative Approach for Nonlinear Problems And The Formulations 85

3.5 Benchmark Problems 89

3.5.1 Poisson problems 91
3.5.2 Helmholtz problem and nonlinear problem 98
3.5.3 Comparison with existing approach 105

3.6 Conclusions 106

REFERENCES 109

VITA 113

LIST OF TABLES

Table 1. Calculation of $\int_{\Gamma}^{cpv} \frac{\hat{x}_i^2 \hat{x}_k n_k}{r^5} ds$ via the semi-analytic approach. 34

Table 2. Errors of drag force obtained from slip code I comparing with [41]. $\left(\frac{|F_{drag} - F_{ref}|}{F_{ref}} \times 100\%\right)$ 38

Table 3. Errors of drag force obtained from slip code II comparing with [41]. $\left(\frac{|F_{drag} - F_{ref}|}{F_{ref}} \times 100\%\right)$ 38

Table 4. Relative errors of the simulated drag forces compared with the Bassat's solutions. 39

Table 5. Relative errors in drag forces obtained from the second-order slip model compared with the first-order slip model ($\frac{|F|_{b=-1.8571} - F|_{b=0}|}{F|_{b=0}}$) for steady case. 40

Table 6. Simulated drag forces acting on the oscillating sphere based on Slip Model I (σ_v = 1.0, $b = -1.8571$, *frequency* = 0.01). 42

Table 7. Simulated drag forces acting on the oscillating sphere based on Slip Model II (σ_v = 1.0, *frequency* = 0.01). 43

Table 8. Drag forces per unit viscosity (10^{-6} m^2/sec) for plate problem obtained from Model I (σ =1.0, b=0, *frequency* =19200Hz). 46

Table 9. Drag forces per unit viscosity (10^{-6} m^2/sec) for plate problem obtained from Model II (σ =1.0, *frequency* =19200Hz). 46

Table 10. Traction force (per unit viscosity per area) at the center of the bottom face of the oscillating plate (10^{-6} /sec). 47

Table 11. Drag forces per unit viscosity on oscillating plates with different thickness (Kn = 0.03, σ =1.0). 48

Table 12. Drag forces per unit viscosity on an oscillating comb structure (*frequency* =19200Hz). 50

Table 13. Coordinates of the vertices in different coordinate systems 69

Table 14. Accuracy of different integration schemes. 84

Table 15. CPU time of different schemes. 84

Table 16. Number of volume elements 91

Table 17. Results for the Helmholtz problem on domain 1. 98

Table 18. Results for the Helmholtz problem on domain 2. 100

Table 19. Results for the nonlinear problem on domain 1 102

Table 20. Results for the nonlinear problem on domain 2 104

Table 21. Weight of different integrals in step 1 for sphere domain 105

Table 22. Comparison with existing approach 106

LIST OF FIGURES

Figure 1. (a) Side view of a sphere discretized into 320 panels, with spatial decomposition into a 3×3×3 array of cubes. (b) Superimposed grid charges corresponding to the cube decomposition of (a). In each cell, a 3×3×3 array of grid charges is used to represent the long-range potential of the charged panels in the cell. Some of the grid charges are shared among cells. Note that the grid is "coarser" than the triangular panels used to discretize the sphere. The grid extends outside the problem domain because the number of grid points is required to be a factor of two [31]. 7

Figure 2. A 2-D pictorial representation of the four steps of the precorrected-FFT method. Interactions with nearby panels (in the grey area) are computed directly, interactions between distant panels are computed using the grid [31]. 7

Figure 3. Illustration of the fluid domain and its boundary. 23

Figure 4. A discretized oscillating sphere. 36

Figure 5. Convergence plot of drag force using slip code I (Model I, $Kn = 0$, $b=0$, $\sigma_v = 1.0$). 37

Figure 6. Convergence plot of drag force using slip code II (Model II, $Kn = 0$, $\sigma_v = 1.0$). 37

Figure 7. Drag forces per unit viscosity simulated using the first-order slip model for steady case ($\sigma_v = 1.0$, $b=0$). 39

Figure 8. Drag forces per unit viscosity simulated using the second-order slip model for steady case ($\sigma_v = 1.0$, $b = -1.8571$). 40

Figure 9. The computation time versus the number of panels for steady case. 41

Figure 10. Convergence plot of drag force using slip code I for unsteady case (Model I, $b = 0$, $\sigma_v = 1.0$). 42

Figure 11. Convergence plot of drag force using slip code II for unsteady case (Model II, $\sigma_v = 1.0$). 43

Figure 12. A pair of parallel plates 44

Figure 13. Velocity profile for a pair of resonating parallel plates 45

Figure 14. SEM picture of a micro resonator (top view)..... 50

Figure 15. A discretized comb drive structure. 50

Figure 16. Near-field and far-field for the domain integral for the node highlighted by the dot. Notice that near-field elements (light grey) are far less than far-field elements (dark grey). Matrix entries are stored only for the near-field elements (light gray) [16]..... 57

Figure 17. Schematic of a complex, multiply-connected domain in two dimensions. A domain integral must be evaluated over the computational domain Ω_0. There are N auxiliary domains Ω_i 59

Figure 18. A mesh where only Ω_0 is discretized using an unstructured quadrilateral mesh (left), and the corresponding discretization, where the augmented domain Ω_+ and each of the subdomain Ω_i are discretized (right). Notice that elements used to discretize the augmented domain in some cases span both Γ_0 and at least one Γ_i..... 60

Figure 19. Side view of a sphere encompassed by a parallelpiped with a 3×3×3 array of cubes. 63

Figure 20. 2-D illustration of relationship between problem domain and the uniform FFT grid - Shaded area is the problem domain Ω, B is the uniform grid. 64

Figure 21. (a) A partially filled boundary cube, Ω_i is the problem domain; (b) A partially filled cube with surface panels. 67

Figure 22. Transformation of a prism to a regular domain. 69

Figure 23. (a) A singular prism being divided into three smaller prisms; (b) A smaller prism with the evaluation point located on one of its vertices; (c) Coordinate transformation: prism 1 is transformed into a regular domain. Similar transformations can be done for prisms 2 and 3. 70

Figure 24. Illustration of scenario in which overlapping errors may occur. 72

Figure 25. Panel truncation using cube boundaries. 73

Figure 26. Use three angles formed by the evaluation point and three vertices of the panel to judge if a panel is singular or not. 74

Figure 27. Illustration of the compensation volume. 74

Figure 28. Compensation volume for irregular boundary in nearby interactions. 75

Figure 29. Coordinate transformation for a general quadrilateral cell. 76

Figure 30. (a) Illustration for direct approach; (b) Illustration for precorrected-FFT acceleration: 1) Projection, 2) Convolution, 3) Interpolation, 4) Nearby interactions. 77

Figure 31. Convergence of evaluation schemes for singular volume integrals. 78

Figure 32. The minimum distance from the evaluation point to a cube. 79

Figure 33. Illustration of the FFT accelerated approach. 79

Figure 34. Convergence for the case $b = 1$. 80

Figure 35. Convergence for the case when $b = x^2y$. 81

Figure 36. A solid sphere domain: (a) original surface mesh (768 panels), (b) new surface mesh (2322 panels), (c) sphere with the 3-D uniform grid. 83

Figure 37. Subdivision of singular prism and singular cube. 88

Figure 38. L_2-norm errors in $\partial u/\partial n$ of Poisson problem 1. 92

Figure 39. L_2-norm errors in $\partial u/\partial n$ of the Laplace problem. 93

Figure 40. CPU time for solving the Poisson problem 1 on a spherical domain..... 94

Figure 41. CPU time plot for approach III 95

Figure 42. Surface mesh of domain 2 - A solid ellipsoid with two spherical exclusions. The total number of surface panels shown is 1152. 96

Figure 43. L_2-norm error as function of surface discretization for Poisson problem 2 on two domains. 97

Figure 44. CPU time as function of surface discretization for Poisson problem 2 on two domains. 97

Figure 45. Results for the Helmholtz problem on domain 1: (a) L_2-norm errors, (b) CPU time. 99

Figure 46. Results for the Helmholtz problem on domain 2: (a) L_2-norm errors, (b) CPU time. 100

Figure 47. Results for the nonlinear problem on domain 1: (a) L_2-norm errors, (b) CPU time.....103

Figure 48. Results for the nonlinear problem on domain 2: (a) L_2-norm errors, (b) CPU time.....104

SUMMARY

Efficiency is one of the key issues in numerical simulation of large-scale problems with complex 3-D geometry. Traditional domain based methods, such as finite element methods and finite difference methods, may not be suitable for these problems due to, for example, the complexity of mesh generation. The Boundary Element Method (BEM), based on integral formulations, offers one possible solution to this issue. By discretizing only the surface of the domain, this method greatly reduces the complexity of mesh generation as well as the size of the problem. However, to date, successful applications of the BEM are mostly limited to linear and continuum problems. The challenges in the extension of the BEM to nonlinear problems or problems with non-continuum boundary conditions include, but are not limited to, the lack of appropriate boundary integral formulations and the difficulties in the treatment of the volume integrals that result from the nonlinear terms. In this thesis work, new approaches and techniques based on the BEM have been developed for 3-D nonlinear problems and Stokes problems with slip boundary conditions.

For nonlinear problems, a major difficulty in applying the BEM is the treatment of the volume integrals in the integral formulation. An efficient approach, based on the precorrected-FFT technique, is developed to evaluate the volume integrals. In this approach, the 3-D uniform grid constructed initially to accelerate surface integration is used as the baseline mesh for the evaluation of volume integrals. The cubes enclosing part of the boundary are further discretized by projection of surface panels onto the boundary of corresponding cubes. No additional volume discretization of the interior

cubes is necessary. Therefore, complicated volume discretization for the interior domain is avoided. This grid is also used to accelerate volume integration to further reduce the computational cost. Based on this approach, accelerated BEM solvers for non-homogeneous problems and nonlinear problems are developed and tested on several problems. The results are compared with analytical solutions and good agreement has been achieved. Good performance is achieved comparing with current approaches.

Stokes problems with slip boundary conditions are of particular importance in micro gas flows such as those encountered in MEMS devices. An efficient approach based on the BEM combined with the precorrected-FFT technique has been proposed and various techniques have been developed to solve these problems. As the applications of the developed method, drag forces on oscillating objects immersed in an unbounded slip flow are calculated and validated with either analytic solutions or experimental results.

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

The fast pace of development in MEMS devices and fabrication processes requires a thorough understanding of the underlying physics to fully realize their potentials. This poses a great challenge in numerical modeling and simulation since the problems are usually coupled nonlinear problems and have complex three-dimensional geometry.

To treat nonlinear partial differential equations, domain methods are very popular in a wide range of engineering problems. Of these finite element methods, finite difference methods, and spectral methods have become widely known and have gained widespread acceptance among the engineering analysis community. One of the major difficulties in using these very powerful analyses is the formidable demand on the data preparation effort and thus the total computing costs. The Boundary element method (BEM) was developed to reduce the complexity in mesh generation, and it has been established as a powerful numerical method for solving engineering problems, including but not limited to electromagnetics, elasticity, acoustics, potential and viscous flow applications. The essence of BEM is the transformation of the differential equations into equivalent sets of integral ones as the first step in their solution [3]. Such an operation, if successful, would involve only values of the variables at the extremes of the range of integration so that any discretization scheme needed subsequently would only involve subdivisions of the bounding surface of a body. The solution variables will then vary continuously throughout the region and all approximations of geometry, etc., will only occur on its

outer boundaries. Since the formulation only involves unknowns on surfaces, BEM can handle rapid transitions and steep gradients of the fields, and it reduces the dimensionality of the problem by one, leading to a much smaller system of algebraic equations than any domain method. Given the reduced cost of mesh generation and smaller size of the resulting systems, the BEM is an efficient method for solving engineering problems, especially for exterior problems and problems with free surfaces or moving boundaries.

Two major difficulties limit the application of the boundary element method. First BEM usually generates fully populated system matrices which are computationally expensive to form and solve. This offsets some of the computational time saved by the greatly reduced matrix size. The second major difficulty limiting application of the BEM is the presence of volume integrals in boundary integral equations for nonlinear problems. A common approach for treating the volume integrals is to perform a volume discretization and apply some quadrature formula rules. Unless the nonlinearity exists only in a small region of the whole domain, such approaches are not efficient because they not only destroy the notion of the BEM as a purely boundary solution technique, but they are also computationally expensive.

The primary objective of this project is to tackle these two challenges for boundary element method and to develop fast and robust solutions combining precorrected-Fast Fourier Transform (precorrected-FFT) technique and the boundary element method for solving large-scale problems with complex three-dimensional geometry. A precorrected-FFT accelerated BEM solver for a linear problem, Stokes flow with slip boundary conditions, is developed first as an extension from a no slip Stokes solver. Then a general

yet efficient approach for volume integration is presented to solve non-homogeneous and nonlinear problems. This approach can be used on multiply-connected complex domains so it has the potential to be applied to a wide variety of MEM structures. The performances of the developed 3-D solvers are verified through several examples.

In the following sections, a brief introduction to the BEM and the precorrected-FFT technique is given.

1.2 THE BOUNDARY ELEMENT METHOD

The most widely used numerical methods tackle the differential equations directly in the form in which they are derived, without any further mathematical manipulation. These methods work either by approximating the differential operators in the equations using simpler, localized algebraic ones valid at a series of nodes within the region as in the finite difference method, or by representing the region itself using finite elements of material which are assembled to provide an approximation to the real system as in the finite element method.

An obvious alternative approach to the above methods would be to attempt to integrate the differential equations analytically in some way before either proceeding to any discretization scheme or introducing any approximations. The essence of boundary element methods is the transformation of the differential equations into equivalent sets of integral ones as the first step in their solution.

There are six steps involved in the implementation for boundary element methods:

(1) Problem modeling and statement of the fundamental solutions: the Green's functions.

(2) Derivation of the boundary integral equations from the governing equations.

(3) Discretization of the surface integrals.

(4) Integration of the integrals and formation of the system matrices.

(5) Assembly and solution of the system of equations for the prescribed boundary values.

(6) Postprocessing: Back-substitution of the boundary solutions to obtain the exterior and interior results.

The 4^{th} step, integration of the integrals, is probably the most crucial and time consuming step in the numerical implementation of BEM, and it is much more involved than the finite element method. The problem lies in the fact that the Green's functions which have to be integrated exhibit singularities at certain points. Special treatment is needed for these singular or near singular integrals as will be seen in the following Chapters.

The BEM usually generates fully populated system matrices which are computationally expensive to form and solve. Solving them requires $O(n^3)$ operations (n is the system size) if direct methods are employed or $O(n^2)$ operations if iterative methods are used. Recent development of matrix sparsification techniques such as the Fast Multipole Method (FMM) [12][13][32] and the precorrected-FFT technique [6][31] has reduced the computational complexity to $O(n\log n)$ or $O(n)$. The system matrix is not formed explicitly in these fast methods in order to reduce memory and CPU time. Therefore solvers that compute forward matrix-vector products are required. Iterative solvers like GMRES [34] are a natural choice for these solvers. In other words, the matrix-vector products required by GMRES are computed only approximately. The

combination of Krylov-subspace method based iterative solvers and the matrix sparsification techniques have been used to create very fast boundary element codes. Applications of such techniques include fast solvers using the BEM coupled with fast multipole techniques for potential problem [22][32] and Stokes flow problems [11][19]. The basic idea in developing fast techniques is to approximate the matrix-vector product by some computationally efficient means. In the FMM, multipole and local expansions are used to calculate the long-range interactions, which are also called far-field interactions, with the amount of work being of the order $O(n)$ which depends on the various operators used to speed up the computation. In the precorrected-FFT method, a uniform grid is used to represent the long-range interactions. This grid representation allows the fast Fourier transform (FFT) technique to be used to efficiently perform the computation at a cost of $O(n\log n)$ where n is the number of grid points. A detailed description of this method is presented in next section. One advantage of the precorrected-FFT method is that it can handle a much broader class of kernels than the FMM [30]. Fast solvers based on the precorrected-FFT technique have been developed for Laplace equation, Helmholtz equation [30], scattering problems [6], Stokes equation [1][43][44], etc. These solvers have also been applied successfully in solving practical problems, for example, in the design of Very Large Scale Integration (VLSI) circuits [36] and in the modeling of complex MEMS devices [45].

1.3 PRECORRECTED-FFT TECHNIQUE

The key idea of the precorrected-FFT technique is the following. At each iteration inside GMRES, the most time-consuming step is the calculation of the matrix-vector

product, $\tilde{\mathbf{b}} = \mathbf{P}\mathbf{x}$, where $\mathbf{P}$ is the system matrix and $\mathbf{x} = (\mathbf{u}, \mathbf{f})^T$. A standard procedure for a matrix-vector product requires $O(n^2)$ for a $n \times n$ system. By exploring the fact that when the evaluation panel (i.e., the panel at which $\tilde{\mathbf{b}}$ is evaluated) is far away from the source panel (i.e., the panel at which $\mathbf{x}$ is applied), the source distribution ($\mathbf{x}$) of the panel can be represented using a small number of weighted point sources. If the point sources all lie on a uniform grid, then the computation of the matrix-vector product (i.e., the interaction) at the grid points due to grid point sources is a discrete convolution, which can be performed using the Fast Fourier Transformation (FFT).

First a parallelepiped is constructed to enclose a three-dimensional problem after it has been discretized into n surface panels. This parallelepiped is then subdivided into a $k \times l \times m$ array of small cubes so that each small cube contains only a few panels. Figure 1(a) shows a discretized sphere, with the associated space subdivided into a $3 \times 3 \times 3$ array of cubes. Regard the matrix-vector product ($\mathbf{A}\mathbf{x}$) obtained from the discretization of a boundary integral equation as pseudo potentials evaluated at the evaluation panels due to the pseudo charges ($\mathbf{x}$) distributed on the source panels. Here it is limited to cases when the kernels in the boundary integral equations have piecewise-smooth convolutional form. Potentials at evaluation panels distant from a cube can be accurately computed by representing the given cube's charge distribution using a small number of cube vertices, and then the computation of the potential at the grid points due to the grid charges is a discrete convolution which can be performed using the FFT. A possible set of grid charges for the cube subdivisions is also shown in Figure 1 (b).

Four steps are involved in the procedure for the precorrected-FFT method for approximating $\mathbf{A}\mathbf{x}$ (illustrated in Figure 2):

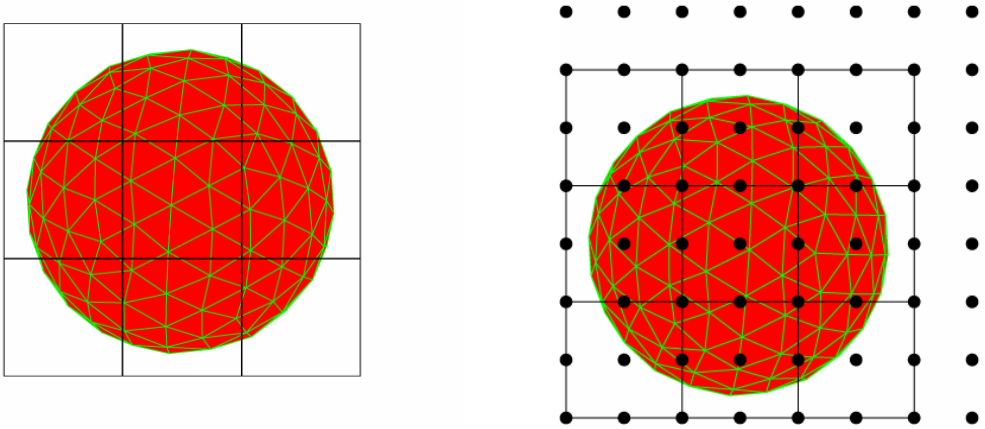


Figure 1. (a) Side view of a sphere discretized into 320 panels, with spatial decomposition into a 3×3×3 array of cubes. (b) Superimposed grid charges corresponding to the cube decomposition of (a). In each cell, a 3×3×3 array of grid charges is used to represent the long-range potential of the charged panels in the cell. Some of the grid charges are shared among cells. Note that the grid is "coarser" than the triangular panels used to discretize the sphere. The grid extends outside the problem domain because the number of grid points is required to be a factor of two [31].

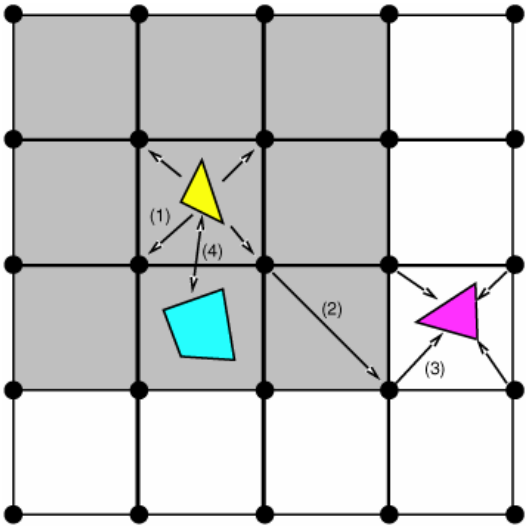


Figure 2. A 2-D pictorial representation of the four steps of the precorrected-FFT method. Interactions with nearby panels (in the grey area) are computed directly, interactions between distant panels are computed using the grid [31].

(1) Project the source vector (**x**) onto the uniform grid points that surround the source panel,

(2) Compute the matrix-vector product (**Ax**) on the grid points due to the projected sources at grid points using the FFT,

(3) Interpolate the products at the grid points onto the evaluation panel,

(4) For nearby panels, the matrix-vector product is computed directly.

The implementation of the first three steps follows the methods described in [1][42].

1.3.1 Projection and interpolation

Projection and interpolation are accomplished using transposed polynomial interpolation scheme as opposed to the grid-potential-collocation approach [30]. The transposed polynomial interpolation is more efficient than the collocation method. Also it leads to a more stable acceleration scheme [42]. The idea of the transposed polynomial interpolation is quite straightforward. Consider the interpolation operator used to interpolate the grid-point potentials to a panel that is surrounded by these points. Take the transpose of the interpolation operator and employ it to project the panel charges onto the surrounding grid points. The accuracy of the projection depends on the order of the polynomial interpolation (p).

1.3.2 Precorrection

The difficulty with steps (1) to (3) is that although the calculations using the FFT on the grid do not accurately approximate the nearby interactions, the portions of the close interactions have already been poorly approximated in the projection/interpolation. A

more accurate calculation of interactions between nearby panels is needed, but it is also necessary to remove or avoid the inaccurate contribution from the use of the gird. An efficient way to correct the poor approximation is to modify the way nearby interactions are computed, i.e., the approximation for close interactions is taken out from the direct matrix first before the computation of the nearby interactions. This is where the term 'precorrect' comes from.

1.3.3 Nearby interactions

For nearby interactions, the precorrected-FFT method cannot be used any more to ensure accuracy. Because of the singular integrands, an accurate integration scheme must be employed to compute the integrals, particularly when the evaluation point is on or near the source panel [39]. The matrix-vector product is then computed directly by forming the matrix entries explicitly and multiplying them with the source. For a flat panel, weakly singular integrals with $1/r$ singularity (r is the distance between the evaluation point and source point) in surface integrals and $1/r^2$ singularity in volume integrals can be converted into nonsingular integrals through a transformation when the evaluation point is on the source panel [42][43]. In non-singular cases, i.e., the evaluation point is not on the source panel, an analytic approach can be employed to evaluate integrals with $1/r$ kernel and its derivatives [42][43].

1.3.4 Complexity

In [30], a detailed complexity analysis of the precorrected-FFT method is given. For problems with field sources relatively uniformly spaced, the method is $O(n\log n)$ nearly

independent of the kernel. For problems which are very inhomogeneous, that is, which have regions of densely concentrated field sources interspersed with large regions of empty space, the method slows to order $n^{6/5}$ for $1/r$ kernels and order $n^{4/3}$ for Helmholtz kernels.

1.4 THESIS ORGANIZATION

The rest of the dissertation is divided into two chapters corresponding to the two objectives of the project.

Chapter 2 describes an efficient numerical method for modeling oscillatory incompressible slip Stokes flows in three dimensions. First the integral representations for slip flows with two different slip models are formulated. The resulting integral equations are then solved using the boundary element method combined with the precorrected-FFT acceleration technique. 3-D numerical codes have been developed and used to calculate the drag forces on oscillating objects immersed in an unbounded slip flow. Three objects are considered, namely a sphere, a pair of plates and a comb structure. The simulated drag forces on these objects obtained from the two slip models are presented. In the sphere case, the simulated results are also compared with the analytical solutions for both the steady state case and the no-slip oscillatory case. In addition, qualitative comparison of the simulation results with the experimental results in the plate problem is also presented.

In Chapter 3, an efficient approach that only requires surface discretization as its input is developed to evaluate the volume integrals that arise in the boundary integral formulations of non-homogeneous and nonlinear problems with 3-D complex geometry.

The volume integral that contains the nonlinear term is evaluated directly with the aid of the 3-D uniform grid constructed initially for the acceleration of surface integration. This grid is also used to accelerate volume integration to further reduce the computational cost. The approach for volume integration is described first, followed by the issues in implementation. After verification of the volume integration approach using numerical studies, the formulation for nonlinear problem is given. Several examples are then used to test the developed 3-D accelerated BEM solvers for non-homogeneous problems and nonlinear problems. Results from the simulation on two domains, a solid sphere and a solid ellipsoid with two spherical exclusions, are presented and compared with analytical solutions.

CHAPTER 2

A FAST INTEGRAL APPROACH FOR DRAG FORCE CALCULATION DUE TO OSCILLATORY SLIP STOKES FLOW

2.1 INTRODUCTION

Over the last decade, there has been great interest in the applications of micron scale mechanical devices in such diverse areas as instrumentation, microelectronics, bioengineering, and advanced energy microsystems [26]. In the beginning, much emphasis was in developing efficient fabrication techniques for microdevices. However, the technology is advancing at a rate that far exceeds that of our understanding of the unconventional physics involved in the operation. Inherent with these new technologies is the need and requirement for better understanding and modeling of these microdevices. This chapter focuses on the modeling of rarefied gas flows that are encountered in microelectromechanical systems (MEMS). The interaction between flows and devices, in particular the drag force exerting on the devices by the flows, is crucial in determining the dynamic behavior of the devices.

There are basically two ways of modeling a flow field: either as the fluid really is, a collection of molecules, or as a continuum where the matter is assumed continuous and indefinitely divisible. The former modeling is subdivided into deterministic methods and probabilistic ones, while in the latter approach the velocity, density, pressure, etc., are defined at every point in space and time, and conservation of mass, energy and momentum lead to a set of nonlinear partial differential equations (Euler, Navier-Stokes, Burnett, etc.).

The continuum model, embodied in the Navier-Stokes equations, is applicable to numerous flow situations. The model ignores the molecular nature of gases and liquids and regards the fluid as a continuous medium describable in terms of the spatial and temporal variations of density, velocity, pressure, temperature and other macroscopic flow quantities. Basically, the continuum model leads to fairly accurate predictions as long as local properties such as density and velocity can be defined as averages over elements, which are large comparing with the microscopic structure of the fluid but small enough in comparison with the scale of the macroscopic phenomena, to permit the use of differential calculus to describe them. Additionally, the flow must not be too far from thermodynamic equilibrium. The former condition is almost always satisfied.

An important parameter of gas flows is the Knudsen number (Kn) which is defined as the ratio of the mean free path of the gas molecule to the characteristic length of the flow ($Kn = \lambda/L$). For gases, the mean free path λ is the average distance traveled by molecules between collisions. For an ideal gas modeled as rigid spheres, the mean free path is related to temperature T and pressure p as follows,

$$\lambda = \frac{1}{\sqrt{2}\pi n\sigma^2} = \frac{kT}{\sqrt{2}\pi p\sigma^2}$$

where n is the number density (number of molecules per unit volume), σ is the molecular diameter, and k is the Boltzmann constant.

The local value of Knudsen number determines the degree of rarefaction and the degree of validity of the continuum model [9]. In the limit of zero Knudsen number, the transport terms in the continuum momentum and energy equations are negligible and the Navier-Stokes equations then reduce to the inviscid Euler equations. As Kn increases, rarefaction effects become more important, and eventually the continuum approach

breaks down altogether. The different Knudsen number regimes can be summarized as follows [35],

Navier-Stokes equations with no-slip boundary conditions $Kn \leq 10^{-3}$

Navier-Stokes equations with slip boundary conditions $10^{-3} \leq Kn \leq 10^{-1}$

Transition regime $10^{-1} \leq Kn \leq 10$

Free-molecule flow $Kn > 10$

It should be noted that the different Knudsen number regimes are determined empirically and are therefore only approximate for particular flow geometry.

For most microsystems, the minimum feature sizes are on the order of 1 μm. Flows inside these systems fall into the slip regime, that is, the regime at which the Knudsen number of the flow is in the range of 0.001 and 0.1. For example, at the standard conditions, i.e., room temperature and 1 atmosphere pressure, the Knudsen number of air inside a microchannel with a width of 1 μm is around 0.06 (λ=65nm), indicating that air in such a condition is in the slip regime.

In the slip-flow regime, Navier-Stokes equations are still valid. However, the classical no-slip boundary condition is no longer valid at a fluid-structure interface. A certain degree of slip exists between the tangential velocities of the flow and that of the solid surface (the wall). The slip amount depends on the accommodation coefficient of the surface (defined as the fraction of molecules reflected diffusively), the mean free path of the molecules, and the normal derivatives of the tangential velocities at the interface. A simple slip model that accounts for the first-order effects was derived by Maxwell in 1879 [21]. In a non-dimensional form, the slip velocity at an isothermal wall is given as

$$U_g - U_w = \frac{2-\sigma_v}{\sigma_v}(Kn)\frac{\partial U_g}{\partial n}, \tag{2.1}$$

where U_g, U_w are normalized tangential velocities of the flow and the wall at the interface respectively, $\partial U_g/\partial n$ is the normal derivative of the normalized tangential velocity at the interface and σ_v is the tangential momentum accommodation coefficient, which is defined as the fraction of molecules reflected diffusively [21] :

$$\sigma_v = \frac{\tau_r - \tau_i}{\tau_r - \tau_w}$$

here the subscripts *i, r, w* stand for incident, reflected and solid wall conditions respectively, and τ is a tangential momentum flux. For example, σ_v = 0.0 corresponds to specular reflection, and σ_v = 1.0 corresponds to diffuse reflection. In the first case, the tangential velocity of the molecules reflected from the walls is unchanged and the normal velocity of the molecules is reversed due to the normal momentum transfer to the wall. In the second case, the molecules are reflected from the walls with zero average tangential velocity. The diffuse reflection, in particular, is an important phenomenon for tangential momentum exchange (and thus friction) of the gas with the walls. In Equation (2.1), slip is assumed even for the diffuse reflection case. Therefore, the no-slip condition on the walls is obtained only when *Kn* = 0.0. The accommodation coefficient depends on the fluid, the solid and the surface finish, and has been determined experimentally to be between 0.2-0.8 [9], the lower limit being for exceptionally smooth surfaces while the upper limit is typical for most practical surfaces.

Another more important dimensionless parameter in fluid mechanics is the Reynolds number, the factor that determines which type of flow is present, expressed as the ratio of inertia forces to viscous forces within the fluid,

$$\mathrm{Re} = \frac{UL}{\nu}$$

where U is a characteristic velocity, and ν is the kinematic viscosity of the fluid.

It is common for flows encountered in MEMS to have small Reynold numbers due to small spatial scales of the devices. This fact implies that the nonlinear convective term in the Navier-Stokes equations is negligible and thus the governing equations can be reduced to Stokes equations. Solving even linear Stokes equations with slip boundary conditions (for example, the one shown in Equation (2.1)), is a challenging problem because of the requirement for high accuracy in computing the normal derivatives of the velocities near the wall. This problem is further complicated by the complex three-dimensional flow domain. Current methods are based largely on domain methods such as finite difference, finite element and spectral methods. These methods require volume discretization of the whole flow domain. In a problem with a complex three-dimensional domain and/or unbounded domain, this requirement could result in a prohibitively large computational cost. For this reason, most work that has been reported in this area is limited to either 2-D cases or to structures with simple geometries [4][5][17][41]. In this work, we propose an efficient boundary integral approach for solving Stokes equations with slip boundary conditions. In this method, integral formulations of Stokes equations with slip boundary conditions are derived first. The resulting integral equations are then solved using the Boundary Element Method [3] combined with the precorrected-FFT acceleration technique [30][31]. This approach requires only surface discretization of the flow domain — greatly reducing the complexity of the meshing process and the size of the resulting system. With the precorrected-FFT technique, the dense linear system resulting from the discretization of the integral equations can be solved very efficiently.

The total computational cost of this approach is on the order of $n\log n$ for most cases, where n is on the order of the size of the linear system [30]. In addition to the simplicity in discretization, the normal derivatives of the velocity can be computed accurately near the wall, and high-order slip effects can be easily included in this approach.

In this chapter, the governing equations, velocity-slip boundary condition models and the corresponding boundary integral formulations are presented first. Then we describe numerical methods, implementation and discuss some critical issues related to the implementation. Results from three case studies, namely a sphere, a pair of plates and a micro comb structure moving in an unbounded slip flow are presented next. Comparison of the simulated drag forces with analytical solutions (if any exist) and/or experimental results are given.

2.2 Governing Equations, Slip Models and the Boundary Integral Formulations

2.2.1 Governing equations

In the slip flow regime, when the Reynolds number and the compressibility effects are small, flows are governed by the incompressible Stokes equations with velocity-slip boundary conditions (isothermal condition is assumed). The incompressible Stokes equations for oscillatory flows are [33]

$$\begin{aligned} &(\nabla^2 - i\omega\frac{\rho}{\mu})\mathbf{u} = \frac{1}{\mu}\nabla P - \frac{\rho}{\mu}\mathbf{b}\,, \\ &\nabla \bullet \mathbf{u} = 0\,, \end{aligned} \tag{2.2}$$

where P , $\mathbf{b}$, $\mathbf{u}$ are the amplitudes of the pressure, the body force and the velocity of the fluid, ω is the oscillating frequency and ρ, μ are the density and viscosity of the fluid

respectively. In this study, the body force is neglected.

2.2.2 Slip models

Two slip models are considered in this study, which will be referred to as models I and II. Both models are obtained by an approximate analysis of the motion of a monatomic gas near an isothermal surface [5]. In this analysis, it is assumed that approximately half of the molecules are coming from the layer of gas one mean free path (λ) away from the surface, while the other half of the molecules are reflected from the wall (Equation (2.3)). Furthermore, σ_v of the molecules are assumed to be reflected "diffusively" from the wall (with average tangential velocity that of the wall) and $(1-\sigma_v)$ of the molecules are reflected from the wall specularly (i.e., conserving their average incoming tangential velocity).

$$U_g = \frac{1}{2}\left[U_\lambda + (1-\sigma_v)U_\lambda + \sigma_v U_w\right] \tag{2.3}$$

Here, U_λ is the velocity of flow molecules located about one mean free path (λ) away from the interface. Following [4], replacing λ with Kn in nondimensional form and expanding U_λ in the above equation in terms of U_g using Taylor series expansion, we have

$$\begin{aligned} U_g = {} & \frac{1}{2}\left[U_g + (Kn)\left(\frac{\partial U}{\partial n}\right)_g + \frac{(Kn)^2}{2}\left(\frac{\partial^2 U}{\partial n^2}\right)_g + \cdots \right] \\ & + \frac{1}{2}\left\{ (1-\sigma_v)\left[U_g + (Kn)\left(\frac{\partial U}{\partial n}\right)_g + \frac{(Kn)^2}{2}\left(\frac{\partial^2 U}{\partial n^2}\right)_g + \cdots \right] + \sigma_v U_w \right\} \end{aligned} \tag{2.4}$$

where n indicates the normal direction of the wall. By rearranging Equation (2.4), we can get the slip velocity for isothermal surfaces in dimensionless form as

$$U_g - U_w = \frac{2-\sigma_v}{\sigma_v}\left[(Kn)\frac{\partial U_g}{\partial n} + \frac{(Kn)^2}{2!}\frac{\partial^2 U_g}{\partial n^2} + \frac{(Kn)^3}{3!}\frac{\partial^3 U_g}{\partial n^3} + \cdots\right]. \tag{2.5}$$

To implement the above slip condition directly using domain methods is rather difficult because the second-order and higher-order derivatives of velocity cannot be computed accurately near the wall [9].

Then a general form of slip boundary condition is introduced [4],

$$U_g - U_w = \frac{2-\sigma}{\sigma}\frac{(Kn)}{1 - B(Kn)\cdot(Kn)}\frac{\partial U}{\partial n} \tag{2.6}$$

where $B(Kn)$ is a parameter to be determined. For a general choice of $B(Kn)$, Equation (2.6) is first-order accurate in Kn. The value of $B(Kn)$ as $Kn \to 0.0$ is used to make Equation (2.6) second-order accurate in Kn for finite but small values of Kn. And it is obvious from Equation (2.6) that $B(Kn) \le 0$. Then we can determine the value of $B(Kn)$ for small Kn corresponding to the slip-flow regime by Taylor series expension of $B(Kn)$ about $Kn = 0$,

$$B(Kn) = B\big|_0 + \left.\frac{dB}{dKn}\right|_0 \cdot (Kn) + \cdots = b + c\cdot(Kn) + \cdots \tag{2.7}$$

Assuming that $B(Kn)\cdot(Kn) < 1$, expanding Equation (2.6) with a geometric series incorporating Equation (2.7) for $B(Kn)$ results in,

$$U_g - U_w = \frac{2-\sigma}{\sigma}\cdot(Kn)\cdot\frac{\partial U}{\partial n}[1 + b\cdot(Kn) + (b^2 + c^2)\cdot(Kn)^2 + \cdots] \tag{2.8}$$

A high-order slip boundary condition can be obtained by comparing the right-hand-side of Equation (2.5) with Equation (2.8) and matching the second-order terms. The second-order slip coefficient b can be determined as,

$$b = (U_0''/2U_0')\big|_w \tag{2.9}$$

where U_0' and U_0'' denote the first and second derivatives of tangential component of the velocity vector along the normal direction to the surface and corresponds to a continuum (no-slip) solution. Equation (2.9) includes second-order accuracy in Knudsen number. This boundary condition only requires the computation of higher-order derivatives of the continuum field, so it is easier to implement. Now the second-order slip boundary condition, our Model I hereafter, can be stated as,

Model I: $$U_g - U_w = \frac{2-\sigma_v}{\sigma_v}\frac{(Kn)}{1-b\cdot(Kn)}\frac{\partial U_g}{\partial n}. \qquad (2.10)$$

In Model I, *b* is the high-order slip coefficient. By choosing a proper value of *b*, high-order slip effects can be included in this model. For example, if *b* is zero, Equation (2.6) recovers the Maxwell model shown in Equation (2.1). When *b* is in the form of Equation (2.9), it includes the second-order slip effect. Using a similar method, we can also obtain higher order slip coefficients such as third-order or fourth-order etc. and thus obtain higher order slip boundary conditions.

Model I (shown in Equation (2.10)) has the advantage of including the high-order slip effects without the need for higher-order derivatives of slip velocity in the model. However, it has been pointed out that in a time-dependent problem, the overall Navier-Stokes solution obtained from a domain method, with explicitly implemented velocity-slip boundary conditions in Equation (2.10), becomes unstable when the calculated slip amount ($U_g - U_w$) at a certain time step is sufficiently large to cause a sudden change of the sign of wall vorticity in the next time step. Thus, in order to avoid sign changes in the vorticity at the wall, a significant restriction must be imposed on the spatial step size,

especially for spectral-based methods. To avoid this difficulty, the starting point to derive Model I is used as our Model II.

Model II: $$U_g = \frac{1}{2}\left[U_\lambda + (1-\sigma_v)U_\lambda + \sigma_v U_w\right]. \tag{2.11}$$

2.2.3 The boundary integral formulations

A direct integral formulation of the oscillatory Stokes equations (Equation (2.2)) is given in [33] as

$$c(\mathbf{x})u_q(\mathbf{x}) = -\frac{1}{\mu}\int_D G_{qp}(\mathbf{x},\mathbf{y})f_p(\mathbf{y})ds(\mathbf{y}) + \int_D u_p(\mathbf{y})T_{pqk}(\mathbf{x},\mathbf{y})n_k(\mathbf{y})ds(\mathbf{y}) \tag{2.12}$$

with

$$c(\mathbf{x}) = \begin{cases} 8\pi, & \mathbf{x} \in V \\ 4\pi, & \mathbf{x} \in D \\ 0, & \mathbf{x} \notin (D \cup V) \end{cases}.$$

Here D is the boundary of the fluid domain V (a simply- or multiply-connected, piecewise Lyapunov surface), $u_q(\mathbf{x})$ is the q^{th} component of the amplitude of the velocity at the evaluation point $\mathbf{x}$, $f_p(\mathbf{y})$ is the p^{th} component of the amplitude of the traction at a boundary point $\mathbf{y}$ (also called the field point) (For simplicity in presentation, **u** and **f** are referred to as velocity and traction in the rest of this paper), $n_k(\mathbf{y})$ is the k^{th} component of the outward normal vector at the boundary point $\mathbf{y}$ (pointing toward the fluid domain), p,q,k = 1,2,3, and G, T are the free space Green's functions of oscillatory Stokes flows defined as follows:

$$G_{qp} = A(R)\frac{\delta_{pq}}{r} + B(R)\frac{\hat{x}_q\hat{x}_p}{r^3}, \qquad \text{with } \hat{\mathbf{x}} = \mathbf{y}\text{-}\mathbf{x},\ r = |\hat{\mathbf{x}}|, \text{ and } R = r\sqrt{i\omega L^2\frac{\rho}{\mu}},$$
$$T_{pqk} = -\frac{2}{r^3}\left(\delta_{pq}\hat{x}_k + \delta_{kq}\hat{x}_p\right)\left[e^{-R}(R+1) - B\right] - \frac{2}{r^3}\delta_{pk}\hat{x}_q(1-B) - 2\frac{\hat{x}_p\hat{x}_q\hat{x}_k}{r^5}\left[5B - 2e^{-R}(R+1)\right], \tag{2.13}$$

where L is the characteristic length and

$$A = 2e^{-R}\left(1+\frac{1}{R}+\frac{1}{R^2}\right) - \frac{2}{R^2}, \qquad B = -2e^{-R}\left(1+\frac{3}{R}+\frac{3}{R^2}\right) + \frac{6}{R^2}.$$

In the case of zero frequency (steady flow), both A and B are equal to 1 and Green's functions of Stokes flows reduce to

$$G_{qp} = \frac{\delta_{pq}}{r} + \frac{\hat{x}_q\hat{x}_p}{r^3}, \quad T_{pqk} = -6\frac{\hat{x}_p\hat{x}_q\hat{x}_k}{r^5}, \qquad \text{with } \hat{\mathbf{x}} = \mathbf{y} - \mathbf{x} \text{ and } r = |\hat{\mathbf{x}}|. \tag{2.14}$$

2.2.3.1. Boundary integral equation for Model I

To derive the boundary integral formulation for Model I (Equation (2.10)), an integral representation of the normal derivatives of the velocity at the boundary points is required.

(1) Steady Case:

Consider the integral formulation of the velocity at a point **x** located inside the fluid domain (shown in Figure 3)

$$u_q(\mathbf{x}) = -\frac{1}{8\pi\mu}\int_D G_{qp}(\mathbf{x},\mathbf{y}) f_p(\mathbf{y}) ds(\mathbf{y}) + \frac{1}{8\pi}\int_D u_p(\mathbf{y}) T_{pqk}(\mathbf{x},\mathbf{y}) n_k(\mathbf{y}) ds(\mathbf{y}). \tag{2.15}$$

Taking the derivative of $u_q(\mathbf{x})$ with respect to x_m, one obtains

$$\frac{\partial u_q}{\partial x_m}(\mathbf{x}) = -\frac{1}{8\pi\mu}\int_D \frac{\partial G_{qp}(\mathbf{x},\mathbf{y})}{\partial x_m} f_p(\mathbf{y}) ds(\mathbf{y}) + \frac{1}{8\pi}\int_D u_p(\mathbf{y}) \frac{\partial T_{pqk}(\mathbf{x},\mathbf{y})}{\partial x_m} n_k(\mathbf{y}) ds(\mathbf{y}), \quad \mathbf{x} \in V. \tag{2.16}$$

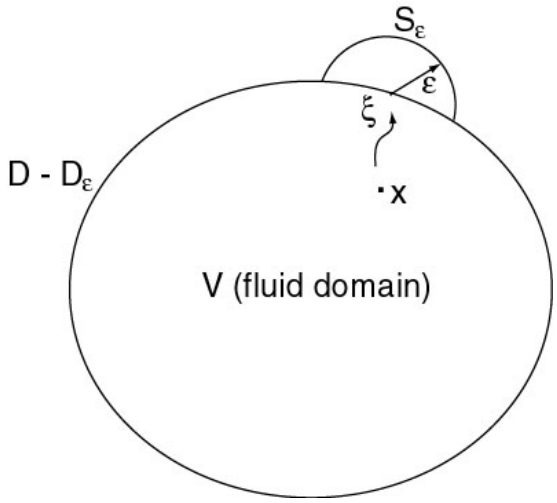


Figure 3. Illustration of the fluid domain and its boundary.

The integral formulation for normal derivatives of velocity at a boundary point ξ which is locally smooth (i.e., the normal vector at ξ is uniquely defined) can be obtained by first letting **x** approach to ξ in Equation (2.16) and then multiplying the resulting equation with $n_m(\xi)$ as shown in Equation (2.17),

$$\frac{\partial u_q}{\partial n}(\boldsymbol{\xi}) = n_m(\boldsymbol{\xi}) \cdot \lim_{\mathbf{x}\to\xi} \left\{ -\frac{1}{8\pi\mu} \int_D \frac{\partial G_{qp}(\mathbf{x},\mathbf{y})}{\partial x_m} f_p(\mathbf{y}) ds(\mathbf{y}) + \frac{1}{8\pi} \int_D u_p(\mathbf{y}) \frac{\partial T_{pqk}(\mathbf{x},\mathbf{y})}{\partial x_m} n_k(\mathbf{y}) ds(\mathbf{y}) \right\}. \tag{2.17}$$

However, because of the singularities of the kernels ($\frac{\partial G}{\partial x} \approx O(r^{-2}), \frac{\partial T}{\partial x} \approx O(r^{-3})$), the integrands on the right-hand side of Equation (2.17) are singular when ξ coincides with **y**. To regularize the above singular integral equation, we first note that

$$\int_D T_{pqk}(\mathbf{x},\mathbf{y}) n_k(\mathbf{y}) ds = 8\pi\delta_{pq} \text{ , for } \mathbf{x} \in V \text{ [33].}$$

Thus, we have

$$\int_D \frac{\partial T_{pqk}(\mathbf{x},\mathbf{y})}{\partial x_m} n_k(\mathbf{y}) ds = 0 . \tag{2.18}$$

So Equation (2.17) is equivalent to

$$\frac{\partial u_q}{\partial n}(\boldsymbol{\xi}) = n_m(\boldsymbol{\xi}) \cdot \lim_{\mathbf{x}\to\boldsymbol{\xi}} \left\{ -\frac{1}{8\pi\mu} \int_D \frac{\partial G_{qp}(\mathbf{x},\mathbf{y})}{\partial x_m} f_p(\mathbf{y}) ds(\mathbf{y}) + \frac{1}{8\pi} \int_D \left(u_p(\mathbf{y}) - u_p(\mathbf{x}) \right) \frac{\partial T_{pqk}(\mathbf{x},\mathbf{y})}{\partial x_m} n_k(\mathbf{y}) ds(\mathbf{y}) \right\} \tag{2.19}$$

Now consider a sphere $V_\varepsilon(\xi)$ with radius of ε and centered at point ξ as indicated in Figure 3. S_ε is the "half sphere", i.e., portion of sphere $V_\varepsilon(\xi)$ which is outside of V. $D_\varepsilon = D \cap V_\varepsilon(\boldsymbol{\xi})$. By Equation (2.12), when V is replaced by $V_\varepsilon(\xi) \setminus V$, we have

$$-\frac{1}{8\pi\mu} \int_{D_\varepsilon} \frac{\partial G_{qp}}{\partial x_m} f_p ds + \frac{1}{8\pi} \int_{D_\varepsilon} u_p \frac{\partial T_{pqk}}{\partial x_m} n_k ds = -\frac{1}{8\pi\mu} \int_{S_\varepsilon} \frac{\partial G_{qp}}{\partial x_m} f_p ds + \frac{1}{8\pi} \int_{S_\varepsilon} u_p \frac{\partial T_{pqk}}{\partial x_m} n_k ds$$

because the evaluation point **x** is outside the domain. Thus

$$\begin{aligned} &-\frac{1}{8\pi\mu} \int_D \frac{\partial G_{qp}(\mathbf{x},\mathbf{y})}{\partial x_m} f_p(\mathbf{y}) ds(\mathbf{y}) + \frac{1}{8\pi} \int_D \left(u_p(\mathbf{y}) - u_p(\mathbf{x}) \right) \frac{\partial T_{pqk}(\mathbf{x},\mathbf{y})}{\partial x_m} n_k(\mathbf{y}) ds(\mathbf{y}) \\ &= -\frac{1}{8\pi\mu} \int_{D-D_\varepsilon} \frac{\partial G_{qp}(\mathbf{x},\mathbf{y})}{\partial x_m} f_p(\mathbf{y}) ds(\mathbf{y}) + \frac{1}{8\pi} \int_{D-D_\varepsilon} \left(u_p(\mathbf{y}) - u_p(\mathbf{x}) \right) \frac{\partial T_{pqk}(\mathbf{x},\mathbf{y})}{\partial x_m} n_k(\mathbf{y}) ds(\mathbf{y}) . \\ &\quad -\frac{1}{8\pi\mu} \int_{S_\varepsilon} \frac{\partial G_{qp}(\mathbf{x},\mathbf{y})}{\partial x_m} f_p(\mathbf{y}) ds(\mathbf{y}) + \frac{1}{8\pi} \int_{S_\varepsilon} \left(u_p(\mathbf{y}) - u_p(\mathbf{x}) \right) \frac{\partial T_{pqk}(\mathbf{x},\mathbf{y})}{\partial x_m} n_k(\mathbf{y}) ds(\mathbf{y}) \end{aligned}$$

As $\varepsilon \to 0$, Equation (2.19) is equivalent to

$$\frac{\partial u_q}{\partial n}(\boldsymbol{\xi}) = n_m(\boldsymbol{\xi}) \cdot \left(I_m^1 + I_m^2 \right) \tag{2.20}$$

with

$$I_m^1 = \lim_{\varepsilon\to 0} \left[-\frac{1}{8\pi\mu} \int_{D-D_\varepsilon} \frac{\partial G_{qp}(\boldsymbol{\xi},\mathbf{y})}{\partial \xi_m} f_p(\mathbf{y}) ds(\mathbf{y}) - \frac{1}{8\pi\mu} \int_{S_\varepsilon} \frac{\partial G_{qp}(\boldsymbol{\xi},\mathbf{y})}{\partial \xi_m} f_p(\mathbf{y}) ds(\mathbf{y}) \right]$$

and

$$I_m^2 = \lim_{\varepsilon\to 0} \left[\frac{1}{8\pi} \int_{D-D_\varepsilon} \left(u_p(\mathbf{y}) - u_p(\boldsymbol{\xi}) \right) \frac{\partial T_{pqk}(\boldsymbol{\xi},\mathbf{y})}{\partial \xi_m} n_k(\mathbf{y}) ds(\mathbf{y}) + \frac{1}{8\pi} \int_{S_\varepsilon} \left(u_p(\mathbf{y}) - u_p(\boldsymbol{\xi}) \right) \frac{\partial T_{pqk}(\boldsymbol{\xi},\mathbf{y})}{\partial \xi_m} n_k(\mathbf{y}) ds(\mathbf{y}) \right]$$

With the following observations, it can be shown that

$$\frac{\partial u_q}{\partial n}(\boldsymbol{\xi}) = -\frac{1}{4\pi\mu}\int_D^{cpv} \frac{\partial G_{qp}(\boldsymbol{\xi},\mathbf{y})}{\partial \xi_m} n_m(\boldsymbol{\xi}) f_p(\mathbf{y}) ds(\mathbf{y}) + \frac{1}{4\pi}\int_D^{cpv} \left[u_p(\mathbf{y}) - u_p(\boldsymbol{\xi})\right] \frac{\partial T_{pqk}(\boldsymbol{\xi},\mathbf{y})}{\partial \xi_m} n_m(\boldsymbol{\xi}) n_k(\mathbf{y}) ds(\mathbf{y}). \tag{2.21}$$

where

$$\frac{\partial G_{qp}(\boldsymbol{\xi},\mathbf{y})}{\partial \xi_m} = \frac{\hat{x}_m\delta_{pq} - \hat{x}_q\delta_{pm} - \hat{x}_p\delta_{qm}}{r^3} + 3\frac{\hat{x}_m\hat{x}_q\hat{x}_p}{r^5},$$

$$\frac{\partial T_{pqk}(\boldsymbol{\xi},\mathbf{y})}{\partial \xi_m} = 6\left[\frac{\hat{x}_q\hat{x}_k\delta_{pm} + \hat{x}_p\hat{x}_k\delta_{qm} + \hat{x}_p\hat{x}_q\delta_{km}}{r^5} - 5\frac{\hat{x}_m\hat{x}_q\hat{x}_p\hat{x}_k}{r^7}\right]$$

and $\hat{\mathbf{x}} = \mathbf{y} - \boldsymbol{\xi}$, $r = |\hat{\mathbf{x}}|$. All the integrals in Equation (2.21) exist in the Cauchy principal value (CPV, indicated by the superscript "cpv") sense (For the detailed proof see [7]).

(2) Unsteady Case:

The integral equation for the derivatives of velocity of oscillatory flows takes the same form as it is in Equation (2.21) with the Green' functions being the corresponding oscillatory kernels. To show it, we separate the oscillatory Greens' functions in the integral formulation for velocity into two parts (taking G as an example, extensions to other kernels can be easily done):

$$G = (G - G^S) + G^S. \tag{2.22}$$

Here G^S refers to the corresponding steady kernels. The remaining part, i.e. the terms inside the bracket on the right hand side of Equation (2.30) is continuous. To show this, $G - G^s$ reads

$$G_{qp} - G_{qp}{}^S = \left(A(R) - 1\right)\frac{\delta_{pq}}{r} + \left(B(R) - 1\right)\frac{\hat{x}_q\hat{x}_p}{r^3}.$$

From the Taylor series expansion of $A(R)$ and $B(R)$, it is clear that $(G_{qp}-G_{qp}^{S})$ is no longer singular as $R\to 0$ (i.e. $r\to 0$) because

$$A = 1-\frac{4}{3}R+\frac{3}{4}R^2-\frac{4}{15}R^3+o(R^4) \text{ and } B = 1-\frac{1}{4}R^2+\frac{2}{15}R^3-\frac{1}{24}R^4+o(R^5).$$

Thus, the integral formulation for the derivatives of the velocity of oscillatory flows can be obtained by following the procedure described previously (in the steady case) for the steady part of the kernels and adding the contribution from the remaining part of kernels, which is very easy to derive since this part of the kernels is not singular. The integral formulation reads

$$\frac{\partial u_q}{\partial n}(\mathbf{x}) = -\frac{1}{4\pi\mu}\int_D^{cpv}\frac{\partial G_{qp}(\mathbf{x},\mathbf{y})}{\partial \mathbf{x}_m}n_m(\mathbf{x})f_p(\mathbf{y})ds(\mathbf{y})+\frac{1}{4\pi}\int_D^{cpv}\left[u_p(\mathbf{y})-u_p(\mathbf{x})\right]\frac{\partial T_{pqk}(\mathbf{x},\mathbf{y})}{\partial \mathbf{x}_m}n_m(\mathbf{x})n_k(\mathbf{y})ds(\mathbf{y}), \tag{2.23}$$

where

$$\begin{aligned}
\frac{\partial G_{qp}(\mathbf{x},\mathbf{y})}{\partial x_m} &= \frac{\partial A}{\partial x_m}\frac{\delta_{pq}}{r}+A\frac{\delta_{pq}\hat{x}_m}{r^3}+\frac{\partial B}{\partial x_m}\frac{\hat{x}_p\hat{x}_q}{r^3}+B\left(-\frac{\hat{x}_q\delta_{pm}+\hat{x}_p\delta_{qm}}{r^3}+3\frac{\hat{x}_m\hat{x}_q\hat{x}_p}{r^5}\right),\\
\frac{\partial T_{pqk}(\mathbf{x},\mathbf{y})}{\partial x_m} &= \left(\frac{\partial C}{\partial x_m}-\frac{\partial B}{\partial x_m}\right)\left(-2\frac{\delta_{pq}\hat{x}_k+\delta_{kq}\hat{x}_p}{r^3}\right)+\frac{\partial B}{\partial x_m}2\frac{\delta_{pk}\hat{x}_q}{r^3}-\left(5\frac{\partial B}{\partial x_m}-2\frac{\partial C}{\partial x_m}\right)2\frac{\hat{x}_p\hat{x}_q\hat{x}_k}{r^5}\\
&\quad -2(C-B)\left(3\frac{\hat{x}_k\hat{x}_m\delta_{pq}+\hat{x}_p\hat{x}_m\delta_{kq}}{r^5}-\frac{\delta_{pq}\delta_{km}+\delta_{kq}\delta_{pm}}{r^3}\right)-2(1-B)\left(3\frac{\hat{x}_q\hat{x}_m\delta_{pk}}{r^5}-\frac{\delta_{pk}\delta_{qm}}{r^3}\right)\\
&\quad -2(5B-2C)\left(5\frac{\hat{x}_q\hat{x}_p\hat{x}_k\hat{x}_m}{r^7}-\frac{\hat{x}_q\hat{x}_k\delta_{pm}+\hat{x}_p\hat{x}_k\delta_{qm}+\hat{x}_p\hat{x}_q\delta_{km}}{r^5}\right),
\end{aligned} \tag{2.24}$$

and $C = e^{-R}(R+1)$. Parameters A and B are defined in Equation (2.13).

2.2.3.2. Boundary integral equation for Model II

For Model II (Equation (2.11)), we need the boundary integral equation for the points located inside the flow domain with distance λ (one mean free path) away from the boundary. This integral reads

$$u_q(\mathbf{x}^\lambda) = -\frac{1}{8\pi\mu}\int_D G_{qp}\left(\mathbf{x}^\lambda,\mathbf{y}\right) f_p\left(\mathbf{y}\right) dS\left(\mathbf{y}\right) + \frac{1}{8\pi}\int_D u_p\left(\mathbf{y}\right) T_{pqk}\left(\mathbf{x}^\lambda,\mathbf{y}\right) n_k\left(\mathbf{y}\right) dS\left(\mathbf{y}\right), \quad (2.25)$$

where $\mathbf{x}^\lambda$ represents the evaluation points that are located λ away from the boundary.

The integral Equation (2.23), together with integral Equation (2.12) and the slip boundary condition in Equation (2.10), forms a complete set of Equations that allow us to solve for unknowns **u** and **f** based on slip Model I as shown in Equation (2.26), while Equation (2.12), Equation (2.25) and Equation (2.11) form a complete set of equations that allow us to solve for unknowns **u** and **f** based on slip Model II as shown in Equation (2.27). The subscripts (t and n) indicate the tangential and normal components of the velocity respectively.

$$\begin{cases} u_q(\mathbf{x}) = -\dfrac{1}{4\pi\mu}\displaystyle\int_D G_{qp}(\mathbf{x},\mathbf{y}) f_p(\mathbf{y}) ds(\mathbf{y}) + \frac{1}{4\pi}\int_D^{cpv} u_p(\mathbf{y}) T_{pqk}(\mathbf{x},\mathbf{y}) n_k(\mathbf{y}) ds(\mathbf{y}) \\ \dfrac{\partial u_q}{\partial n}(\mathbf{x}) = -\dfrac{1}{4\pi\mu}\displaystyle\int_D^{cpv} \frac{\partial G_{qp}(\mathbf{x},\mathbf{y})}{\partial n(\mathbf{x})} f_p(\mathbf{y}) ds(\mathbf{y}) + \frac{1}{4\pi}\int_D^{cpv} (u_p(\mathbf{y}) - u_p(\mathbf{x})) \frac{\partial T_{pqk}(\mathbf{x},\mathbf{y})}{\partial n(\mathbf{x})} n_k(\mathbf{y}) ds(\mathbf{y}) \\ u_t - u_t^w = \dfrac{2-\sigma_v}{\sigma_v}\dfrac{(Kn)}{1-b(Kn)}\dfrac{\partial u_t}{\partial n} \\ u_n = u_n^w \end{cases}, \quad (2.26)$$

$$\begin{cases} u_q(\mathbf{x}) = -\dfrac{1}{4\pi\mu}\displaystyle\int_D G_{qp}(\mathbf{x},\mathbf{y}) f_p(\mathbf{y}) ds(\mathbf{y}) + \frac{1}{4\pi}\int_D^{cpv} u_p(\mathbf{y}) T_{pqk}(\mathbf{x},\mathbf{y}) n_k(\mathbf{y}) ds(\mathbf{y}) \\ u_q(\mathbf{x}^\lambda) = -\dfrac{1}{8\pi\mu}\displaystyle\int_D G_{qp}(\mathbf{x}^\lambda,\mathbf{y}) f_p(\mathbf{y}) ds(\mathbf{y}) + \frac{1}{8\pi}\int_D^{cpv} u_p(\mathbf{y}) T_{pqk}(\mathbf{x}^\lambda,\mathbf{y}) n_k(\mathbf{y}) ds(\mathbf{y}) \\ u_t = \dfrac{1}{2}\left[u_t^\lambda + (1-\sigma_v) u_t^\lambda + \sigma_v u_t^w\right] \\ u_n = u_n^w \end{cases}. \quad (2.27)$$

2.3 NUMERICAL METHODS AND IMPLEMENTATION

The integral equations described in last section are solved using a boundary element method [3] combined with the precorrected-FFT acceleration technique [31]. A piece-wise constant collocation scheme is used to discretize the integral equations. Although higher-order elements are better to use, it has been found that the constant collocation scheme works reasonably well in the numerical examples presented in the results section. The surface of the fluid volume is discretized into n small panels (elements). On each panel, the components of the velocity and the traction force are assumed to be constant. A system of equations for the panel unknowns is then derived by insisting that the integral equations be satisfied at each panel centroid. The result is a linear system, which relates the known quantities (wall velocity $\mathbf{u}_w$ in our example) to the unknown quantities (velocity of the flow $\mathbf{u}$ and traction force $\mathbf{f}$ on the structure). Equation (2.28) is the system equations for Model I and Equation (2.29) is for Model II. It should be pointed out that both systems are complex systems, with the actual velocity and traction of the flow being the real parts of the complex velocity and traction.

$$\mathbf{P}\begin{bmatrix}\mathbf{u}\\ \mathbf{f}\end{bmatrix}=\begin{bmatrix}\mathbf{I}+\mathbf{P}^{(1)} & \mathbf{P}^{(2)}\\ \mathbf{I}+\mathbf{Q}\bar{\mathbf{Q}}\mathbf{P}^{(3)} & \mathbf{Q}\bar{\mathbf{Q}}\mathbf{P}^{(4)}\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ \mathbf{f}\end{bmatrix}=\begin{Bmatrix}\mathbf{0}\\ \mathbf{u}_{\mathbf{w}}\end{Bmatrix} \tag{2.28}$$

with

$$\mathbf{P}^{(1)}_{mp}(\mathbf{x})=-\frac{1}{4\pi}\begin{bmatrix}\int_{\Gamma_{(p)}} T_{11k}(\mathbf{x},\mathbf{y})n_k^{(p)}ds(\mathbf{y}) & \int_{\Gamma_{(p)}} T_{21k}(\mathbf{x},\mathbf{y})n_k^{(p)}ds(\mathbf{y}) & \int_{\Gamma_{(p)}} T_{31k}(\mathbf{x},\mathbf{y})n_k^{(p)}ds(\mathbf{y})\\ \int_{\Gamma_{(p)}} T_{12k}(\mathbf{x},\mathbf{y})n_k^{(p)}ds(\mathbf{y}) & \int_{\Gamma_{(p)}} T_{22k}(\mathbf{x},\mathbf{y})n_k^{(p)}ds(\mathbf{y}) & \int_{\Gamma_{(p)}} T_{32k}(\mathbf{x},\mathbf{y})n_k^{(p)}ds(\mathbf{y})\\ \int_{\Gamma_{(p)}} T_{13k}(\mathbf{x},\mathbf{y})n_k^{(p)}ds(\mathbf{y}) & \int_{\Gamma_{(p)}} T_{23k}(\mathbf{x},\mathbf{y})n_k^{(p)}ds(\mathbf{y}) & \int_{\Gamma_{(p)}} T_{33k}(\mathbf{x},\mathbf{y})n_k^{(p)}ds(\mathbf{y})\end{bmatrix},$$

$$\mathbf{P}_{mp}^{(2)}(\mathbf{x})=\frac{1}{4\pi\mu}\begin{bmatrix}\int_{\Gamma_{(p)}} G_{11}(\mathbf{x},\mathbf{y})ds(\mathbf{y}) & \int_{\Gamma_{(p)}} G_{21}(\mathbf{x},\mathbf{y})ds(\mathbf{y}) & \int_{\Gamma_{(p)}} G_{31}(\mathbf{x},\mathbf{y})ds(\mathbf{y}) \\ \int_{\Gamma_{(p)}} G_{12}(\mathbf{x},\mathbf{y})ds(\mathbf{y}) & \int_{\Gamma_{(p)}} G_{22}(\mathbf{x},\mathbf{y})ds(\mathbf{y}) & \int_{\Gamma_{(p)}} G_{32}(\mathbf{x},\mathbf{y})ds(\mathbf{y}) \\ \int_{\Gamma_{(p)}} G_{13}(\mathbf{x},\mathbf{y})ds(\mathbf{y}) & \int_{\Gamma_{(p)}} G_{23}(\mathbf{x},\mathbf{y})ds(\mathbf{y}) & \int_{\Gamma_{(p)}} G_{33}(\mathbf{x},\mathbf{y})ds(\mathbf{y}) \end{bmatrix},$$

$$\mathbf{P}_{mp}^{(3)}(\mathbf{x})=-\frac{c_1}{4\pi}\begin{bmatrix}\int_{\Gamma_{(p)}} \frac{\partial T_{11k}(\mathbf{x},\mathbf{y})}{\partial n^{(m)}} n_k^{(p)} ds(\mathbf{y}) & \int_{\Gamma_{(p)}} \frac{\partial T_{21k}(\mathbf{x},\mathbf{y})}{\partial n^{(m)}} n_k^{(p)} ds(\mathbf{y}) & \int_{\Gamma_{(p)}} \frac{\partial T_{31k}(\mathbf{x},\mathbf{y})}{\partial n^{(m)}} n_k^{(p)} ds(\mathbf{y}) \\ \int_{\Gamma_{(p)}} \frac{\partial T_{12k}(\mathbf{x},\mathbf{y})}{\partial n^{(m)}} n_k^{(p)} ds(\mathbf{y}) & \int_{\Gamma_{(p)}} \frac{\partial T_{22k}(\mathbf{x},\mathbf{y})}{\partial n^{(m)}} n_k^{(p)} ds(\mathbf{y}) & \int_{\Gamma_{(p)}} \frac{\partial T_{32k}(\mathbf{x},\mathbf{y})}{\partial n^{(m)}} n_k^{(p)} ds(\mathbf{y}) \\ \int_{\Gamma_{(p)}} \frac{\partial T_{13k}(\mathbf{x},\mathbf{y})}{\partial n^{(m)}} n_k^{(p)} ds(\mathbf{y}) & \int_{\Gamma_{(p)}} \frac{\partial T_{23k}(\mathbf{x},\mathbf{y})}{\partial n^{(m)}} n_k^{(p)} ds(\mathbf{y}) & \int_{\Gamma_{(p)}} \frac{\partial T_{33k}(\mathbf{x},\mathbf{y})}{\partial n^{(m)}} n_k^{(p)} ds(\mathbf{y}) \end{bmatrix},$$

$$\mathbf{P}_{mp}^{(4)}(\mathbf{x})=\frac{c_1}{4\pi\mu}\begin{bmatrix}\int_{\Gamma_{(p)}} \frac{\partial G_{11}(\mathbf{x},\mathbf{y})}{\partial n^{(m)}} ds(\mathbf{y}) & \int_{\Gamma_{(p)}} \frac{\partial G_{21}(\mathbf{x},\mathbf{y})}{\partial n^{(m)}} ds(\mathbf{y}) & \int_{\Gamma_{(p)}} \frac{\partial G_{31}(\mathbf{x},\mathbf{y})}{\partial n^{(m)}} ds(\mathbf{y}) \\ \int_{\Gamma_{(p)}} \frac{\partial G_{12}(\mathbf{x},\mathbf{y})}{\partial n^{(m)}} ds(\mathbf{y}) & \int_{\Gamma_{(p)}} \frac{\partial G_{22}(\mathbf{x},\mathbf{y})}{\partial n^{(m)}} ds(\mathbf{y}) & \int_{\Gamma_{(p)}} \frac{\partial G_{32}(\mathbf{x},\mathbf{y})}{\partial n^{(m)}} ds(\mathbf{y}) \\ \int_{\Gamma_{(p)}} \frac{\partial G_{13}(\mathbf{x},\mathbf{y})}{\partial n^{(m)}} ds(\mathbf{y}) & \int_{\Gamma_{(p)}} \frac{\partial G_{23}(\mathbf{x},\mathbf{y})}{\partial n^{(m)}} ds(\mathbf{y}) & \int_{\Gamma_{(p)}} \frac{\partial G_{33}(\mathbf{x},\mathbf{y})}{\partial n^{(m)}} ds(\mathbf{y}) \end{bmatrix},$$

$$\mathbf{Q}=\begin{bmatrix}\mathbf{Q^{(1)}} & & \\ & \mathbf{Q^{(2)}} & \\ & & \ddots \\ & & & \mathbf{Q^{(N)}}\end{bmatrix},\ \bar{\mathbf{Q}}=\begin{bmatrix}\bar{\mathbf{Q}}^{\mathbf{(1)}} & & \\ & \bar{\mathbf{Q}}^{\mathbf{(2)}} & \\ & & \ddots \\ & & & \bar{\mathbf{Q}}^{\mathbf{(N)}}\end{bmatrix},$$

where $c_1=\frac{2-\sigma}{\sigma}\frac{Kn}{1-bKn}$, $1\le m,p\le N$, m is the panel index for the evaluation panel, p refers to field panel, $n^{(m)}$ and $n^{(p)}$ are the normal vectors of the evaluation and field panels respectively, $\mathbf{Q}^{(k)}$ is the 3×3 transformation matrix that transforms the global Cartesian coordinates into the local coordinates of the k^{th} field panel which are constructed by the normal and tangential directions of that panel,

$$\overline{\mathbf{Q}}^{(k)} = \begin{bmatrix} \mathbf{Q}^{(k)}(1,:) \\ \mathbf{Q}^{(k)}(2,:) \\ \mathbf{0\,0\,0} \end{bmatrix}.$$

For Model II, the system (resulting from the discretization of the BIE (2.27)) is

$$\mathbf{P}\begin{bmatrix} \mathbf{u} \\ \mathbf{f} \end{bmatrix} = \begin{bmatrix} \mathbf{I}+\mathbf{P}^{(1)} & \mathbf{P}^{(2)} \\ \mathbf{I}+\mathbf{Q}\overline{\mathbf{Q}}\mathbf{P}^{(3)} & \mathbf{Q}\overline{\mathbf{Q}}\mathbf{P}^{(4)} \end{bmatrix}\begin{bmatrix} \mathbf{u} \\ \mathbf{f} \end{bmatrix} = \begin{Bmatrix} \mathbf{0} \\ \mathbf{Q}\overline{\overline{\mathbf{Q}}}\mathbf{u}_{\mathbf{w}} \end{Bmatrix} \tag{2.29}$$

with same $\mathbf{P}^{(1)}$ and $\mathbf{P}^{(2)}$ as that of Model I, but different $\mathbf{P}^{(3)}$ and $\mathbf{P}^{(4)}$ shown below,

$$\mathbf{P}^{(3)}_{mp}(\mathbf{x}^{\lambda}) = -\frac{c_2}{8\pi}\begin{bmatrix} \int_{\Gamma_{(p)}} T_{11k}(\mathbf{x}^{\lambda},\mathbf{y})n_k^{(p)}ds(\mathbf{y}) & \int_{\Gamma_{(p)}} T_{21k}(\mathbf{x}^{\lambda},\mathbf{y})n_k^{(p)}ds(\mathbf{y}) & \int_{\Gamma_{(p)}} T_{31k}(\mathbf{x}^{\lambda},\mathbf{y})n_k^{(p)}ds(\mathbf{y}) \\ \int_{\Gamma_{(p)}} T_{12k}(\mathbf{x}^{\lambda},\mathbf{y})n_k^{(p)}ds(\mathbf{y}) & \int_{\Gamma_{(p)}} T_{22k}(\mathbf{x}^{\lambda},\mathbf{y})n_k^{(p)}ds(\mathbf{y}) & \int_{\Gamma_{(p)}} T_{32k}(\mathbf{x}^{\lambda},\mathbf{y})n_k^{(p)}ds(\mathbf{y}) \\ \int_{\Gamma_{(p)}} T_{13k}(\mathbf{x}^{\lambda},\mathbf{y})n_k^{(p)}ds(\mathbf{y}) & \int_{\Gamma_{(p)}} T_{23k}(\mathbf{x}^{\lambda},\mathbf{y})n_k^{(p)}ds(\mathbf{y}) & \int_{\Gamma_{(p)}} T_{33k}(\mathbf{x}^{\lambda},\mathbf{y})n_k^{(p)}ds(\mathbf{y}) \end{bmatrix},$$

$$\mathbf{P}^{(4)}_{mp}(\mathbf{x}^{\lambda}) = \frac{c_2}{8\pi\mu}\begin{bmatrix} \int_{\Gamma_{(p)}} G_{11}(\mathbf{x}^{\lambda},\mathbf{y})ds(\mathbf{y}) & \int_{\Gamma_{(p)}} G_{21}(\mathbf{x}^{\lambda},\mathbf{y})ds(\mathbf{y}) & \int_{\Gamma_{(p)}} G_{31}(\mathbf{x}^{\lambda},\mathbf{y})ds(\mathbf{y}) \\ \int_{\Gamma_{(p)}} G_{12}(\mathbf{x}^{\lambda},\mathbf{y})ds(\mathbf{y}) & \int_{\Gamma_{(p)}} G_{22}(\mathbf{x}^{\lambda},\mathbf{y})ds(\mathbf{y}) & \int_{\Gamma_{(p)}} G_{32}(\mathbf{x}^{\lambda},\mathbf{y})ds(\mathbf{y}) \\ \int_{\Gamma_{(p)}} G_{13}(\mathbf{x}^{\lambda},\mathbf{y})ds(\mathbf{y}) & \int_{\Gamma_{(p)}} G_{23}(\mathbf{x}^{\lambda},\mathbf{y})ds(\mathbf{y}) & \int_{\Gamma_{(p)}} G_{33}(\mathbf{x}^{\lambda},\mathbf{y})ds(\mathbf{y}) \end{bmatrix},$$

$$\overline{\overline{\mathbf{Q}}} = \begin{bmatrix} \overline{\overline{\mathbf{Q}}}^{(1)} & & & \\ & \overline{\overline{\mathbf{Q}}}^{(2)} & & \\ & & \ddots & \\ & & & \overline{\overline{\mathbf{Q}}}^{(N)} \end{bmatrix},$$

where $c_2 = \dfrac{2-\sigma_\nu}{2}$ and

$$\overline{\overline{\mathbf{Q}}}^{(k)} = \left[\frac{\sigma}{2}\mathbf{Q}^{(k)}(1,:); \frac{\sigma}{2}\mathbf{Q}^{(k)}(2,:); \mathbf{Q}^{(k)}(3,:) \right].$$

The resulting linear systems are solved using an iterative method, GMRES [34]. The computational cost associated with the GMRES grows as $O(n^2)$ with n being the size of the system. For problems with complicated three-dimensional flow domain, this solver could be computationally intensive. To accelerate the linear solver, the precorrected-FFT technique described in Chapter 1.3 is employed. With the precorrected-FFT acceleration technique, the cost can be reduced to $O(n\log n)$, and it grows almost linearly with the number of unknowns.

For nearby interactions, to ensure accuracy, the matrix-vector product is computed directly by forming the matrix entries explicitly and multiplying them with the source. This requires the integration of all the kernels (G, T, $\frac{\partial G}{\partial n}, \frac{\partial T}{\partial n}$) present in Equations (2.13), (2.21), (2.23) and (2.24) on each panel.

One challenge in calculating the integrals of G, T, $\frac{\partial G}{\partial n}, \frac{\partial T}{\partial n}$ on each panel is due to the singularities of the kernels. When the evaluation point coincides with the source panel (i.e., r goes to zero in Equations (2.13) and (2.21)), kernels G, T, $\frac{\partial G}{\partial n}, \frac{\partial T}{\partial n}$ become singular (see the expressions for G, T, $\frac{\partial G}{\partial n}, \frac{\partial T}{\partial n}$ in Equations (2.13) and (2.21)). To resolve this issue, we again separate the kernels into two parts: steady kernels (for example, G^S) and the remaining kernels (G–G^S). In the paragraphs that follow, we present methods employed to evaluate integrals of G, T, $\frac{\partial G}{\partial n}, \frac{\partial T}{\partial n}$ on each panel for both parts.

2.3.1 Steady kernels

The integrals that need to be evaluated are $\int_\Gamma \frac{1}{r} ds$, $\int_\Gamma \frac{\hat{x}_p}{r^3} ds$, $\int_\Gamma \frac{\hat{x}_p \hat{x}_q}{r^3} ds$, $\int_\Gamma \frac{\hat{x}_p \hat{x}_q}{r^5} ds$, and $\int_\Gamma \frac{\hat{x}_p \hat{x}_q \hat{x}_k \hat{x}_m}{r^7} ds$ with p, q, k, m =1,2,3.

For non-singular cases (that is, cases in which the evaluation point is not on the panel Γ), an analytical approach has been developed to evaluate these integrals based on a method described in [25].

According to [25], $\int_\Gamma \frac{1}{r} ds$, $\int_\Gamma \frac{\hat{x}_p}{r^3} ds$, and $\int_\Gamma \frac{\hat{x}_p \hat{x}_q}{r^3} ds$ can be evaluated analytically. We note that $\int_\Gamma \frac{\hat{x}_p \hat{x}_q}{r^5} ds$ and $\int_\Gamma \frac{\hat{x}_p \hat{x}_q \hat{x}_k \hat{x}_m}{r^7} ds$ can be expressed as linear combinations of $\int_\Gamma \frac{1}{r^3} ds$, $\int_\Gamma \frac{\hat{x}_p}{r^3} ds$, $\int_\Gamma \frac{\hat{x}_p \hat{x}_q}{r^3} ds$ and their derivatives as shown in Equations (2.30) – (2.35).

$$\int_\Gamma \frac{\hat{x}_p^{\,2}}{r^5} ds = \frac{1}{3}\left[\frac{\partial}{\partial x_i}\left(\int_\Gamma \frac{\hat{x}_p}{r^3} ds \right) + \int_\Gamma \frac{1}{r^3} ds \right], \tag{2.30}$$

$$\int_\Gamma \frac{\hat{x}_p \hat{x}_q}{r^5} ds = \frac{1}{3}\left[\frac{\partial}{\partial x_q}\left(\int_\Gamma \frac{\hat{x}_p}{r^3} ds \right) \right], \tag{2.31}$$

$$\int_\Gamma \frac{\hat{x}_p^{\,2} \hat{x}_q^{\,2}}{r^7} ds = \frac{1}{5}\left[\int_\Gamma \frac{\hat{x}_p^{\,2}}{r^5} ds + \frac{1}{3}\frac{\partial}{\partial x_q}\left(\int_\Gamma \frac{\hat{x}_q}{r^3} ds \right) + \frac{1}{3}\frac{\partial^2}{\partial x_p \partial x_q}\left(\int_\Gamma \frac{\hat{x}_p \hat{x}_q}{r^3} ds \right) \right], \tag{2.32}$$

$$\int_\Gamma \frac{\hat{x}_p^{\,4}}{r^7} ds = \int_\Gamma \frac{\hat{x}_p^{\,2}}{r^5} ds - \int_\Gamma \frac{\hat{x}_p^{\,2} \hat{x}_q^{\,2}}{r^7} ds - \int_\Gamma \frac{\hat{x}_p^{\,2} \hat{x}_k^{\,2}}{r^7} ds. \tag{2.33}$$

$$\int_\Gamma \frac{\hat{x}_p^{\,3} \hat{x}_q}{r^7} ds = \frac{1}{5}\left[2\int_\Gamma \frac{\hat{x}_p \hat{x}_q}{r^5} ds + \frac{1}{3}\frac{\partial}{\partial x_p}\left(\int_\Gamma \frac{\hat{x}_q}{r^3} ds \right) + \frac{1}{3}\frac{\partial^2}{\partial x_p^{\,2}}\left(\int_\Gamma \frac{\hat{x}_p \hat{x}_q}{r^3} ds \right) \right], \tag{2.34}$$

$$\int_{\Gamma} \frac{\hat{x}_p{}^2 \hat{x}_q \hat{x}_k}{r^7} ds = \frac{1}{5}\left[\int_{\Gamma} \frac{\hat{x}_p \hat{x}_k}{r^5} ds + \frac{1}{3} \frac{\partial^2}{\partial x_p \partial x_k} \left(\int_{\Gamma} \frac{\hat{x}_p \hat{x}_q}{r^3} ds \right) \right]. \tag{2.35}$$

Here, $p, q, k = 1, 2, 3$ and $p \neq q \neq k$. Therefore, by taking the derivatives of the analytical expressions of $\int_{\Gamma} \frac{1}{r} ds$, $\int_{\Gamma} \frac{\hat{x}_p}{r^3} ds$ and $\int_{\Gamma} \frac{\hat{x}_p \hat{x}_q}{r^3} ds$, the analytical solutions for $\int_{\Gamma} \frac{\hat{x}_p \hat{x}_q}{r^5} ds$ and $\int_{\Gamma} \frac{\hat{x}_p \hat{x}_q \hat{x}_k \hat{x}_m}{r^7} ds$ can be readily obtained.

In the singular cases (that is, cases in which the evaluation point ($\boldsymbol{\xi}$) is on the panel Γ), the integration of *G* kernel can be computed analytically [42]) via a transformation similar to the one described in [23]. The Cauchy principal value of $\int_{\Gamma} Tnds$ is simply zero on a flat panel because $\hat{x}_k n_k = 0$. Regarding the integration of $\frac{\partial G}{\partial n}$ on Γ, we employed a semi-analytic approach. Let

$$I^+ = \lim_{\lambda \to 0} \int_{\Gamma} \frac{\partial G_{qp}(\mathbf{x}, \mathbf{y})}{\partial x_m} n_m(\boldsymbol{\xi}) ds(\mathbf{y}), \qquad \text{where } \mathbf{x} = \boldsymbol{\xi} + \lambda \mathbf{n}(\boldsymbol{\xi}), \quad \lambda > 0,$$

and

$$I^- = \lim_{\lambda \to 0} \int_{\Gamma} \frac{\partial G_{qp}(\mathbf{x}, \mathbf{y})}{\partial x_m} n_m(\boldsymbol{\xi}) ds(\mathbf{y}), \qquad \text{where } \mathbf{x} = \boldsymbol{\xi} - \lambda \mathbf{n}(\boldsymbol{\xi}), \quad \lambda > 0.$$

It can be proved [7] that

$$\overset{cpv}{\int_{\Gamma}} \frac{\partial G_{qp}(\boldsymbol{\xi}, \mathbf{y})}{\partial \xi_m} n_m(\boldsymbol{\xi}) ds(\mathbf{y}) = \frac{1}{2}(I^+ + I^-) \tag{2.36}$$

Thus, the Cauchy principal value of $\int_{\Gamma} \frac{\partial G}{\partial n} ds$ was calculated by averaging the limiting values of the integrals as the evaluation point ($\mathbf{x}$) approaches the boundary along the

normal direction from the external and internal sides of the boundary. These limiting values were obtained via the analytical expressions derived in non-singular cases.

To verify, the proposed semi-analytic approach was used to compute $\int_{\Gamma}^{cpv} Tnds$, which in theory is equal to zero. Table 1 shows the corresponding I^+ and I^- and the average value of these two for several panels located on the surface of a sphere. (Only results for $\int_{\Gamma}^{cpv} T_{11k} n_k ds$ are shown. Other components of this kernel behave similarly to that of $\int_{\Gamma}^{cpv} T_{11k} n_k ds$.) The semi-analytic approach produces accurate results in this case.

Table 1. Calculation of $\int_{\Gamma}^{cpv} \frac{\hat{x}_i^2 \hat{x}_k n_k}{r^5} ds$ via the semi-analytic approach.

	$\frac{\lambda}{\min(panel\_diag)} = 10^{-9}$			$\frac{\lambda}{\min(panel\_diag)} = 10^{-8}$			$\frac{\lambda}{\min(panel\_diag)} = 10^{-7}$		
	I^+	I^-	$\frac{I^+ + I^-}{2}$	I^+	I^-	$\frac{I^+ + I^-}{2}$	I^+	I^-	$\frac{I^+ + I^-}{2}$
Panel 1	314.71	-314.71	0	314.71	-314.71	0	314.709	-314.709	0
Panel 2	232.05	-232.05	0	232.05	-232.05	0	232.05	-232.05	0
Panel 3	200.66	-200.66	0	200.66	-200.66	0	200.66	-200.66	0

As for the evaluation of $\int_{\Gamma}^{cpv} (u(\mathbf{y}) - u(\xi)) \frac{\partial T}{\partial n(\xi)} n(\mathbf{y}) ds$, since the velocity is constant within a constant element, therefore $u_p(\mathbf{y}) = u_p(\xi)$, $\mathbf{y} \in \Gamma$ and

$$\int_{\Gamma}^{cpv} (u_p(\mathbf{y}) - u_p(\xi)) \frac{\partial T_{pqk}(\xi, \mathbf{y})}{\partial \xi_m} n_m(\xi) n_k(\mathbf{y}) ds(\mathbf{y}) = 0$$

in singular cases.

2.3.2 Remaining kernels

We have shown in Chapter 2.2 that the remaining kernels are no longer singular as $r \to 0$. Therefore, we can use standard Gauss Quadrature to compute the integral. In order to improve accuracy, for near-singular cases (that is, for cases in which the evaluation point is not on the panel Γ but is very close) and singular cases, we first use Taylor-series expansion to approximate the integrands and remove the r in the denominators explicitly and then compute the integrals using Gauss Quadrature. By truncating the expansion at the right term, we can achieve the desired order of approximation.

2.4 RESULTS AND DISCUSSION

In this section, we first discuss the studies of a canonical problem, a sphere oscillating in an unbounded viscous slip flow. The purpose of studying this problem is to verify the derived boundary integral formulations and the developed numerical codes for slip Stokes flows. Next, we present two applications of the developed numerical tool. In the first application, we simulate the drag force (defined as the amplitude of the drag) acting on a plate oscillating above a fixed plate in an unbounded slip flow. In the second application, we simulate the drag force on a comb structure due to the surrounding slip flow and give a comparison with the drag force due to no-slip flows. In all the examples considered, isothermal conditions are assumed. This assumption is valid because of the typically small Mach number M, defined as the ratio of the flow velocity v to the sonic velocity c, associated with MEMS devices. The mean free path (λ) of molecules is calculated according to the Knudsen number (Kn) and the characteristic length (L) of flows ($\lambda = Kn \cdot L$).

2.4.1 Case study: A sphere oscillating in an unbounded fluid

Figure 4 illustrates a sphere oscillating with a velocity of $u = u_0 \cos(\omega t)$ in an unbounded fluid. The drag force exerted on the sphere is obtained by first simulating the traction forces on the boundary of the sphere using our 3-D codes and then integrating the real part of traction forces along the surface of the sphere. In all the simulations, non-dimensionalized parameters are used (the radius of the sphere a = 1, u_0 = 1). The accommodation coefficient σ_v is set to 1.0.

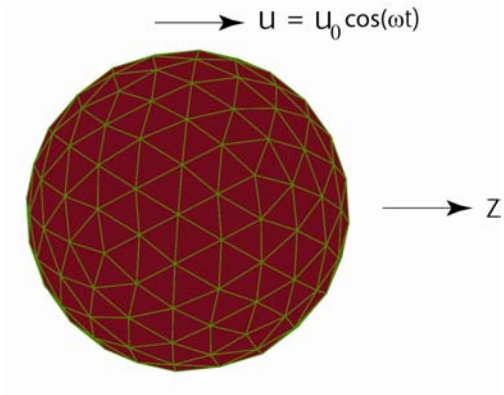


Figure 4. A discretized oscillating sphere.

2.4.1.1. Drag force due to no-slip flows

By letting Kn = 0 in our slip Stokes codes, we can calculate the drag force on the oscillating sphere due to no-slip flows. For such cases, the analytical solutions exist [17]. By comparing our simulation results with the analytical results, we have obtained the first confirmation of the fidelity of our codes. Figure 5 and Figure 6 are the convergence plots of drag forces simulated using our slip codes corresponding to Models I and II respectively (Kn = 0 in both cases). From these plots, it is clear that good convergences

have been achieved. Table 2 and Table 3 give the errors of our results compared with the analytical solutions at different discretizations. With increasingly finer meshes, the differences between the simulated results and the analytical results are reduced, indicating good agreements. From the tables, it is also shown that at higher frequencies, finer meshes are needed to achieve greater accuracy. This is consistent with the fact that at high frequencies, the development of the boundary layers requires boundary elements of small sizes.

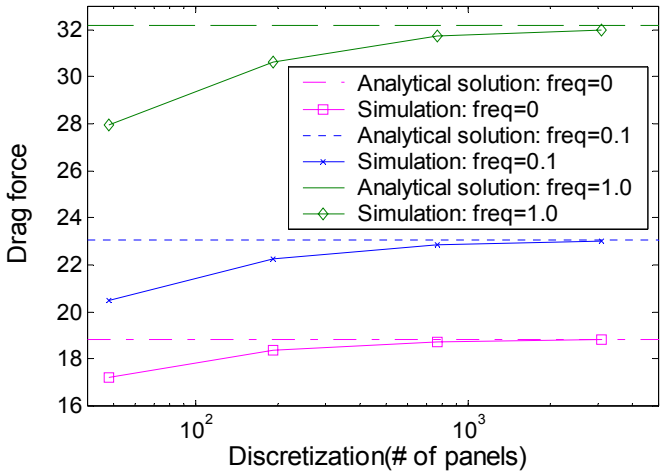


Figure 5. Convergence plot of drag force using slip code I (Model I, Kn = 0, b=0 , σ_v =1.0).

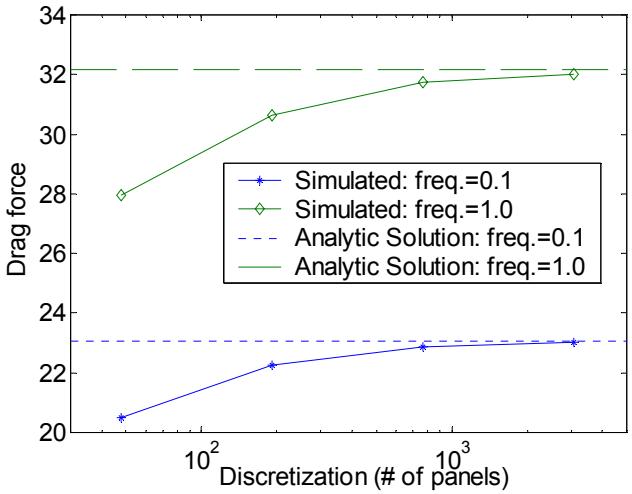


Figure 6. Convergence plot of drag force using slip code II (Model II, Kn = 0, σ_v =1.0).

Table 2. Errors of drag force obtained from slip code I comparing with [41].

$\left(\frac{|F_{drag}-F_{ref}|}{F_{ref}}\times 100\%\right)$

Number of surface panels→	48	192	768	3072
frequency =0.0	8.70%	2.55%	0.67%	0.17%
frequency =0.1	11.16%	3.56%	0.88%	0.25%
frequency =1.0	13.13%	4.85%	1.32%	0.54%

Table 3. Errors of drag force obtained from slip code II comparing with [41].

$\left(\frac{|F_{drag}-F_{ref}|}{F_{ref}}\times 100\%\right)$

Number of surface panels→	48	192	768	3072
frequency =0.1	11.16%	3.29%	1.06%	0.20%
frequency =1.0	13.13%	4.05%	1.14%	0.16%

2.4.1.2. Drag force due to slip flow (Steady Case)

In this case study, we calculated the drag force exerted on the sphere due to slip flows when frequency is zero. Results were compared with an asymptotic solution provided by Bassat [14]. When *Kn* is small, the asymptotic solution for the drag force on the sphere moving in a slip flow is given by

$$F_{drag} = 6\pi\mu a U \frac{\beta a + 2\mu}{\beta a + 3\mu} \tag{2.37}$$

Here μ is the viscosity of the fluid, a is the characteristic dimension of the sphere, U is the non-dimensionalized velocity of the sphere, and β is a coefficient. Comparing with the slip Model I shown in Equation (2.10), the relationship between the coefficient β and *Kn* is $\frac{\mu}{\beta a} = \frac{(2-\sigma_v)\cdot(Kn)}{\sigma_v + (\sigma_v - 2)\cdot(Kn)}$. At zero Knudsen number, F_{drag} reduces to $6\pi\mu a U$.

For different Knudsen numbers, drag forces per unit viscosity were simulated with

different discretizations. The results are plotted in Figure 7. Table 4 gives a comparison between the simulated drag forces and the Basset's solutions. In all the simulations, the slip coefficient b was chosen as 0. Thus the slip model used in this calculation is a first-order model. It is clear from Figure 7 that with the increasing number of panels, our simulations have achieved a good convergence. Also as indicated in Table 4, the numerical results based on the 3-D simulations agree well with those obtained from the asymptotic solution in the slip regime.

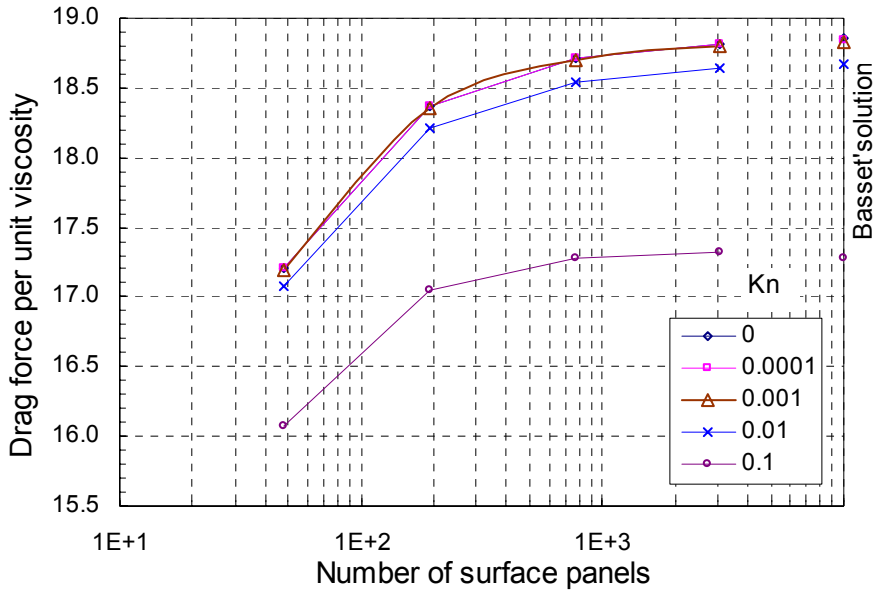


Figure 7. Drag forces per unit viscosity simulated using the first-order slip model for steady case (σ_v =1.0, b=0).

Table 4. Relative errors of the simulated drag forces compared with the Bassat's solutions.

Number of surface panels→	48	192	768	3072
$Kn = 0.0001$	8.69%	2.56%	0.70%	0.18%
$Kn = 0.001$	8.68%	2.55%	0.70%	0.17%
$Kn = 0.01$	8.51%	2.43%	0.63%	0.13%
$Kn = 0.1$	6.97%	1.30%	0.03%	0.24%

The effects of high-order slip models on the drag force were also studied. Figure 8 shows the drag forces simulated using a second-order slip model with different Knudsen numbers. The slip coefficient b is −1.8571 and was calculated based on the Equation (2.9). Table 5 shows a comparison between the drag forces obtained from the first-order slip model and the second-order slip model. As expected, there is almost no difference when the Knudsen number is small. At Kn = 0.1, the difference in drag force is 1.15%.

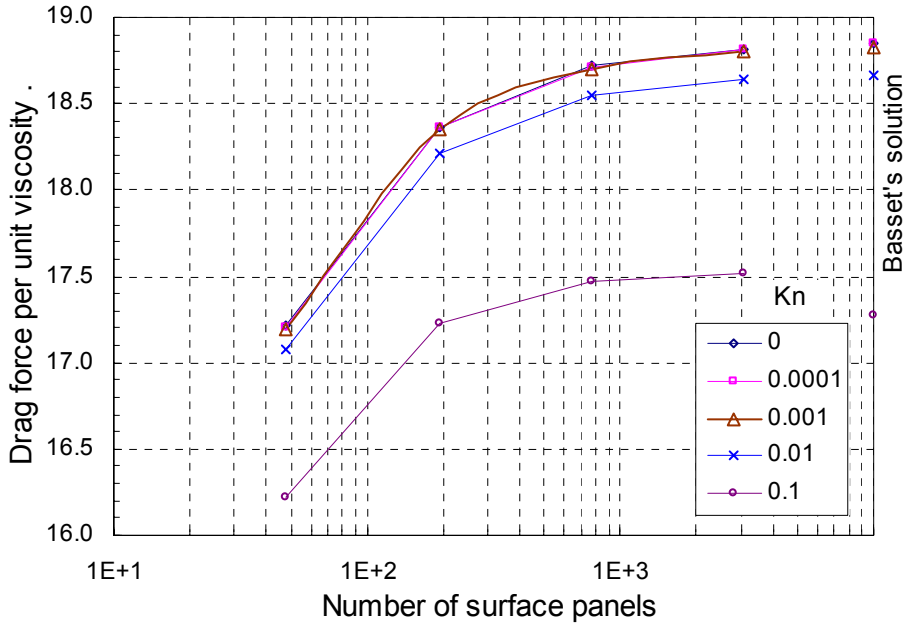


Figure 8. Drag forces per unit viscosity simulated using the second-order slip model for steady case (σ_v =1.0, b = −1.8571).

Table 5. Relative errors in drag forces obtained from the second-order slip model compared with the first-order slip model ($\frac{|F|_{b=-1.8571} - F|_{b=0}|}{F|_{b=0}}$) for steady case.

Number of surface panels→	48	192	768	3072
Kn = 0.0001	0	0	0	0
Kn = 0.001	0	0	5.35E−4	0
Kn = 0.01	1.41E−4	1.54E−4	1.67E−2	1.72E−4
Kn = 0.1	9.37E−3	1.02E−2	1.11E−2	1.15E−2

To illustrate the speed of our code, a plot of the computational time versus the number of panels is shown in Figure 9 (only the case of $Kn = 0$ is shown for clarity, for other Kn, the plot exhibits a similar trend). Two lines corresponding to $O(n^2)$ and $O(n)$ are also shown in Figure 9 for comparison. It is clear that the computational time of our code is much less than $O(n^2)$.

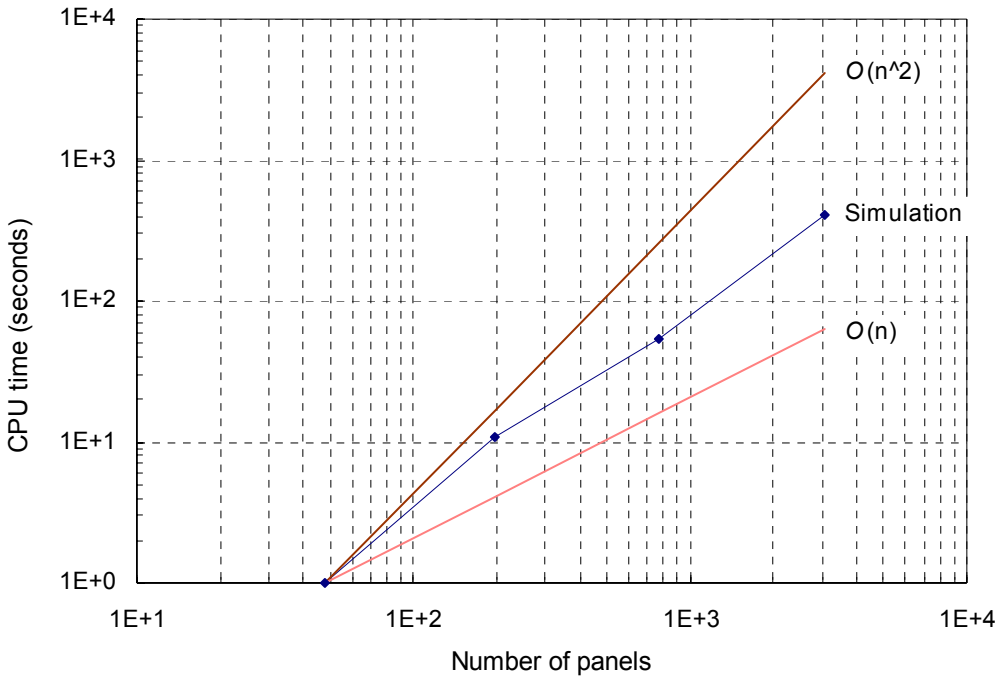


Figure 9. The computation time versus the number of panels for steady case.

2.4.1.3. Drag force due to slip flow (Unsteady Case)

For an oscillating sphere with slip boundary conditions, there is unfortunately no analytical solution available to compare with. We show here the results obtained from two slip codes corresponding to two slip models. The comparison between two sets of results shows some degree of validation of our codes. Table 6 and Table 7 give the drag

forces at *frequency* = 0.01 and Figure 10 and Figure 11 are the corresponding convergence plots. In Model I, the slip coefficient b is also set to be −1.8571 to include the second order slip effects.

Table 6. Simulated drag forces acting on the oscillating sphere based on Slip Model I (σ_v = 1.0, b = −1.8571, *frequency* = 0.01).

Number of surface panels→	48	192	768	3072
$Kn = 0$	18.2913	19.6172	20.0305	20.1434
$Kn = 0.001$	18.2758	19.5992	20.0109	20.1229
$Kn = 0.01$	18.1421	19.4435	19.8411	19.9456
$Kn = 0.1$	17.1850	18.3270	18.6203	18.6708

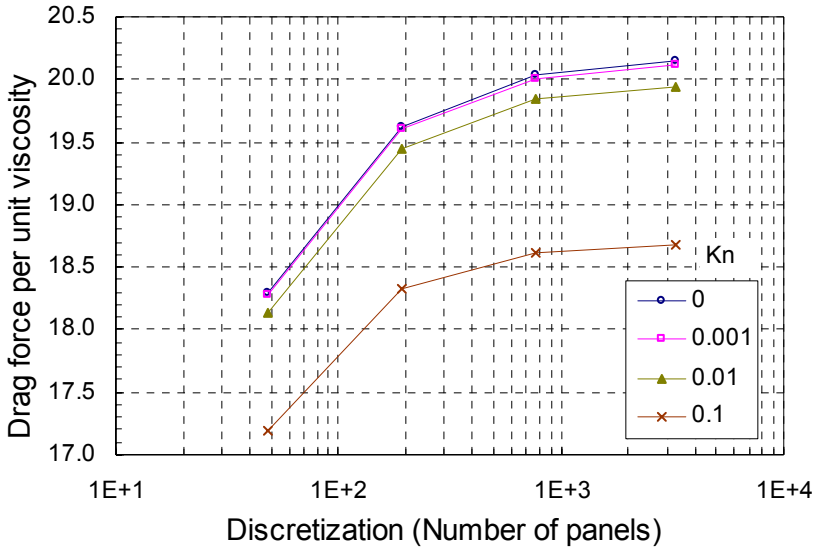


Figure 10. Convergence plot of drag force using slip code I for unsteady case (Model I, b = 0, σ_v = 1.0).

Table 7. Simulated drag forces acting on the oscillating sphere based on Slip Model II (σ_v = 1.0, *frequency* = 0.01).

Number of surface panels→	48	192	768	3072
$Kn = 0$	18.3217	19.5714	19.9685	20.1284
$Kn = 0.001$	18.3041	19.5429	19.9204	20.0527
$Kn = 0.01$	18.1579	19.2971	19.5198	19.4769
$Kn = 0.1$	17.0325	17.6982	18.2122	18.4529

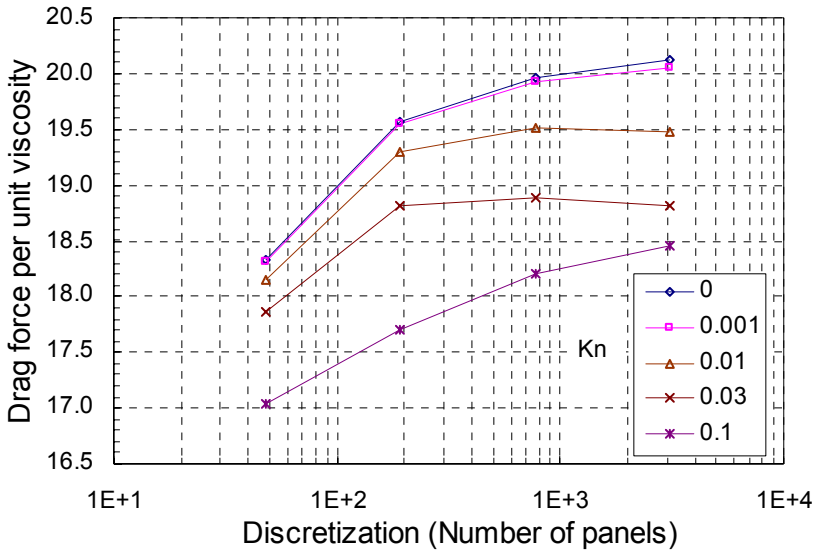


Figure 11. Convergence plot of drag force using slip code II for unsteady case (Model II, σ_v = 1.0).

Overall, convergence and consistency between two sets of results have been achieved. Drag forces obtained from Model II are in general smaller than those obtained from Model I particularly in the cases that have high Knudsen number. This is probably because Model II includes all the high-order terms of *Kn* while Model I only includes up to second-order term. This fact also explains the observed large difference between the two sets of results at large Knudsen number. It is also observed that the number of

iterations required to reach convergence is higher in Model II than in Model I. This perhaps is because the resultant linear system from Model II is ill-conditioned (the condition number of the corresponding system is on the order of 10^5). The ill-conditioned nature of the system in Model II can be understood by examining Equation (2.27) carefully. Equation (2.27) is obtained by collocating the integral equations at two points that are very close to each other (about one mean free path away). Thus, although Model II is easy to implement and includes more high order *Kn* terms compared to Model I, the ill-conditioned system limits its applications.

2.4.2 Application 1: Parallel plates

The first application we consider is a pair of parallel plates immersing in air with one plate fixed while the other one oscillats parallel to the fixed plate (shown in Figure 12). The top plate has dimensions of 50μm×50μm×5μm. The bottom one has dimensions of 100μm×100μm×20μm. The gap between them is 2μm.

Figure 12. A pair of parallel plates

For the unsteady case, the velocity profile is shown in Figure 13. When this kind of structure is used in surface-micromachined devices in which the plates move in the

direction parallel to their surfaces, the squeeze-film effective viscosity does not apply. For infinite long plates with no-slip boundary condition, the fluid can be modeled using the 1D stokes model [41], in which the amplitude of the fluid oscillation decays exponentially with the distance from the plate surface, while the phase lag increases linearly Figure 13.

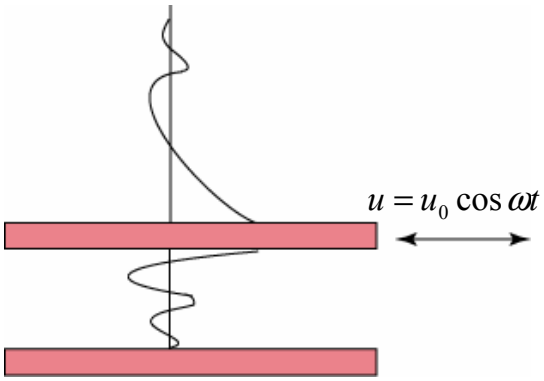


Figure 13. Velocity profile for a pair of resonating parallel plates

The corresponding drag forces for the fluids above the upper plate and the fluids between the two plates can be expressed as Equations (2.38) and (2.39) respectively.

$$F_{drag} = \mu\beta A u_0, \quad \text{where } \beta = \sqrt{\frac{\omega}{2\nu}}, \tag{2.38}$$

$$F_{drag} = \mu\beta A u_0 \left(\frac{\sinh 2\beta d + \sin 2\beta d}{\cosh 2\beta d - \cos 2\beta d} \right), \tag{2.39}$$

where ν is the dynamic viscosity of the fluid and $\nu = \mu / \rho$, and ω is the angular velocity. Equations (2.38) and (2.39) imply that a slowly oscillating plate in a viscous medium is expected to drag substantially more fluid compared to a fast moving one in a fluid of low viscosity, so the drag force will increase with frequency.

Drag forces on the oscillating plate (*frequency* = 19200 Hz) at different Knudsen numbers are calculated and shown in Table 8 and Table 9. Results shown in Table 8 are obtained using slip Model I and those shown in Table 9 are calculated using slip Model II. In slip Model I, the slip coefficient *b* is set to be 0. Thus only the first-order slip effect is included. The total drag force is broken down into three components, namely a top force (force due to the fluid on top of the oscillating plate), a bottom force (force came from the fluid between the plates) and a side force (force acting on the sides of the oscillating plate). Only the top, bottom and total forces are shown in the tables. The results from two different slip models agree with each other within 4%.

Table 8. Drag forces per unit viscosity (10^{-6} m^2/sec) for plate problem obtained from Model I (σ =1.0, *b*=0, *frequency* =19200Hz).

	Top drag	Bottom drag	Total drag
Kn = 0.0	260.14	1303.24	1972.16
Kn = 0.03	259.99	1265.35	1931.39
Kn = 0.06	259.84	1229.67	1892.91
Kn = 0.1	259.63	1185.23	1844.85

Table 9. Drag forces per unit viscosity (10^{-6} m^2/sec) for plate problem obtained from Model II (σ =1.0, *frequency* =19200Hz).

	Top drag	Bottom drag	Total drag
Kn = 0.0	258.55	1497.73	2061.96
Kn = 0.03	256.06	1450.11	2012.42
Kn = 0.06	253.68	1405.90	1966.23
Kn = 0.1	250.69	1351.65	1909.34

To validate the accuracy of the results, traction forces located at the center of the bottom face of the oscillating plate are calculated and compared with the analytic solutions obtained from the 1-D Stokes model (Equation (2.39)). In the 1-D Stokes slip model, the plates are assumed to be infinitely long and the slip model is the first-order Maxwell slip model (Equation (2.40)) [41]. The results are shown in Table 10. Good agreements have been achieved in particular at cases with low *Kn* numbers. At high *Kn* numbers, the accuracy can be improved by using a finer mesh.

$$\mu_{eff} = \frac{\mu}{1 + 2 \cdot (Kn)} \tag{2.40}$$

Table 10. Traction force (per unit viscosity per area) at the center of the bottom face of the oscillating plate (10^{-6} /sec).

	Slip Model I	Slip Model II	1-D Stokes Slip Model
Kn = 0.0	0.5040	0.5029	0.5000
Kn = 0.03	0.4894	0.4919	0.4717
Kn = 0.06	0.4756	0.4808	0.4464
Kn = 0.1	0.4584	0.4663	0.4167

Although the simulated center traction force agrees reasonably well with the 1-D analytic solution, the total drag force obtained from our 3-D simulation differs significantly from the 1-D solution (as much as 23%). This indicates strong edge and finite-size effects, namely the 3-D effects. In an experimental study conducted by Zhang *et al.*, it has also been pointed out that the 3-D effects of air damping are significant in laterally driven microstructures [47]. In this study, the effect of structure thickness on the damping is also examined. The quality factors of two plates oscillating above a substrate

are measured. These two plates have identical lateral dimensions but different thickness. The measured quality factor of the thick plate (h = 4.1 μm) is 1.6 times the quality factor of the thin plate (h = 2.1 μm). According to the relationship between the quality factor Q and the stiffness k, the mass m and the damping coefficient c ($c = drag\ force / velocity$),

$$Q = \frac{\sqrt{km}}{c},$$

the ratio between the drag forces acting on two plates from the experimental study is

$$\frac{drag\ force\ on\ thick\ plate}{drag\ force\ on\ thin\ plate} = \frac{c_1}{c_2} = \frac{Q_2}{Q_1}\frac{\sqrt{k_1 m_1}}{\sqrt{k_2 m_2}} = \frac{Q_2}{Q_1}\frac{h_1}{h_2} = 1.2.$$

Note both k and m depends on h linearly [47]. The experimental result shows that the drag force increases with thickness. Our simulation has also verified this fact. We applied our codes to calculate the drag force on a thin plate (50μm×50μm×2.5μm) and compared the results with those obtained for a thick plate (50μm×50μm×5μm) in Table 11. An increase in drag force results as thickness increases.

Table 11. Drag forces per unit viscosity on oscillating plates with different thickness (Kn = 0.03, σ =1.0).

	Thin plate (h = 2.5 μm)	Thick plate (h = 5μm)	Ratio (thick vs. thin)
Model I	1837.55	1931.39	1.05
Model II	1837.89	2012.42	1.09

2.4.3 Application 2: Micro comb structure

In the micro-resonator shown in Figure 14, the comb drive is used to drive the beam into resonance. When the device is packaged in air, the performance of the resonator is largely affected by the viscous damping caused by air. Previous modeling of the viscous drag force based on no-slip boundary condition has achieved a very good agreement with the experimental results (within 10%) [45]. However, the typical gap between the resonator and the substrate in such a device is about 1~2 μm. At the standard air conditions, i.e., room temperature and 1 atmosphere pressure, the Knudsen number of air in this device is around 0.06~0.03. A question about the validation of the no-slip assumption arises. To study the effects of rarefaction on the drag force of the resonators, we simulated the drag force on a comb structure (similar to the moving comb in the micro resonator as shown in Figure 14) moving in air as shown in Figure 15 (coarse mesh is shown for clarity). The number of elements corresponding to the finest mesh used in the simulation is 12848. It took about 2 hours to run this mesh on a PC with an AMD ATHLON XP 1900 processor (1.6GHz). It should be pointed out that near the corners sufficient elements are needed in order to resolve the relatively more complicated flow pattern. The results at *frequency* = 19200Hz and at different Knudsen numbers are recorded in Table 12. The percentage shown in the table is the reduction of drag force compared with the no-slip case ($Kn = 0$). When $Kn = 0.03$, the difference in drag force is at most 2%, indicating small drag reduction in this case. According to Equation (2.10), the velocity of flow is proportional to the normal derivative at the wall. But in this case the normal derivative is small. If we can include the substrate in simulation, we can expect higher normal derivative of velocity thus more drag reduction.

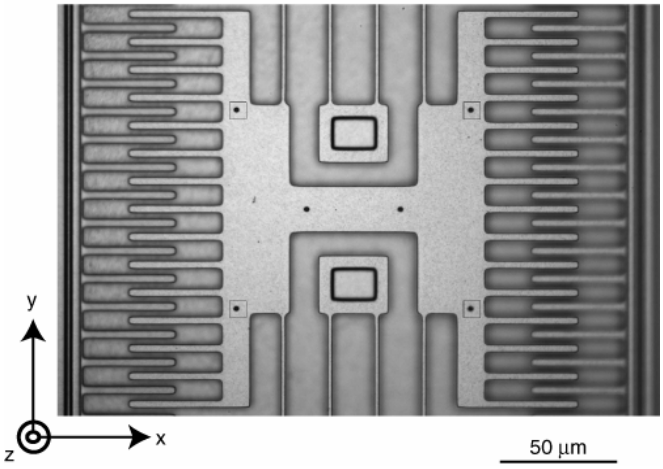


Figure 14. SEM picture of a micro resonator (top view).

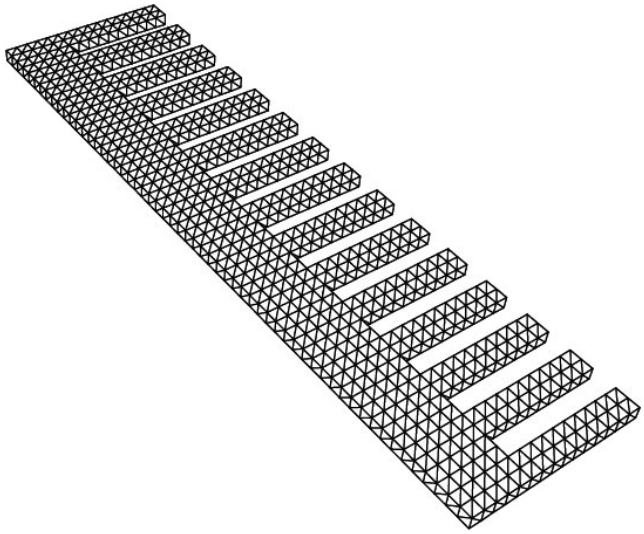


Figure 15. A discretized comb drive structure.

Table 12. Drag forces per unit viscosity on an oscillating comb structure (*frequency* =19200Hz).

	$Kn = 0.0$	$Kn = 0.03$	$Kn = 0.06$	$Kn = 0.1$
Model I (with b=0)	408.319	406.479(0.45%)	404.766(0.87%)	402.631(1.39%)
Model II	409.278	401.570(1.88%)	396.161(3.20%)	392.697(4.05%)

2.5 CONCLUSIONS

In this chapter we describe an efficient numerical approach for solving oscillatory incompressible Stokes equations with slip boundary conditions. Two slip models are considered and the boundary integral equations for slip flows are derived for both models. These integral equations are then solved using the boundary element method combined with the precorrected-FFT accelerated techniques. Based on this method, 3-D codes have been developed. The codes are tested by simulating the drag force on a sphere moving in an unbounded slip flow. In the steady state case, i.e., the sphere moves at a constant speed, the simulation results are compared with the analytic solutions and a good agreement has been achieved. In oscillating cases, the slip codes are first validated by setting $Kn = 0$ in the codes and comparing the simulated drag forces with the analytical results. Good agreements between the simulation and analytic solutions have been achieved. For slip cases in which no analytic solution is available for comparison, drag forces obtained from two different slip models are compared. As expected, we have obtained a good agreement between the two sets of results at low Knudsen numbers. At high Knudsen numbers, because of the different order of Kn terms included in the two slip models, a discrepancy between two sets of results has been observed. It is also observed that Slip Model II requires more iterations to achieve certain convergence because of the ill-conditioned system. As applications of the developed 3D slip codes, drag forces on a pair of plates and a comb structure moving in air at different Knudsen numbers are simulated. Qualitative agreements between the simulation results and the experimental results have been achieved.

CHAPTER 3

A FAST BEM SOLVER FOR NON-HOMOGENEOUS AND NONLINEAR PROBLEMS IN 3-D COMPLEX DOMAINS

3.1 INTRODUCTION

It is perhaps fair to say that to date most applications of the BEM have been limited to linear and homogeneous problems. This is largely due to the difficulties associated with the evaluation of volume integrals in the boundary integral formulations that result from non-homogeneous or nonlinear problems.

For example, the Poisson equation, as shown in Equation (3.1), can be cast into a boundary integral formulation shown in Equation (3.2),

$$\nabla^2 u = b, \tag{3.1}$$

where u is an unknown function of the independent variable $\mathbf{x}$ and the time variable t, and b represents all the nonlinear terms, which is a function of u and $\mathbf{x}$.

$$c(\mathbf{x})u(\mathbf{x}) + \int_{\partial\Omega} \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial n(\mathbf{y})} u(\mathbf{y}) dS(\mathbf{y}) - \int_{\partial\Omega} G(\mathbf{x},\mathbf{y}) \frac{\partial u(\mathbf{y})}{\partial n(\mathbf{y})} dS(\mathbf{y}) = -\int_{\Omega} G(\mathbf{x},\mathbf{y}) b(\mathbf{y}) dv(\mathbf{y}) \tag{3.2}$$

where

$$c(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \Omega \\ 0, & \mathbf{x} \notin (\Omega \cup \partial\Omega) \\ \alpha, & \mathbf{x} \in \partial\Omega \end{cases}.$$

Here α is the 'free term' coming from evaluation of the integral $\int_{\partial\Omega} \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial n(\mathbf{y})} u(\mathbf{y}) dS(\mathbf{y})$ with strong singularity when $\mathbf{x}$ is approaching the boundary. Ω is the domain of the problem with boundary $\partial\Omega$, $n(\mathbf{y})$ is the unit outward normal vector at the field point $\mathbf{y}$, $\mathbf{x}$

is the evaluation point and $G(\mathbf{x},\mathbf{y})$ is the Green's function of the Laplace operator. In 3-D space, it is given by

$$G(\mathbf{x},\mathbf{y}) = \frac{1}{4\pi r}, \text{ where } r = |\mathbf{x} - \mathbf{y}|.$$

On the right-hand-side of Equation (3.2), the integral which contains b is a volume integral.

The volume integral can also arise from a time-dependent problem. Consider a equation illustrated in Equation (3.3),

$$u_t = \nabla^2 u + b \tag{3.3}$$

we can use a finite-difference scheme to approximate u_t in terms of u at the current time step and its values at previous time steps. For instance, let

$$u_t \approx (u_m - u_{m-1}) / \Delta t$$

where u_m is u at m^{th} time step, u_{m-1} is u at $(m\text{-}1)^{th}$ step and Δt is the time step. The corresponding boundary integral equation for Equation (3.3) is [40]:

$$c(\mathbf{x})u_m(\mathbf{x}) + \int_{\partial\Omega} \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial n(\mathbf{y})} u_m(\mathbf{y}) dS(\mathbf{y}) - \int_{\partial\Omega} G(\mathbf{x},\mathbf{y}) \frac{\partial u_m(\mathbf{y})}{\partial n(\mathbf{y})} dS(\mathbf{y}) = -\int_{\Omega} G(\mathbf{x},\mathbf{y}) \left(b(\mathbf{y}) - \frac{u_{m-1}}{t} \right) dv(\mathbf{y}) \tag{3.4}$$

Again, the above integral equation (3.4) contains a volume integral. To solve integral equations (3.2) or (3.4), standard BEM can be applied [3]. However, a crucial step in the implementation is the evaluation of the volume integral that appears on the right-hand-side of the above integral equations.

Efficient and accurate treatment of these volume integrals is the major bottleneck in the extension of the BEM to nonlinear or non-homogeneous problems. The challenge and the significance of this problem have attracted the attention of many researchers, and over

last several decades, various approaches have been proposed and developed. For a detailed review, readers are referred to a paper by Hsiao, *et al*. [15]. In summary, these approaches can be classified into two main categories: domain discretization free methods (or meshless methods) and cell-based direct integration schemes.

3.1.1 Domain discretization free methods

Domain discretization free methods are based on either transforming of volume integral to surface integral [10][46] or particular solution methods [18][29]. The exact transformation of the volume integral to equivalent boundary integrals is limited by the significantly more complicated fundamental solutions. Among all the efforts for eliminating domain integrals associated with the BEM, the dual reciprocity method (DRM) [29] based on particular solution approach is probably the most popular one. Nardini and Brebbia [24] first developed the DRM to analyze free vibration problems in which the non-homogeneous terms in the governing equation were approximated with a weighted summation of radial basis functions (RBF). A particular solution was available for the RBF which enabled the elimination of the domain integral through a second reciprocity. Instead of complete discretization of the domain, internal nodal points at the center of each radial basis function are used and must be distributed in the domain. Nowak and Brebbia [26][27] developed a technique for analysis of transient heat transfer problems in which the domain integral was replaced by an infinite series of boundary integrals involving higher order fundamental solutions. They showed in their method which they called the multiple reciprocity method (MRM), that this series of integrals converged quickly and could be evaluated efficiently. Particular solution methods use a

closed form representation of a particular solution to eliminate the volume integral [1]. The major advantage of this type of method is that the boundary only nature of the BEM is retained.

For example, to solve the Poisson problem shown in Equation (3.1) using the dual reciprocity method, the non-homogeneous function $b(\mathbf{x})$ is first approximated using radial basis functions. That is,

$$b(\mathbf{x}) \approx \sum_{i=1}^{N} \alpha_i f_i \tag{3.5}$$

where N is the number of radial basis functions and the α_i's are coefficients to be determined. The particular radial basis functions are chosen empirically; a popular one is

$$f_i(\mathbf{x}) = 1 + r_i \tag{3.6}$$

where r_i is the distance between the point $\mathbf{x}$ and the center of the radial basis functions (internal nodes). Particular solutions $\hat{u}_i$ are available so that $\nabla^2 \hat{u}_i = f_i$. In particular, for the radial basis functions given in Equation (3.6),

$$\hat{u}_i = r_i^2 / 4 + r_i^3 / 9 \tag{3.7}$$

Now, using a second reciprocity on the domain integral in Equation (3.2), the boundary integral equation can be rewritten as

$$\begin{aligned} & c(\mathbf{x})u(\mathbf{x}) + \int_{\partial\Omega} \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial n(\mathbf{y})} u(\mathbf{y}) dS(\mathbf{y}) - \int_{\partial\Omega} G(\mathbf{x},\mathbf{y}) \frac{\partial u(\mathbf{y})}{\partial n(\mathbf{y})} dS(\mathbf{y}) \\ & = \sum_{i=1}^{N} \alpha_i \left[\eta(\mathbf{x})\hat{u}(\mathbf{x}) + \int_{\partial\Omega} \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial n(\mathbf{y})} \hat{u}(\mathbf{y}) dS(\mathbf{y}) - \int_{\partial\Omega} G(\mathbf{x},\mathbf{y}) \frac{\partial \hat{u}(\mathbf{y})}{\partial n(\mathbf{y})} dS(\mathbf{y}) \right] \end{aligned} \tag{3.8}$$

Now the above equation only involves discretization of the boundary. Internal nodes may be defined in the number and at the locations desired. When interior nodes are defined, each one is independently placed, and they do not form part of any element or cell, thus

only the coordinates are needed as input data. Hence, these nodes may be defined in any order. The α vector can be calculated by inverting the coefficient matrix **F** in Equation (3.5)

$$\boldsymbol{\alpha} = F^{-1} \cdot b(\mathbf{x}) .$$

However, as pointed out in [15][16] and also based on our own experience, the quality of these methods depends on the quality of the radial basis functions approximation. The choice of radial basis functions and the number and location of internal nodes are all determined empirically so the performance of DRM is often problem-dependent. We have tested the performance of DRM using a 1-D nonlinear equation: Fisher's equation $u_t = u_{xx} + u(1-u)$ with boundary condition $u(0) = u(1) = 0$ and initial condition $u = \sin(\pi x)$. The radial basis functions described in Equation (3.6) are used. It turned out that for this problem DRM performed poorly when less than 4 internal nodes were used, although the position of the internal nodes is not so important [8]. But a time step size as large as 0.1, which is in the order of n^{-1}, can be used to achieve good accuracy in this case. But for initial condition $u = \cos(\pi x)$, much worse convergence is observed using same RBFs and same series of internal nodes indicating the problem-dependence of this method.

Despite the proven applicability of the DRM, important theoretical questions remain. Finding the optimal methods for approximations is still an active research topic. In addition, very large, fully populated and often poorly conditioned matrices are generated [20], resulting in a significantly high computational cost.

3.1.2 Cell-based direct integration methods

Cell-based direct integration schemes employ an interior volume mesh to directly perform the integration. A major advantage of this type of scheme is high accuracy. Such approaches may also be efficient, as matrix sparsification techniques can be employed to accelerate volume integration. For example, in [16], the operation count per collocation node is reduced from $O(M)$ for classical Gaussian quadrature to $O(\log M)$ for multipole evaluation where M is the number of interior nodes.

In a cell-based direct integration method, the interior of the problem domain is discretized into finite volume elements and the non-homogeneous term b within the interior finite elements is approximated using isoparametric shape functions similar to the Finite Element Method (FEM). When the multipole acceleration technique is used to accelerate the volume integration, the domain integral is evaluated in two parts, a near field and a far field (Figure 16). In the near field, the classical domain integration is performed and matrix entries are explicitly stored for the near-field elements. In the far field, multipole expansions are used to approximate the matrix-vector product, so matrix entries do not need to be stored explicitly.

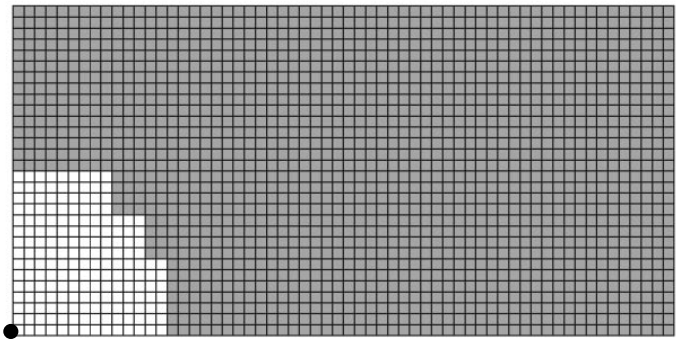


Figure 16. Near-field and far-field for the domain integral for the node highlighted by the dot. Notice that near-field elements (light grey) are far less than far-field elements (dark grey). Matrix entries are stored only for the near-field elements (light gray) [16].

In a paper by Ingber *et al*.[16], a comparison of the different integration schemes has been performed. It was found that the classical cell-based direct integration method coupled with fast multipole method may be significantly better (in terms of computational accuracy and efficiency) than both the dual reciprocity and particular solution methods. This approach is competitive with standard fast solvers like FFT, but it uses grids which are highly inhomogeneous. Moreover, a major disadvantage of cell-based direct integration schemes is the requirement of volume discretization of the problem domain. Regular cell-based methods are easily applied only in simple geometries, while the strong point of the BEM over domain-based techniques is the ability to treat complex domains without incurring severe discretization costs. In cases where the geometry is very complex, generating the domain mesh required for cell-based integration can be an extremely demanding task. Typically, domain meshing of complex geometries is performed automatically by starting from a discretization of all the surfaces, and creating an unstructured interior mesh. Algorithms to perform this task reliably in three dimensions exist. However, it has been shown that the mesh generation costs can be substantial, and that parallelization is difficult [36][37]. As a result, the advantages of cell based integration appear to be accompanied by the difficulties inherent in meshing complex domains, thus severely reducing the attractiveness of this technique, its advantages notwithstanding.

3.1.3 The auxiliary domain method

In an attempt to reduce the complexity caused by volume discretization, the auxiliary domain subtraction method [15][20] has been developed to simplify the mesh generation for multiply-connected problems.

In simple domains, discretization can be performed with little computational expense. In complex domains, for example three-dimensional multiply connected domains which describe two-phase dispersions of rigid particles in a carrier fluid, the discretization can become very difficult, and it is desirable to bypass this obstacle. Consider the multiply connected domain shown in Figure 17, consisting of a continuous phase containing a dispersion of separate phases. Let Ω_0 be the computational domain for the continuous phase (the cross hatched area), each Ω_i an auxiliary domain (not part of the computational domain), and let

$$\Omega_+ = \Omega_0 \cup \sum_{i=1}^{N} \Omega_i$$

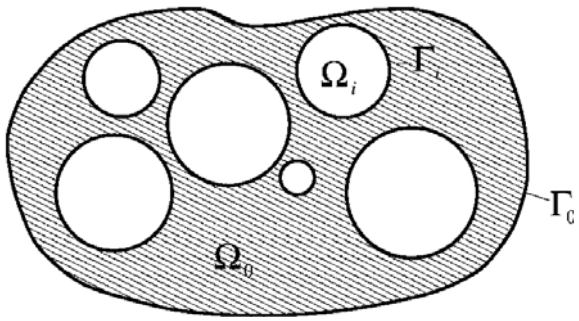


Figure 17. Schematic of a complex, multiply-connected domain in two dimensions. A domain integral must be evaluated over the computational domain Ω_0. There are N auxiliary domains Ω_i.

To evaluate the volume integral in Equation (3.2)

$$I = \int_{\Omega_0} G(\mathbf{x},\mathbf{y}) b(\mathbf{y}) d\Omega ,$$

because the discretization of Ω_0 in general is a complex and time-consuming operation, one could perform the integral over Ω_+, then evaluate the integrals of the same function over each of the auxiliary domains Ω_i and subtract these auxiliary integrals from the integral over Ω_+, resulting in the desired integral over Ω_0:

$$\int_{\Omega_0} G(\mathbf{x},\mathbf{y}) b(\mathbf{y}) d\Omega = \int_{\Omega_+} G(\mathbf{x},\mathbf{y}) b(\mathbf{y}) d\Omega - \sum_{i=1}^{N} \int_{\Omega_i} G(\mathbf{x},\mathbf{y}) b(\mathbf{y}) d\Omega$$

The advantage is that the domains Ω_+ and Ω_i can be discretized much more simply than Ω_0, as shown in Figure 18. However, although in many cases $b(\mathbf{y})$ is defined over Ω_0 and each Ω_i, its gradient is discontinuous at the interface between different domains due to the different physical properties of each phase, resulting in difficulties in the numerical integration. An auxiliary function associated with each auxiliary domain Ω_i needs to be defined to overcome this difficulty. The auxiliary function needs to satisfy certain criteria that arise from numerical integration considerations, e.g., continuity requirements.

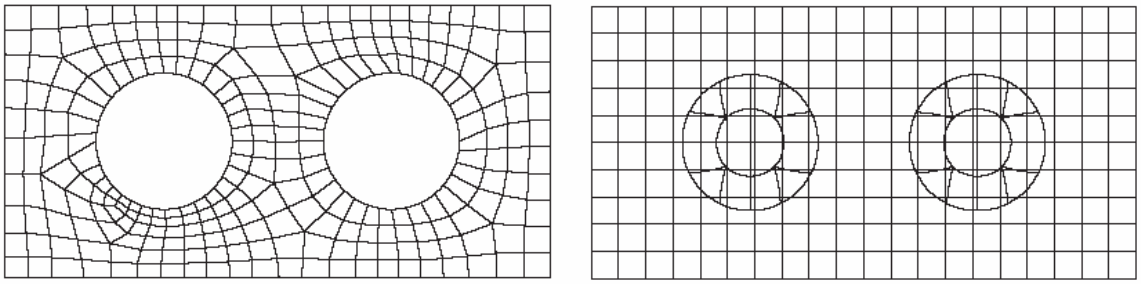


Figure 18. A mesh where only Ω_0 is discretized using an unstructured quadrilateral mesh (left), and the corresponding discretization, where the augmented domain Ω_+ and each of the subdomain Ω_i are discretized (right). Notice that elements used to discretize the augmented domain in some cases span both Γ_0 and at least one Γ_i.

Here the auxiliary functions are reconstructed using information at the interface between Ω_i and Ω_0 by using a linear combination of radial basis functions which is chosen empirically. Similarly, the quality of ADM is limited by the choice of radial basis functions. Moreover, such a method is effective only when the inclusions or holes are of simple geometries, for which volume discretization can be easily performed.

It seems that an ideal approach for the treatment of volume integrals would be the one that has the accuracy and the efficiency of the cell-based direct integration methods and yet does not have the high cost associated with volume discretization**.**

In this chapter, we propose an accelerated surface discretization based BEM approach for non-homogeneous and nonlinear problems in 3-D complex domains. This approach uses the 3-D uniform grid built initially for the purpose of rapid evaluation of surface integrals as the baseline volume cells to perform volume integration. The only inputs required are the surface mesh and the uniform 3-D grid that encompasses/encloses the problem domain. The major advantage of our approach compared with the auxiliary domain method is that it is natural in our approach to use precorrected-FFT acceleration for all matrix-vector multiplication while it is hard to use acceleration techniques on the auxiliary domains.

The precorrected-FFT technique is described in Chapter 1.3. In the following sections, the proposed volume integration approach is presented first followed by a collection of numerical techniques for the implementation of this approach. Results for volume integrals evaluated on a simple geometry, namely a sphere, are presented first to illustrate the accuracy and efficiency of the proposed schemes. Then the formulation and the

iterative approach for solving nonlinear problems is described. 3-D nonlinear solver is developed using the proposed approach and tested using four partial differential equations.

The success of applying the BEM to nonlinear problems depends on several critical issues, the availability of the Green's functions for the linear operators, the accuracy and implementation of the precorrected-FFT method for the evaluation of volume integrals for far-field interactions, the accuracy and implementation of the direct volume integration for local interactions, the approximation for volume integrals over irregular boundary, and the integration of discrete source distribution function. Fortunately, for most linear engineering problems, the fundamental solution to the governing equation is available. In fact the Green's functions for potential problem, linear elastostatic and elastodynamic problems, Stokes flow and Helmholtz equations have been derived and published many years ago. These fundamental solutions can be directly applied to solve the coupled nonlinear fluid-structural problems. The current research will focus on the remaining issues as will be discussed in the following sections in detail.

3.2 PROPOSED VOLUME INTEGRATION APPROACH

The evaluation of the volume integral on the right-hand side of Equation (3.2) requires volume discretization of Ω if a standard cell-based direct integration scheme is employed. In our approach, we used the uniform 3-D FFT grid (see Figure 19), which was set initially for accelerating the surface integration, to perform the volume integration. To do so, let $\tilde{b}(\mathbf{y}) = b(\mathbf{y})$ ($b(u(\mathbf{y}),\mathbf{y})$ is simplified as $b(\mathbf{y})$ in the following discussions on volume integration) if $\mathbf{y} \in \Omega$ and $\tilde{b}(\mathbf{y}) = 0$ if $\mathbf{y} \in B \setminus \Omega$, where B is the

uniform grid. The volume integral in Equation (3.2) $\int_{\Omega} b(\mathbf{y})G(\mathbf{x},\mathbf{y})dv(\mathbf{y})$ can then be replaced by a volume integral performed on the grid, i.e., $\int_{B} \tilde{b}(\mathbf{y})G(\mathbf{x},\mathbf{y})dv(\mathbf{y})$. Since B consists of a set of cubes (see Figure 20), this integral is also equivalent to the sum of integrals evaluated on each cube, i.e.,

$$\int_{B} \tilde{b}(\mathbf{y})G(\mathbf{x},\mathbf{y})dv(\mathbf{y}) = \sum_{i} \int_{C_i} \tilde{b}(\mathbf{y})G(\mathbf{x},\mathbf{y})dv(\mathbf{y}) \tag{3.9}$$

where C_i is the i^{th} cube of B. Now the integration domain has been transferred from the complex problem domain (Ω) to a regular grid (B) consisting of a set of cubes. These cubes will serve as the coarse 'volume elements' for the volume integration.

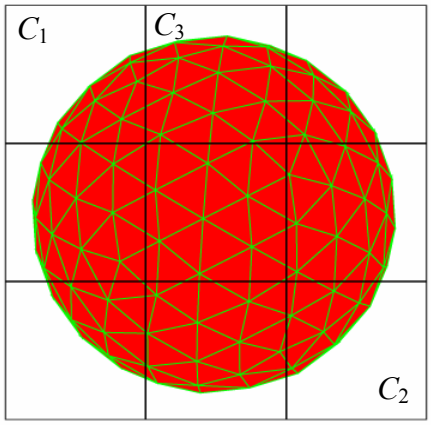


Figure 19. Side view of a sphere encompassed by a parallelpiped with a 3×3×3 array of cubes.

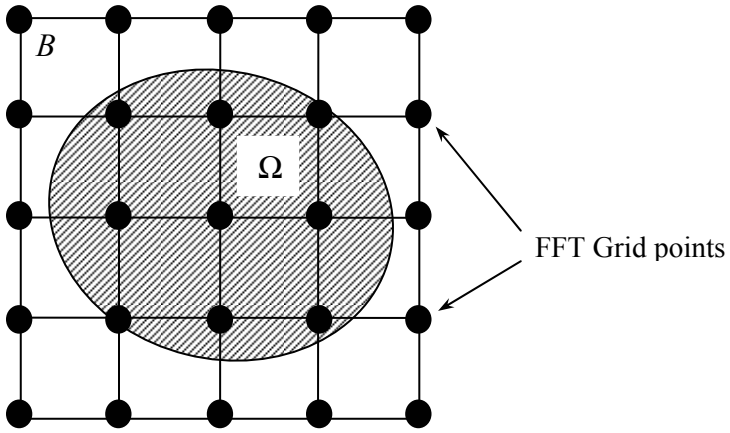


Figure 20. 2-D illustration of relationship between problem domain and the uniform FFT grid - Shaded area is the problem domain Ω, *B* is the uniform grid.

The general scheme for the implementation of the accelerated approach to evaluate volume integrals is similar to that of surface integrals [31]. A polynomial interpolation scheme is used to project sources inside cubes to the surrounding grid points as well as to interpolate values from grid points back to the evaluation points. For the convolution, the Fast Fourier Transform technique is used. The major difference between volume and surface integration, and the most challenging task, is the direct evaluation of volume integrals on the boundary cubes. Such calculations are needed in the direct calculation of nearby interactions and also in the projection and interpolation step.

3.2.1 Evaluation of $\int_{C_i} \tilde{b}(\mathbf{y})G(\mathbf{x},\mathbf{y})dv(\mathbf{y})$ for faraway interactions

If volume integrals present in the integral formulation are also of convolutional form (which is true for many engineering problems), the precorrected-FFT technique can be used to accelerate volume integration in addition to surface integration. For example, the

integral $\int_{C_j} \tilde{b}(\mathbf{y})G(\mathbf{x},\mathbf{y})dv(\mathbf{y})$ with evaluation point $\mathbf{x}$ located inside cube C_j can be approximated by

$$\int_{C_j} \tilde{b}(\mathbf{y})G(\mathbf{x},\mathbf{y})dv(\mathbf{y}) \approx \sum_{\mu} W_{\mu}(\int_{C_j} \tilde{b}(\mathbf{y})G(\mathbf{x}_{\mu},\mathbf{y})dv(\mathbf{y})) \tag{3.10}$$

in which $\mathbf{x}_{\mu}$ represents μ th grid point on cube C_j and W_{μ} is an interpolation operator that interpolates a function value at $\mathbf{x}$ based on values at surrounding grid points ($\mathbf{x}_{\mu}$).

The integral $\int_{C_j} \tilde{b}(\mathbf{y})G(\mathbf{x}_{\mu},\mathbf{y})dv(\mathbf{y})$ in Equation (3.10) can be further approximated as

$$\int_{C_j} \tilde{b}(\mathbf{y})G(\mathbf{x}_{\mu},\mathbf{y})dv(\mathbf{y}) \approx \sum_{\nu} G(\mathbf{x}_{\mu},\mathbf{y}_{\nu}) \cdot \tilde{b}(\mathbf{y}_{\nu}) = \sum_{\nu} G(\mathbf{x}_{\mu},\mathbf{y}_{\nu}) \cdot \int_{C_i} P_{\nu}(\tilde{b}(\mathbf{y}))dv(\mathbf{y}) . \tag{3.11}$$

In Equation (3.11), P_{ν} is a projection operator that projects the source term $\tilde{b}$ located at $\mathbf{y}$ onto the ν th grid point ($\mathbf{y}_{\nu}$) located on cube C_i . Thus the original integral $\int_{C_j} \tilde{b}(\mathbf{y})G(\mathbf{x},\mathbf{y})dv(\mathbf{y})$ is now replaced by

$$\int_{C_j} \tilde{b}(\mathbf{y})G(\mathbf{x}_{\mu},\mathbf{y})dv(\mathbf{y}) \approx \sum_{\mu} W_{\mu} \sum_{\nu} G(\mathbf{x}_{\mu},\mathbf{y}_{\nu}) \cdot \tilde{b}(\mathbf{y}_{\nu}) . \tag{3.12}$$

In summary, three major steps are involved in the evaluation of $\int_{C_j} \tilde{b}(\mathbf{y})G(\mathbf{x},\mathbf{y})dv(\mathbf{y})$:

- Projection: project $\tilde{b}(\mathbf{y})$ to produce $\tilde{b}(\mathbf{y}_{\nu})$ according to Equation (3.11).
- Convolution: evaluate $\sum_{\nu} G(\mathbf{x}_{\mu},\mathbf{y}_{\nu}) \cdot \tilde{b}(\mathbf{y}_{\nu})$ using FFT.
- Interpolation: obtain $\int_{C_i} \tilde{b}(\mathbf{y})G(\mathbf{x},\mathbf{y})dv(\mathbf{y})$ using values at surrounding grid points ($\int_{C_i} \tilde{b}(\mathbf{y})G(\mathbf{x}_{\mu},\mathbf{y})dv(\mathbf{y})$) by interpolation (Equation (3.10).

The interpolation operator is obtained by polynomial approximation. To interpolate the grid points to a point inside the cube, we assume

$$u = a_0 + a_1 x + a_2 y + a_3 z + \cdots + a_m f_m(x, y, z) \tag{3.13}$$

at each grid point, so $u = u_i$, $i = 0, 1, \ldots, m$. There are total $(m+1)$ grid points. Then

$$\begin{bmatrix} 1 & x_0 & y_0 & z_0 & \cdots & f_m(x_0, y_0, z_0) \\ & & \vdots & & \vdots & \\ 1 & x_m & y_m & z_m & \cdots & f_m(x_m, y_m, z_m) \end{bmatrix} \begin{Bmatrix} a_0 \\ \vdots \\ a_m \end{Bmatrix} = \begin{Bmatrix} u_0 \\ \vdots \\ u_m \end{Bmatrix}$$

or $\mathbf{A} \cdot \mathbf{a} = \mathbf{u}_g$ where $\mathbf{A}$ is a $(m+1)\times(m+1)$ matrix and $\mathbf{u}_g$ is the value of u at grid points. Now we have $\mathbf{a} = \mathbf{A}^{-1}\mathbf{u}_g$. By plugging it into Equation (3.13), any point inside the same cube can be expressed as

$$u = \underbrace{\begin{bmatrix} 1 & x & y & z & \cdots & f_m(x, y, z) \end{bmatrix}}_{shapefun} \cdot \mathbf{a} = shapefun \cdot \mathbf{A}^{-1} \cdot \mathbf{u}_g = w^T \cdot \mathbf{u}_g$$

Here *shapefun* is only determined by geometrical information so it is different for different points inside the cube but $\mathbf{A}^{-1}$ is only determined by geometrical information of the grid points so it is same for any points within the cube.

For projection step, it is proved by Phillips and White [31] that given w^{T} as an operator which interpolates potential at $(m+1)$ grid points onto charge coordinates, w may be interpreted as an operator which projects charge onto the grid coordinates, and vice versa. In either case, w^{T} and w have comparable accuracy. So if we have a point inside the cube, we can obtain $\mathbf{u}_g$ as follows:

$$\mathbf{u}_g = w \cdot \mathbf{u} = \left(\mathbf{A}^{-1}\right)^T \cdot shapefun \cdot \mathbf{u}\,.$$

Obviously the above procedure will produce good results only when the evaluation

cube C_j and the field cube C_i are reasonably far away as, for example, with the interactions between cubes C_1 and C_2 in Figure 19. When they are close, direct evaluation of $\int_{C_i} \tilde{b}(\mathbf{y})G(\mathbf{x},\mathbf{y})dv(\mathbf{y})$ is to be performed as, for example, with the interactions between cubes C_1 and C_3 in Figure 19.

3.2.2 Implementation schemes for nearby interactions

If the cube is totally inside the problem domain Ω, i.e., $\tilde{b}(\mathbf{y}) = b(\mathbf{y})$, regular Gauss quadrature may be employed to evaluate the volume integral if it is not singular. When the cube intersects with $\partial\Omega$ or when the integral is singular, special schemes must be developed to accurately and efficiently compute the integrals.

Consider a volume integral defined on a cube that intersects with the problem domain Ω_i. The integrand is nonzero inside Ω_i (shaded region shown in Figure 21(a)) and zero outside Ω_i. The only given information about Ω_i is the surface mesh (triangular panels shown in Figure 21 (b)). To evaluate this integral, a 'projection + transformation' scheme that uses only the existing surface elements has been developed. This scheme can also be readily extended to accurately evaluate singular integrals.

(a) (b)

Figure 21. (a) A partially filled boundary cube, Ω_i is the problem domain; (b) A partially filled cube with surface panels.

The 'projection + transformation' scheme

In the "projection + transformation" scheme, prisms are formed by projecting surface elements to one side of the cube. The projection direction is determined automatically in the code based on the consideration of efficiency. For example, the optimal projection strategy for the case shown in Figure 21 (b) is to project panels down to the bottom face of the cube. Further division of the prism along the projection direction is performed if necessary to maintain a good aspect ratio of prisms. These prisms serve as the fine "volume elements" and volume integration is performed on these elements. With addition of integrals over these prisms and the compensation volume (i.e., the volume which is not covered by prisms, for example, the darker(purple) region on the right in Figure 21 (b)), the desired volume integral on the cube can be obtained. For the case shown in Figure 21, the integral over Ω_i is the summation of the same integral evaluated on prisms and on the compensation volume.

To evaluate integrals on prisms, if the integral is nonsingular, a coordinate transformation is applied to transform prisms into regular domains as shown in Figure 22 and Gaussian quadrature is then employed to perform the integration. For a general prism, the irregular shaped prisms are transformed into a coordinate system composed by η_1, η_2, and η_3 using linear interpolation shape functions. The shape functions and Jacobian for this coordinate transformation are:

$$N_1 = \frac{1}{2}\eta_1(1+\eta_3), N_2 = \frac{1}{2}\eta_2(1+\eta_3), N_3 = \frac{1}{2}(1-\eta_1-\eta_2)(1+\eta_3),$$

$$N_4 = \frac{1}{2}\eta_1(1-\eta_3), N_5 = \frac{1}{2}\eta_2(1-\eta_3), N_6 = \frac{1}{2}(1-\eta_1-\eta_2)(1-\eta_3),$$

$$x_i = \sum_{\alpha=1}^{6} N_\alpha x_{i\alpha} \ , \ J_1 = \|\mathbf{J_1}\| = \left\| \left[\frac{\partial x_i}{\partial \xi_j} \right] \right\| .$$

Then numerical integration will then be performed on all three integrals.

$$\iiint_\Omega dv = \iiint_\Omega J_1 d\eta_1 d\eta_2 d\eta_3 .$$

The coordinates of the corners in different coordinate systems are given in Table 13.

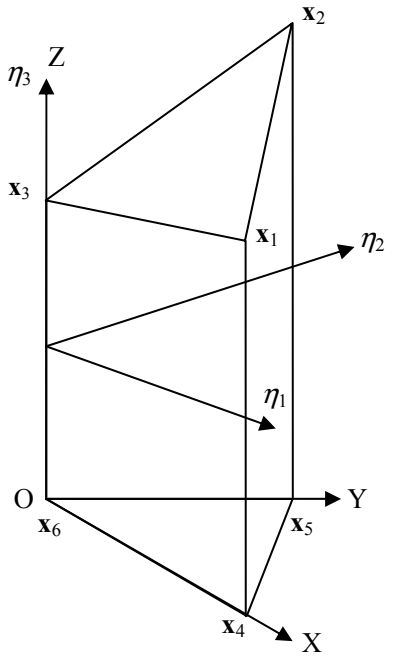


Figure 22. Transformation of a prism to a regular domain.

Table 13. Coordinates of the vertices in different coordinate systems

Corners of the prism	Coordinates in global coordinate system (x, y, z)	Coordinates in natural coordinate system (η_1, η_2, η_3)
$\mathbf{x}_1$	(1,0,1)	(1,0,1)
$\mathbf{x}_2$	(0,1,2)	(0,1,1)
$\mathbf{x}_3$	(0,0,1)	(0,0,1)
$\mathbf{x}_4$	(1,0,0)	(1,0,-1)
$\mathbf{x}_5$	(0,1,0)	(0,1,-1)
$\mathbf{x}_6$	(0 0 0)	(0 0 -1)

For singular integrals, i.e., the integrand approaches to infinity at a certain point (when $\mathbf{x} \in C_i$), the above scheme will not produce accurate results even when the Gauss points do not coincide with the singular point. To improve accuracy, the singularity is first removed from the integral before the Gaussian Quadrature is applied. This is achieved by performing an additional prism subdivision before transformation, and then applying a transformation whose Jacobian cancels the singularity. As illustrated in Figure 23, after projection, the singular prism is separated into three smaller prisms, each with the evaluation point as one of its vertices (Figure 23(a)). A transformation similar to the one described in [23] is then performed for each smaller prism (Figure 23(b)) to remove the singularity analytically and Gaussian quadrature is then applied to evaluate the resulting non-singular integrals on the regular domain shown in Figure 23(c).

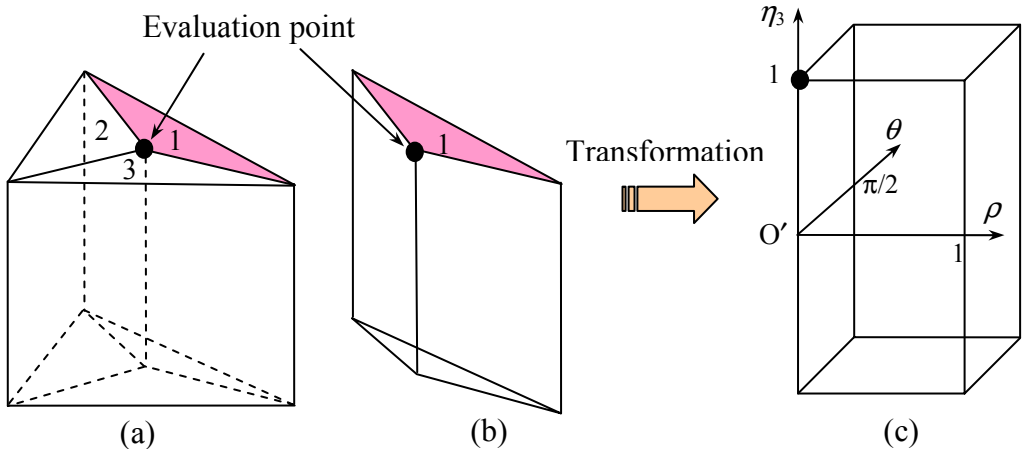


Figure 23. (a) A singular prism being divided into three smaller prisms; (b) A smaller prism with the evaluation point located on one of its vertices; (c) Coordinate transformation: prism 1 is transformed into a regular domain. Similar transformations can be done for prisms 2 and 3.

For the second coordinate transformation, the shape functions and the Jacobian are:

$$\eta_1 = \rho\cos^2(\theta), \eta_2 = \rho\sin^2(\theta), J_2 = \rho\sin(2\theta).$$

The volume integration will be performed like this

$$\iiint_\Omega dv = \iiint_\Omega J_1 d\eta_1 d\eta_2 d\eta_3 = \iiint_\Omega J_1 \cdot J_2 d\rho\, d\theta\, d\eta_3 .$$

Since

$$r = \sqrt{\eta_1^2 + \eta_2^2 + (\eta_3 - 1)^2} = \sqrt{\rho^2\left(\cos^4\theta + \sin^4\theta\right) + (\eta_3 - 1)^2},$$

we can remove the singularity in the volume integral analytically as following:

$$\begin{aligned}
\int_\Omega \frac{b}{r} dv &= \iiint \frac{b}{\sqrt{\eta_1^2 + \eta_2^2 + (\eta_3 - 1)^2}} d\eta_1 d\eta_2 d\eta_3 = \iiint \frac{b \cdot J_1}{\sqrt{\eta_1^2 + \eta_2^2 + (\eta_3 - 1)^2}} d\eta_1 d\eta_2 d\eta_3 \\
&= \iiint \frac{b \cdot J_1 \cdot J_2}{\sqrt{\rho^2\left(\cos^4\theta + \sin^4\theta\right) + (\eta_3 - 1)^2}} d\rho d\theta d\eta_3 \\
&= \iiint \frac{b \cdot J_1 \cdot \rho\sin(2\theta)}{\sqrt{\rho^2\left(\cos^4\theta + \sin^4\theta\right) + (\eta_3 - 1)^2}} d\rho d\theta d\eta_3 \\
&= \iiint \frac{b \cdot J_1 \cdot \sin(2\theta)}{\sqrt{\left(\cos^4\theta + \sin^4\theta\right) + (\eta_3 - 1)^2 / \rho^2}} d\rho d\theta d\eta_3
\end{aligned}$$

When $\rho \to 0$, $\eta_3 \to 0$, the limit $\lim_{\rho\to 0}\left((\eta_3 - 1)/\rho\right)$ exists so the volume integral is no longer singular and we can use regular Gaussian Quadrature to evaluate it.

Issues in the implementation of the 'projection + transformation' scheme

1. Overlapping error and panel truncation scheme

In the implementation of the "projection + transformation" scheme, one issue has arisen because the surface elements and the precorrected-FFT cubes can intersect with each other since the 3-D FFT grid structure is totally independent of the problem domain.

As shown in Figure 24, both surface panels Γ_1 and Γ_2 cross the cube boundaries. Based on the locations of their centroids (red dots in Figure 24), panel Γ_1 is assigned to cube C_2 and Γ_2 is assigned to cube C_4. When the projection directions in two neighboring cubes are different, it is possible to have a region on which the integration has been performed twice. For example, in Figure 24, if the panels in cubes C_1 and C_2 are projected down while panels in cube C_4 are projected to the left, the small region shaded by strips is double counted, which will cause errors (so-called "overlapping" errors) in the volume integration if no care is taken.

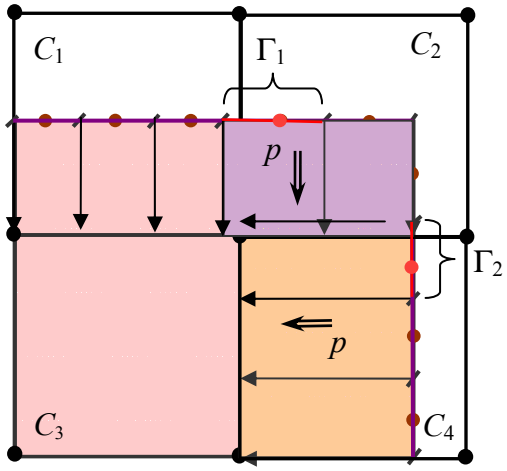


Figure 24. Illustration of scenario in which overlapping errors may occur.

The current strategy to deal with this issue is to truncate the panels that cross the cube boundaries (Figure 25) and to form a new set of surface panels. The projection is then performed on the new set of panels. This approach guarantees that in the new mesh, no panel will cross the cube boundaries and therefore the scenario described above will not occur. One drawback of such an approach is that it increases the number of panels. As

shown in Figure 25, one panel has become three panels after truncation. Nevertheless the increased number of panels will not dramatically increase the computational cost since the new mesh is only used for the evaluation of volume integrals on the boundary cubes. Integration on interior cubes does not need any mesh, and surface integration is still performed using the original surface mesh.

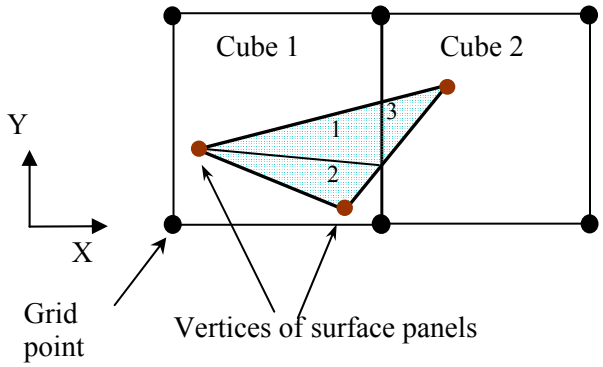


Figure 25. Panel truncation using cube boundaries.

For the original panels, the collocation points are the centroids of the panels since we are using constant panels. In other words, the evaluation points and source points are located on the centroid of the panels, so it is easy to identify the singular cases. But for the new set of surface panels generated from truncation of original panels, the evaluation points may not be the centroid of the new panels. To identify singular interactions, first all the children panels formed by truncation of an original panel bear the same index as the parent panel. Then for a given evaluation point (which is also the centroid of an original panel A) inside cube (j,k,l), the new panels inside cube (j,k,l) with same index as panel A are checked.Three angles in each new panel are calculated as shown in Figure 26.

If the sum of the three angles is 2π, then this panel is singular.

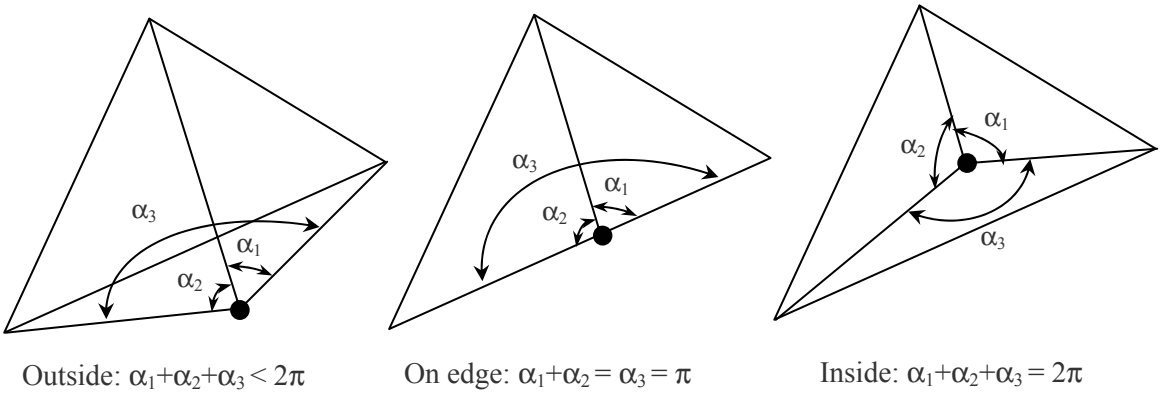


Figure 26. Use three angles formed by the evaluation point and three vertices of the panel to judge if a panel is singular or not.

2. Compensation volume

Another implementation issue is the identification of the compensation volume. As mentioned before, the compensation volume is part of the problem domain (Ω_i) that is not covered by the prisms formed from surface panels. This identification is accomplished by utilizing the new surface mesh to find the intersection curves of the problem domain and the cube as shown in Figure 27.

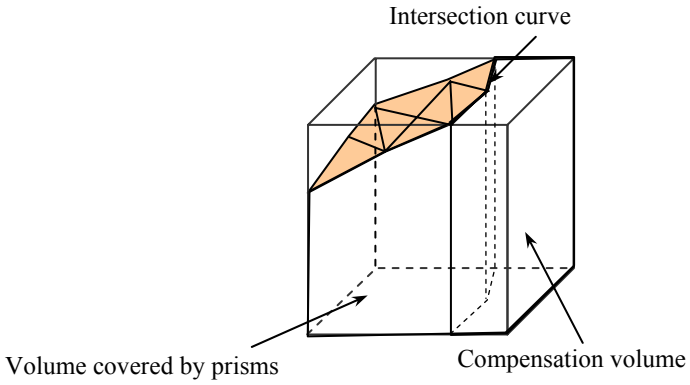


Figure 27. Illustration of the compensation volume.

Once the intersection curves are found, integration over the compensation volume is performed using a scheme illustrated in Figure 28. The intersection of the surface with the upper surface of the cube is typically irregular shape such as ADEF in Figure 28. The irregular shape is then projected from the upper surface of the cube to the bottom surface of the cube to form the compensation volume.

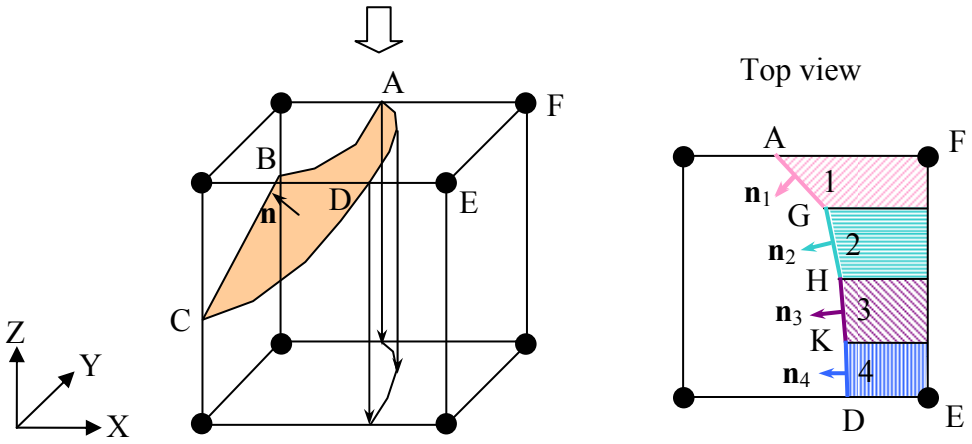


Figure 28. Compensation volume for irregular boundary in nearby interactions.

The intersection line segments, for example AG, GH, HK, KD in Figure 28, which are the boundary of new panels from truncation, are projected to the cube boundary according to their normal directions. Since the stripes are not rectangular, a 2D coordinate transformation is used in order to perform Gauss quadrature (Figure 29). Regular Gauss points are used in the z direction. The shape functions and Jacobian for the 2D coordinate transformation are expressed as the following:

$$N_1 = \frac{1}{4}(1+\eta_1)(1+\eta_2),\ N_2 = \frac{1}{4}(1-\eta_1)(1+\eta_2),\ N_3 = \frac{1}{4}(1-\eta_1)(1-\eta_2),$$

$$N_4 = \frac{1}{4}(1+\eta_1)(1-\eta_2),$$

$$x_i = \sum_{\alpha=1}^{6} N_\alpha x_{i\alpha}\ ,\ J_2 = \|\mathbf{J}_2\| = \left\| \left[\frac{\partial x_i}{\partial \eta_j} \right] \right\|.$$

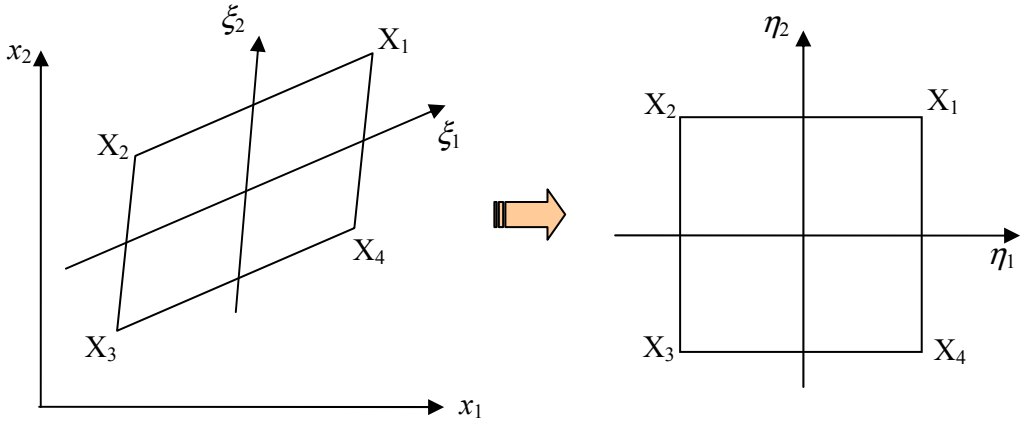
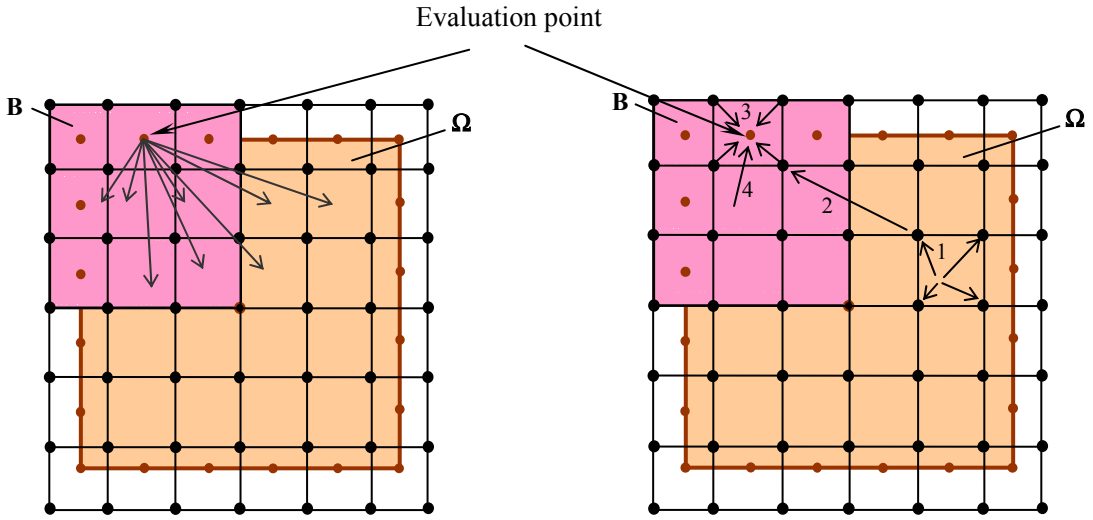


Figure 29. Coordinate transformation for a general quadrilateral cell.

3.2.3 Complexity of volume integration

The complexity analysis of volume integration based on the precorrected-FFT accelerated technique is similar to that of surface integration. In the direct approach (as shown in Figure 30(a)), for each evaluation point, m volume integrals need to be performed, where m represents the number of cubes. Thus the complexity of volume integration for n evaluation points is $O(nm)$. The precorrected-FFT accelerated approach illustrated in Figure 30(b) requires $O(m)+O(m\log m)+O(n)$ operations to perform the same integration. About $O(m)$ operations are used to perform projection. The convolution step takes about $O(m\log m)$ operations. Both interpolation and direct calculations for nearby interactions require $O(n)$ operations.



(a) (b)

Figure 30. (a) Illustration for direct approach; (b) Illustration for precorrected-FFT acceleration: 1) Projection, 2) Convolution, 3) Interpolation, 4) Nearby interactions.

When the number of evaluation points n which is related to the number of panels is larger than or equivalent to the number of cubes m (typically this is the case), acceleration in volume integration based on the precorrected-FFT technique is achieved.

3.3 Results For Volume Integration

3.3.1 Singular integrals – convergence study

The proposed prism subdivision combined with the 'projection + transformation' scheme for handling singular volume integrals was tested by integrating $1/\|\mathbf{x}-\mathbf{y}\|$ on the singular prism shown in Figure 23(b) with $\mathbf{x}$ being the evaluation point and $\mathbf{y}$ being the field point. The convergence plot is shown in Figure 31 together with the results obtained by applying Gauss quadrature directly without removing the singularity (direct Gauss

quadrature scheme). With order of 19, the result obtained by the proposed scheme converges to at least 4 digits while the direct Gauss quadrature scheme fails to converge even with the order of 100. This clearly indicates the influence of the singularity on the integration and the success of the proposed scheme for handling the singular integrals.

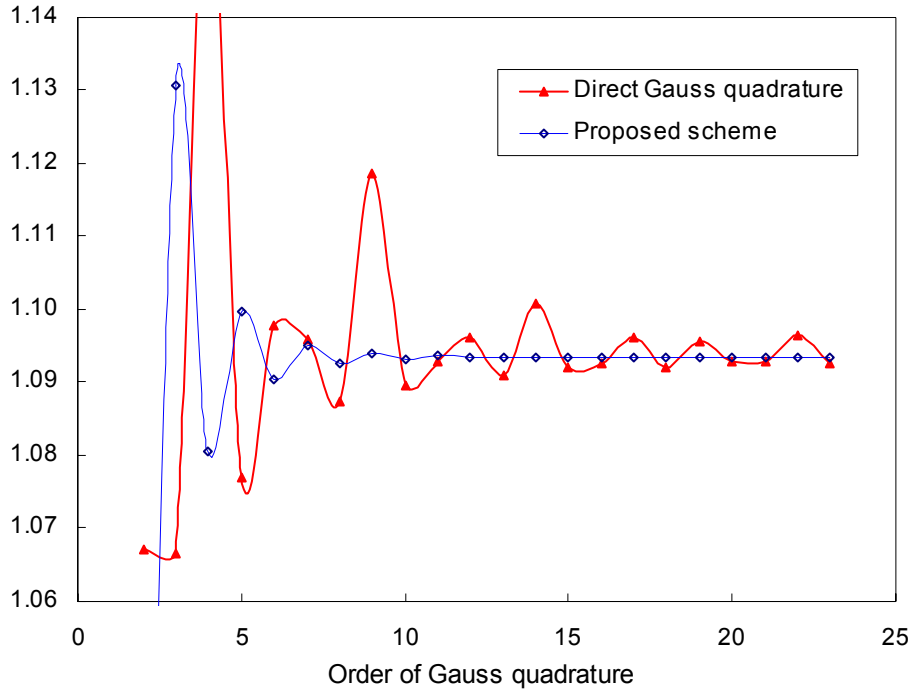


Figure 31. Convergence of evaluation schemes for singular volume integrals.

3.3.2 Non-singular integrals – direct method versus accelerated method

To study the accuracy of the precorrected-FFT accelerated technique and to establish the criteria for where we should apply approximation to evaluate volume integrals, a numerical study is carried out where two approaches are compared: direct evaluation using Gauss quadrature (Figure 32) and the FFT accelerated integration (Figure 33). The

integral tested is $\int_{C_i} \frac{b}{\|\mathbf{x}-\mathbf{y}\|} dv(\mathbf{y})$ with $b = 1$ and $b = x^2y$ respectively, evaluated on several field cubes. Both fully filled cubes, corresponding to the cases when the cubes are totally inside the problem domain, and partially filled cubes, corresponding to the cases when the cubes intersect with the domain boundary, are tested. The integrand is zero in some parts of partially filled cubes. In all the cases considered, the evaluation point is the same and is located at (–0.68449, –0.451469, –0.565426). To study the convergence of the error with respect to the distance of cube from the evaluation point, the normalized distance is calculated which is the minimum distance d from the evaluation point to the field cube divided by the length L of the cube (Figure 32). $O(2)$ Gauss quadrature is used in all cases (8 Gauss points in 3-D).

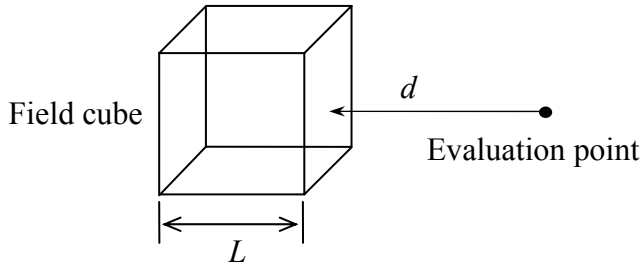
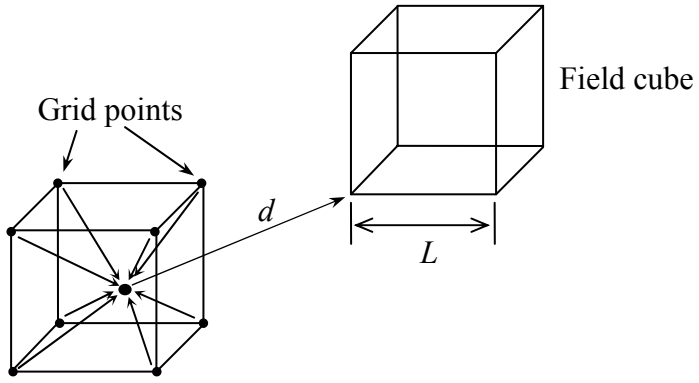


Figure 32. The minimum distance from the evaluation point to a cube.



*The value on the evaluation point is obtained through interpolation

Figure 33. Illustration of the FFT accelerated approach.

Figure 34 compares the results for the integral when $b = 1$ obtained from the direct Gauss quadrature approach and the approximation method using the FFT acceleration. The relative errors are plotted versus the normalized distances from the evaluation point to the cubes. Trend lines are added to illustrate that in both cases, the error reduces exponentially with the increase of normalized distance. When the normalized distance is greater than 1, the error is less than 0.3%. Thus it seems that we can use normalized distance 1 as the criterion for classifying nearby interactions and faraway interactions. In other words, nearby interactions should include only the nearest neighbors. The same conclusion can be drawn from the results for the case when $b = x^2y$ (Figure 35) where exponential reduction of error is also achieved with the increase of normalized distance.

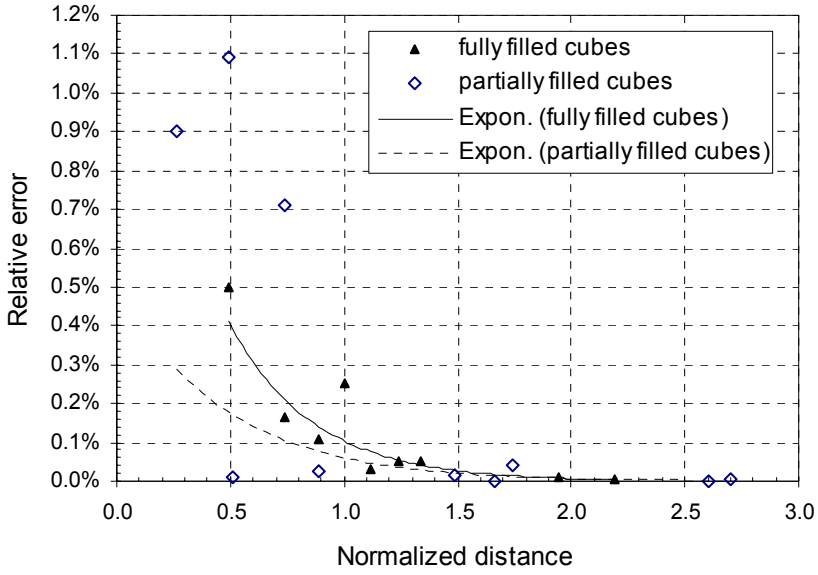


Figure 34. Convergence for the case $b = 1$.

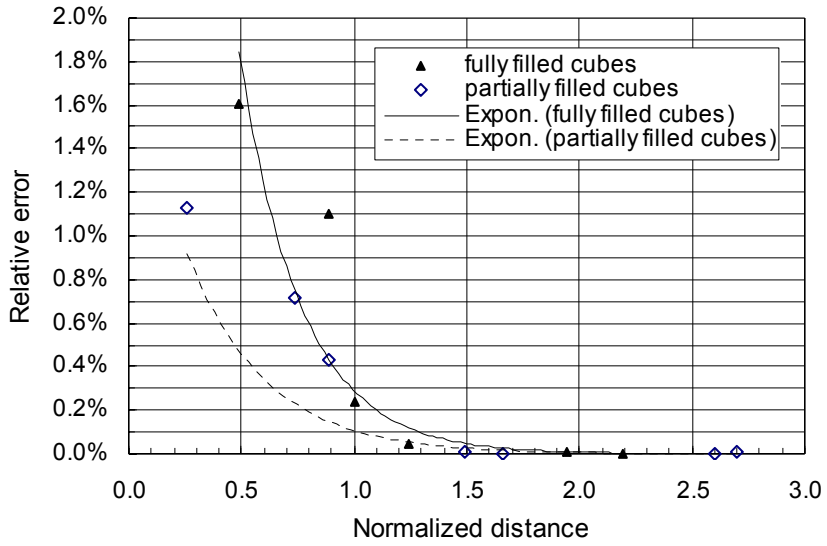


Figure 35. Convergence for the case when $b = x^2 y$.

By comparing Figure 34 and Figure 35, we can see that the performance of the approximation is problem-dependent. This can be explained by the complexity of the source term b. The error increases with the increase of complexity of b since we are using the same order of Gaussian quadrature. Generally speaking, higher-order Gauss quadrature should be used to achieve better accuracy of the integration for higher complexity of b.

3.3.3 The complete volume integral

The accuracy and efficiency of our accelerated volume-integration scheme were tested on $\int_{\Omega} \frac{1}{\|\mathbf{x}-\mathbf{y}\|} dv(\mathbf{y})$ for Ω a solid sphere with a radius of 1, shown in Figure 36. The evaluation point $\mathbf{x}$ is the centroid of the first surface panel in the data structure. Three approaches are compared:

I — Direct calculation using the "projection + transformation" scheme. Since there is no approximation, the results obtained by using this approach serve as the reference and errors are calculated based on it.

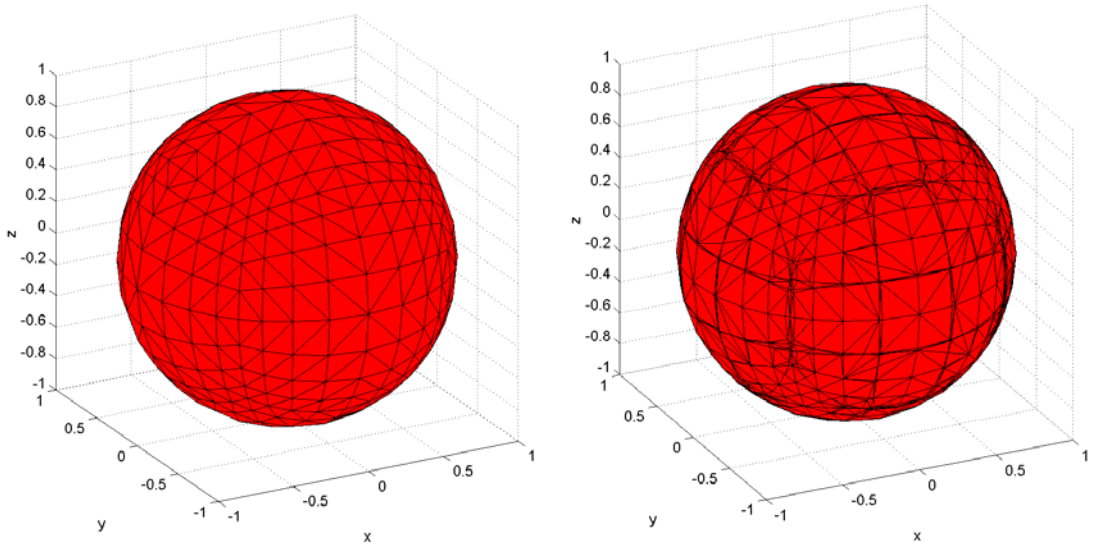
II & III — Acceleration for volume integration

II— Integration is performed directly over the singular cube and its nearest neighbors. Integration on all other cubes is performed approximately.

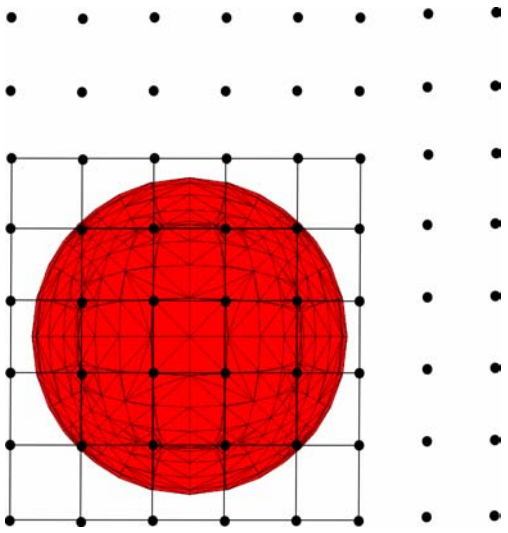
II(2) —Next nearest neighbors are also evaluated directly.

III— Only the singular cube is treated directly.

Three different surface discretizations were used, formed based on three original sets of surface panels with the total number of panels being 192, 768 and 3072 respectively. Figure 36 (a) shows one original set of 768 surface panels. Figure 36 (b) shows the corresponding mesh after truncation and Figure 36 (c) shows the FFT grid on the mesh in Figure 36 (b). Recall that the purpose in forming a new set of panels is to eliminate the overlapping errors. In the case of 192, no panels intersect with cubes; therefore the new set of panels is the same as the original one. The results of the integrals evaluated using the different schemes are listed in Table 14 and Table 15.



(a) (b)



(c)

Figure 36. A solid sphere domain: (a) original surface mesh (768 panels), (b) new surface mesh (2322 panels), (c) sphere with the 3-D uniform grid.

Table 14. Accuracy of different integration schemes.

Number of surface Panels	Evaluation Point (**x**)	Error (II–I)/I	Error (II(2)–I)/I	Error (III–I)/I
192	(-0.6868,-0.4396,-0.5508)	0.000%	0.000%	0.672%
768	(-0.6845,-0.4515,-0.5654)	0.010%	0.004%	0.476%
3072	(-0.7152,-0.6548,-0.2376)	0.011%	0.002%	0.651%

Table 15. CPU time of different schemes.

Number of surface panels	Ratio (II / I)	Ratio (II(2) / I)	Ratio (III / I)
192	144.03%	144.03%	21.25%
768	113.75%	335.05%	8.20%
3072	18.85%	47.05%	1.33%

From Table 14, it is clear that errors introduced by approximation are negligible particularly when approach II is used. In the case of 192, the error is zero when II was used and 0.672% when III was used. This is because in this case the total number of cubes is 8, i.e., 2 per side. Thus all the cubes are the nearest neighbors to each other and according to II, they are to be calculated directly. In III however, only the singular cube is evaluated directly, resulting in a different value from those obtained from I and II. In the cases of 768 and 3072, the number of cubes per side is five. The interactions can be classified into near-field and far-field for which approximations can be used. This is why there are differences between the results obtained by the different schemes.

Table 15 gives a comparison of the computational cost associated with volume integration of each method. All the calculations were performed on a PC with Pentium 4 1.8GHz processor. The savings in CPU time using acceleration is evident from the results,

and more efficiency is achieved when the discretization is finer, indicating better efficiency for large scale problems.

Comparing the results from II and II(2), the latter improved the accuracy slightly by increasing the amount of direct calculation, but at the price of more than doubled computational time. The reason is that when the number of cubes per side is 5, most interactions can be classified as nearby interactions. But since we still use FFT to approximate faraway interactions, the poor approximation for nearby interactions still needs to be precorrected which almost doubled the computational time. This also proved our conclusion in the last section, i.e., only the nearest neighbors should be included in nearby interactions to get good accuracy and efficiency.

Overall, the case study (i.e., $\int_{\Omega} \frac{1}{\|\mathbf{x}-\mathbf{y}\|} dv(\mathbf{y})$ on a solid sphere) shown in this section has demonstrated the accuracy and efficiency of the accelerated approach.

3.4 THE ITERATIVE APPROACH FOR NONLINEAR PROBLEMS AND THE FORMULATIONS

Consider a nonlinear problem in the form of

$$\Delta \mathbf{u} = b(\mathbf{u}, \mathbf{x}), \tag{3.14}$$

where b is function of $\mathbf{u}$ and $\mathbf{x}$. The equivalent integral formulation is

$$c(\mathbf{x})u(\mathbf{x}) + \int_{\partial\Omega} \frac{\partial G(\mathbf{x},\mathbf{y})}{\partial n(\mathbf{y})} u(\mathbf{y}) dS(\mathbf{y}) - \int_{\partial\Omega} G(\mathbf{x},\mathbf{y}) \frac{\partial u(\mathbf{y})}{\partial n(\mathbf{y})} dS(\mathbf{y}) = -\int_{\Omega} G(\mathbf{x},\mathbf{y}) b(u(\mathbf{y}),\mathbf{y}) dv(\mathbf{y}) \tag{3.15}$$

where

$$c(\mathbf{x}) = \begin{cases} 1, & \mathbf{x} \in \Omega \\ 0, & \mathbf{x} \notin (\Omega \cup \partial\Omega) \\ \alpha, & \mathbf{x} \in \partial\Omega \end{cases}.$$

In Equation (3.15), Ω is the domain of the problem with boundary $\partial\Omega$, $n(\mathbf{y})$ is the unit outward normal vector at the field point **y,** **x** is the evaluation point and $G(\mathbf{x},\mathbf{y})$ is the Green's function of the Laplace operator (Equation (3.2)).

Two approaches are available for solving Equation (3.15). The first approach is iteration in which starting values for **u** are assumed in the interior. The boundary integral equation is solved, and interior values for **u** at the next iteration step are determined using the boundary integral equation for interior domain and they are updated. The iteration is continued until a convergence criterion is satisfied. The second approach uses additional interior collocation to close the equation set. It has been reported in [16] that the iterative method is computationally more efficient for solving the non-homogeneous Helmholtz equation. However, the convergence of the iterative method is not guaranteed, especially when large portions of the boundary conditions are specified as Neumann-type. Nevertheless, for nonlinear problems in **u**, neither approach is guaranteed to converge. However, this approach has the advantage of speed and versatility, because potentially any function of **u** can be treated. The iterative approach is used in this work and good convergence is observed in the cases we tested, as shown in the results section. The flow of the nonlinear solver is shown below.

1. Input Dirichlet boundary condition to $\mathbf{u}_{\partial\Omega}$.
2. Compute all the Gauss points used for volume integration and set initial guess for $\mathbf{u}_{\Omega}{}^{0}$ to be zero. Calculate b^0.
3. Start iteration for nonlinear solver. k represents the number of iterations.

   Step 1: Use $\mathbf{u}_{\partial\Omega}$ and b^k to calculate $\mathbf{q}^k$.

   Step 2: Use $\mathbf{u}_{\partial\Omega}$ and $\mathbf{q}^k$ to calculate $\mathbf{u}^{k+1}$ and thus b^{k+1}.

   Step 3: Use $\mathbf{u}_{\partial\Omega}$ and b^{k+1} to calculate $\mathbf{q}^{k+1}$.

If $|\mathbf{q}^{k+1} - \mathbf{q}^k| < 10^{-3}$, then stop. Else $k = k+1$.

End.

4. Calculate L_2-norm error for $\mathbf{q} = |\mathbf{q}^{k+1} - \mathbf{q}^k|$.

The corresponding boundary integral equations for the above steps are given in Equations (3.16~3.18). Here F is the conjugate kernel of G.

Step 1:

$$\int_{\partial\Omega} G(\mathbf{x},\mathbf{y}) q^k(\mathbf{y}) ds(\mathbf{y}) = -\frac{u_{\partial\Omega}(\mathbf{x})}{2} + \int_{\partial\Omega}^{cpv} F(\mathbf{x},\mathbf{y}) u_{\partial\Omega}(\mathbf{y}) ds(\mathbf{y}) - \int_{\Omega} G(\mathbf{x},\mathbf{y}) b^k(u^k(\mathbf{y}),\mathbf{y}) dv(\mathbf{y}) \quad (3.16)$$

Step 2:

$$u^{k+1}(\mathbf{x}) = -\int_{\partial\Omega} G(\mathbf{x},\mathbf{y}) q^k(\mathbf{y}) ds(\mathbf{y}) + \int_{\partial\Omega} F(\mathbf{x},\mathbf{y}) u_{\partial\Omega}(\mathbf{y}) ds(\mathbf{y}) - \int_{\Omega} G(\mathbf{x},\mathbf{y}) b^k(u^k(\mathbf{y}),\mathbf{y}) dv(\mathbf{y}) \quad (3.17)$$

Step 3:

$$\int_{\partial\Omega} G(\mathbf{x},\mathbf{y}) q^{k+1}(\mathbf{y}) ds(\mathbf{y}) = -\frac{u_{\partial\Omega}(\mathbf{x})}{2} + \int_{\partial\Omega}^{cpv} F(\mathbf{x},\mathbf{y}) u_{\partial\Omega}(\mathbf{y}) ds(\mathbf{y}) - \int_{\Omega} G(\mathbf{x},\mathbf{y}) b^{k+1}(u^{k+1}(\mathbf{y}),\mathbf{y}) dv(\mathbf{y}) \quad (3.18)$$

To speed up the volume integration, the Gauss points needed for the evaluation of the volume integrals in Equations (3.16) and (3.18) are saved so we do not need to calculate their position repeatedly. The second step in the above iterative procedures with BIE, shown in Equation (3.17), is to update the **u** values at the saved Gauss points. However, the volume integral is singular when the evaluation point coincides with the source point. The singularity involved in this volume integral $\int_{\Omega} G(\mathbf{x},\mathbf{y}) b^k(\mathbf{y}) dv(\mathbf{y})$ is removed analytically by separating the integrand into two parts as shown in Equation (3.19).

$$\begin{aligned} \int_{\Omega} G(\mathbf{x},\mathbf{y}) b^k(\mathbf{y}) dv(\mathbf{y}) &= \int_{\Omega} G(\mathbf{x},\mathbf{y}) \left(b^k(\mathbf{y}) - b^k(\mathbf{y}_P) \right) dv(\mathbf{y}) + \int_{\Omega} G(\mathbf{x},\mathbf{y}) b^k(\mathbf{y}_P) dv(\mathbf{y}) \\ &= \int_{\Omega} \frac{b^k(\mathbf{y}) - b^k(\mathbf{y}_P)}{4\pi r} dv(\mathbf{y}) + \frac{b^k(\mathbf{y}_P)}{4\pi} \int_{\Omega} \frac{1}{r} dv(\mathbf{y}). \end{aligned} \quad (3.19)$$

Now the first integral on the right-hand-side of Equation (3.19) is no longer singular and can be integrated numerically. The second volume integral on the right-hand-side of Equation (3.19) is evaluated by subdividing the volume using the singular point and applying Gauss quadrature on the new volume elements (Figure 37). A singular prism is subdivided into 6 small prisms while an interior cube is subdivided into 8 small prisms. By letting the singular source point be the vertex of small prisms, the singularity can be removed explicitly as described in Chapter 3.2.2. Since the integrand is $1/r$ now, we avoided the evaluation of b values at the new Gauss points and the value of $\int_{\Omega} \frac{1}{r} dv(\mathbf{y})$ at each inside Gauss point can be computed before the iteration.

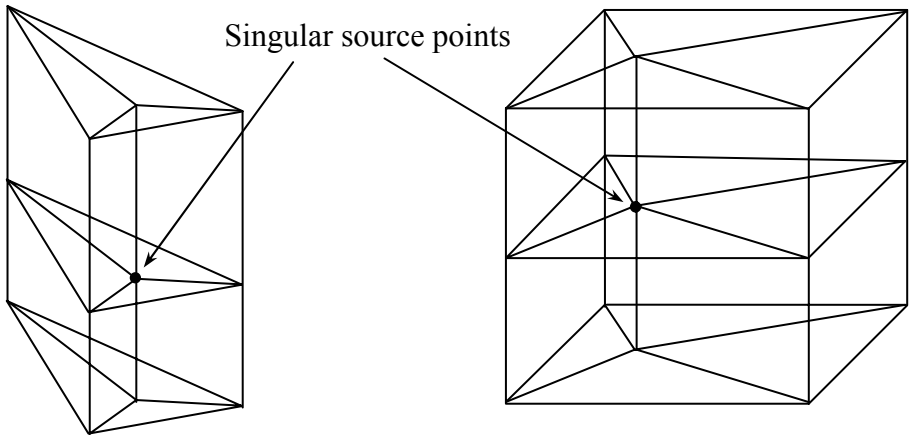


Figure 37. Subdivision of singular prism and singular cube.

Similar to the FastSlipStokes solver, the integral equations shown in Equations (3.16) and (3.18) were solved using a boundary element method together with the precorrected-FFT acceleration technique. For simplicity, a piece-wise constant collocation scheme is

used to discretize the integral equation. The surface of the structure is discretized into small panels. On each panel, $\mathbf{u}$ and $\partial\mathbf{u}/\partial n$ are assumed to be constant. A system of equations for the panel unknowns is then derived by insisting that the integral equations be satisfied at each panel centroid. The results are linear systems which relate the known quantities ($\mathbf{u}_{\partial\Omega}$, $\mathbf{u}^k$, b^k, $\mathbf{u}^{k+1}$ and b^{k+1} in Equations (3.16) and (3.18)) to the unknown quantities ($\mathbf{q}^k = -\partial\mathbf{u}/\partial n$ in Equations (3.16) and (3.18)), as in Equation (3.20):

$$\mathbf{G}\cdot\begin{Bmatrix} q_1^{\,k} \\ q_2^{\,k} \\ \vdots \\ q_n^{\,k} \end{Bmatrix} = -\frac{1}{2}\begin{Bmatrix} u_{\partial\Omega 1} \\ u_{\partial\Omega 2} \\ \vdots \\ u_{\partial\Omega n} \end{Bmatrix} + \mathbf{F}\cdot\begin{Bmatrix} u_{\partial\Omega 1} \\ u_{\partial\Omega 2} \\ \vdots \\ u_{\partial\Omega n} \end{Bmatrix} + \begin{Bmatrix} d_1^{\,k} \\ d_2^{\,k} \\ \vdots \\ d_n^{\,k} \end{Bmatrix}. \tag{3.20}$$

In Equation (3.20), the load vector $\mathbf{d}^k$ consists of the corresponding volume integrals shown in Equations (3.16) and (3.18). The linear system in Equation (3.20) is solved using GMRES [34] and the procedure is accelerated using the precorrected-FFT technique.

Step 2 shown in Equation (3.17) can be expressed as Equation (3.21) and it is also accelerated using the precorrected-FFT technique.

$$\begin{Bmatrix} u_1^{\,k+1} \\ u_2^{\,k+1} \\ \vdots \\ u_n^{\,k+1} \end{Bmatrix} = -\mathbf{G}\cdot\begin{Bmatrix} q_1^{\,k} \\ q_2^{\,k} \\ \vdots \\ q_n^{\,k} \end{Bmatrix} + \mathbf{F}\cdot\begin{Bmatrix} u_{\partial\Omega 1} \\ u_{\partial\Omega 2} \\ \vdots \\ u_{\partial\Omega n} \end{Bmatrix} + \begin{Bmatrix} d_1^{\,k} \\ d_2^{\,k} \\ \vdots \\ d_n^{\,k} \end{Bmatrix}. \tag{3.21}$$

3.5 BENCHMARK PROBLEMS

Based on our proposed volume integration scheme, 3-D BEM solvers have been developed and used for solving for several benchmark problems. The solvers use a

uniform 3-D grid (FFT grid) as the coarse mesh to perform volume integration. Same grid is also used to accelerate the evaluation of both surface and volume integrals. The accuracy and efficiency of the solver are demonstrated through several examples described in the following sections. In all cases, Dirichlet boundary conditions are prescribed on all surfaces of the domain. The solutions found through simulations are the normal flux at the boundaries. Results are compared with the exact solutions and errors measured using the L_2-norm as shown below are reported.

$$Error = \sqrt{\sum_{i=1}^{N}\left(\frac{\partial u^s}{\partial n} - \frac{\partial u^e}{\partial n}\right)^2 ds_i} \Bigg/ \sqrt{\int\left(\frac{\partial u^e}{\partial n}\right)^2 ds},$$

where superscript *s* indicates simulation and *e* represents the exact solution.

For Poisson problems, since the *b* function does not include variable *u*, no iteration is needed to solve for *q*. For Helmholtz and nonlinear problems, the dependent variable *u* is initially assumed to be 0 in the interior of the domain. Following the three steps for an iterative solver, new interior values of *u* are calculated using Equation (3.17) and *b* is updated accordingly. The iteration continues until the difference ε between the L_2-norm of two successive boundary values of *q* at boundary collocation points is less than 10^{-3}.

$$\varepsilon = \sqrt{\sum_{i=1}^{N}\left(\frac{\partial u^{k+1}}{\partial n} - \frac{\partial u^{k}}{\partial n}\right)^2 ds_i} \Bigg/ \sqrt{\int\left(\frac{\partial u^{k+1}}{\partial n}\right)^2 ds} \quad < 10^{-3}.$$

Table 16 shows the number of volume elements generated by the projection scheme for the surface discretizations used in the simulations in following sections. Here ‘s’ is the sphere domain shown in Figure 36 and ‘e2s’ is the ellipsoid with two spherical exclusions domain shown in Figure 42.

Table 16. Number of volume elements

	Ellipsoid-2 spheres			Sphere			
	e2s1	e2s2	e2s3	s0	s1	s2	s3
Number of origianl panels	288	1152	3456	48	192	768	3072
Number of new panels	976	2480	5920	48	192	1776	4848
Number of volume elements	1172	3438	8414	60	260	1976	9532

3.5.1 Poisson problems

For Poisson problems, since there is no unknown function involved in the right-hand-side of the equation, we do not need any iteration. We can use the boundary conditions and the value of function *b* at interior points to obtain the solution *q* on the boundary.

3.5.1.1. Problem 1: $\nabla^2 u = 1$ *on a solid sphere*

We want to use this simple problem to test the overall performance of the 3-D Poisson solver which contains both surface and volume integrals. The problem solved in this case study is $\nabla^2 u = 1$ with the boundary conditions of $u = x^2/2$. The analytic solution for the normal flux at the boundary is $\frac{\partial u}{\partial n} = x \cdot n_x$ where n_x is the *x*-component of the outward normal vector. The solid sphere with radius 1 is shown in Figure 36.

Similar to the previous study on volume integration, four approaches are compared:

I — Direct calculation for both surface and volume integration, uses only the original surface mesh (Figure 36 (a));

II — Acceleration for surface integration only, uses original surface mesh (Figure 36 (a)) for surface integration and new mesh (Figure 36 (b)) for volume integration;

III & IV—Acceleration for both surface and volume integration, uses original surface mesh (Figure 36 (a)) for surface integration and new mesh for volume

integration (Figure 36 (b));

III — Integration on the singular cube and its nearest neighbors is performed directly. Integration on all other cubes is performed approximately;

IV — Only the singular cube is treated directly.

The errors in the normal flux $\partial u/\partial n$ at the boundaries are plotted in Figure 38. There are no data points corresponding to the original discretization of 3072 for approaches I and II due to exhaustion of machine memory. It is interesting to see that errors in approaches I, II and III are very close. This further confirms our previous observation that errors caused by acceleration are negligible. The red dotted line in Figure 38 is for convergence rate $O(h)$. Approaches I, II and III achieved convergence rate $O(h)$. This shows that with the precorrected-FFT acceleration, we can still achieve good convergence rate. Approach IV, on the other hand, fails to converge in this example. Despite the great savings in CPU time, approach IV is therefore not recommended.

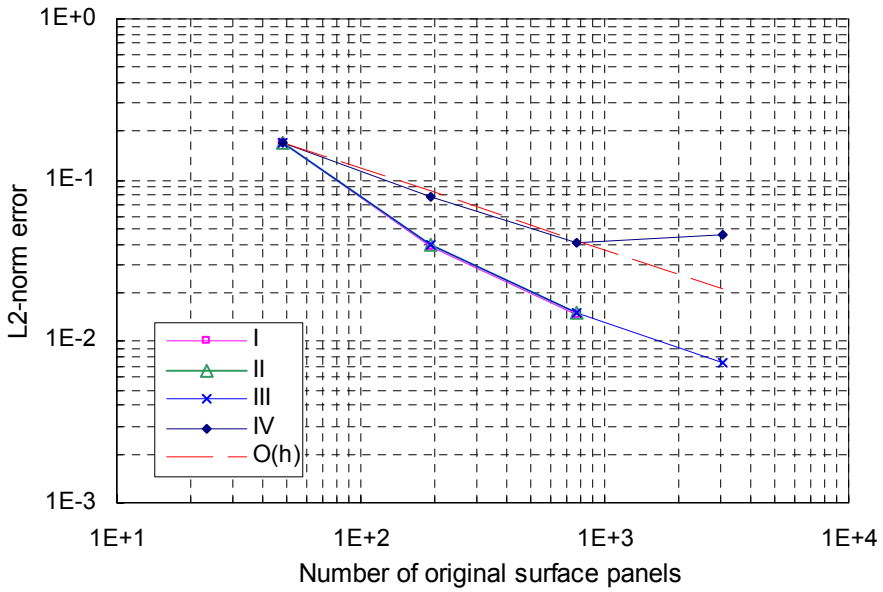


Figure 38. L_2-norm errors in $\partial u/\partial n$ of Poisson problem 1.

Since the accuracy of the Poisson solver is determined by the accuracy of volume integration as well as the surface integration, we would like to examine the relative contributions of errors coming from surface integration and volume integration. Figure 39 shows the error plot for the Laplace problem $\nabla^2 u = 0$ (only results from the accelerated approach are plotted). Although the error is not directly comparable, we can still make a qualitative comparison. The error for the Laplace solver is solely from surface integration and it is comparable with the error for the Poisson problem. So the error of the Poisson solver seems to come mostly from the surface integration. It is worth pointing out that in all the calculations, the order of Gauss quadrature formula used is 3. Obviously the accuracy will further improve when a higher order of Gauss quadrature formula is used, but the CPU time will also increase too.

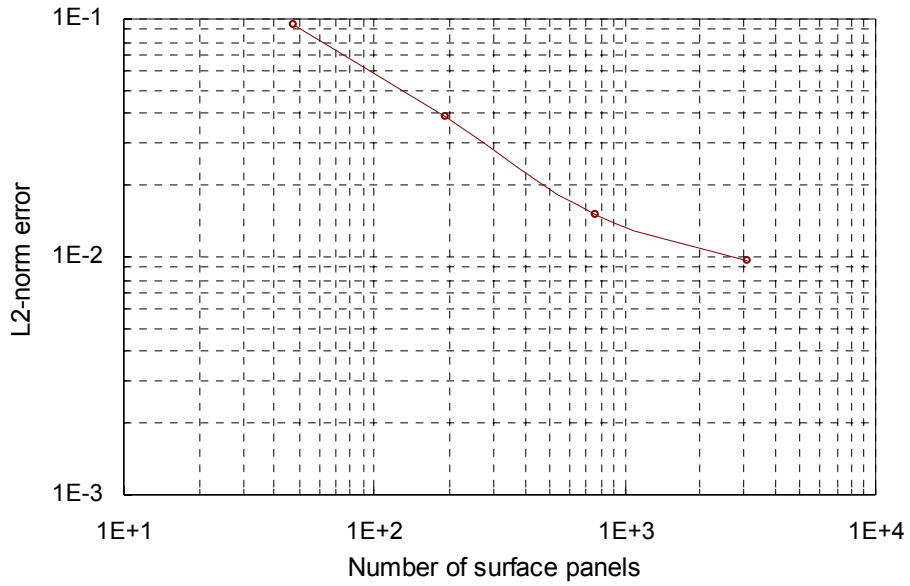


Figure 39. L_2-norm errors in $\partial u/\partial n$ of the Laplace problem.

Figure 40 gives the CPU time used for solving the Poisson problem using different approaches. Recall that the only input to the code is surface mesh. Thus the recorded CPU time includes the time for setting up the FFT grid, generating volume mesh, etc. The data points at 3072 surface panel discretization for approaches I and II are extrapolated. At coarse discretizations particularly in the cases of 48 and 192, the accelerated schemes are more costly. This is because the additional overhead needed for generating new meshes, performing projection, convolution and interpolation offsets or even overpowers the time saved by the acceleration. When the discretization becomes finer, the savings by the acceleration pick up, resulting in much less CPU time cost. Thus, the acceleration scheme is, as expected, more efficient for large-scale problems. From Figure 41, we can see that the efficiency of our proposed approach is somewhere between $O(n^2)$ and $O(n)$.

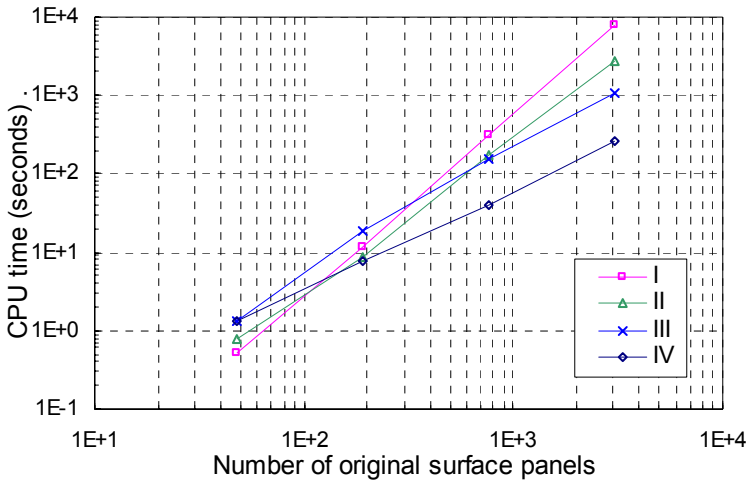


Figure 40. CPU time for solving the Poisson problem 1 on a spherical domain.

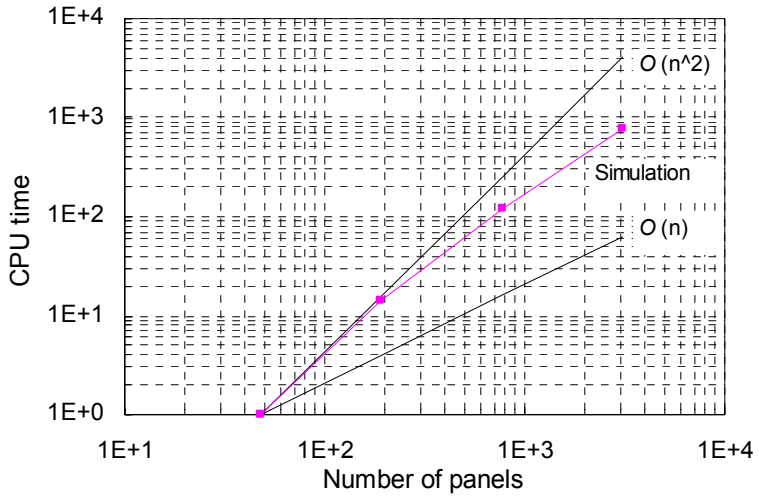


Figure 41. CPU time plot for approach III

3.5.1.2. Problem 2: $\nabla^2 u = (2y^3 + 6y)e^{(x+z)}$

This problem is the benchmark problem 1 studied in [15]. The 3-D solver for this problem is tested on two domains, the solid sphere shown in Figure 36 (referred to domain 1 hereafter) and a solid ellipsoid with two spherical exclusions (see Figure 42, referred to as domain 2 hereafter). In domain 2, the lengths of the semi-axes of the solid ellipsoid are a = 2, b = 4, c = 2, and the diameter for the spheres is 1 with centers at (0, −1,0) and (0,1,0) respectively.

One solution to this problem is shown in Equation (3.22). Again, u is used to prescribe the Dirichlet boundary conditions at all surfaces (the ellipsoid and two spheres). The expression for q serves as the reference for the calculation of errors (measured in L_2-norm) in the simulation.

$$u = y^3 e^{(x+z)},\ q = -\frac{\partial u}{\partial n} = -e^{(x+z)}(y^3 n_x + 3y^2 n_y + y^3 n_z) \tag{3.22}$$

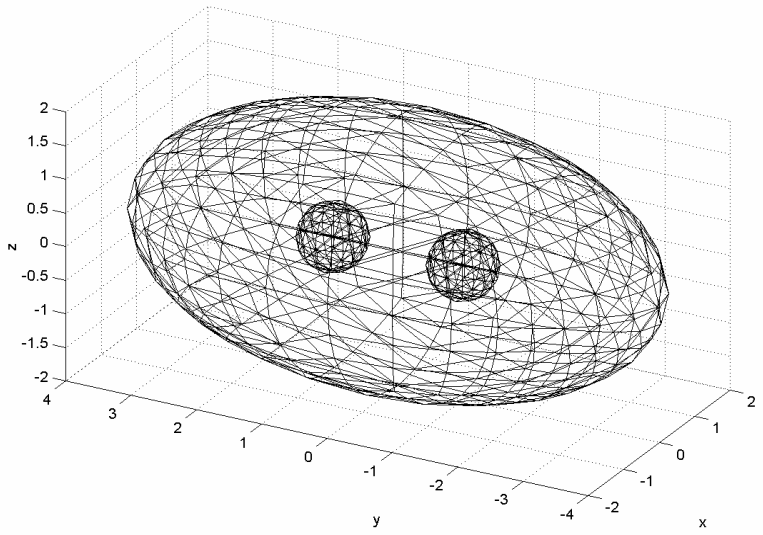


Figure 42. Surface mesh of domain 2 - A solid ellipsoid with two spherical exclusions. The total number of surface panels shown is 1152.

In Figure 43 and Figure 44, L_2-norm error and CPU time are plotted as functions of the number of surface panels. For domain 2, three discretizations were tested (one is shown in Figure 42). With a total of 3456 constant surface elements (3072 for the ellipsoid plus 192 for each sphere), the error is reduced to 0.92% and the associated overall CPU time is 1029 seconds. The reported error for a similar problem in [15] is around 2.3% and the CPU time is 750 seconds using the auxiliary domain method (ADM) (The authors have shown in [15] that the ADM performs better than the Dual Reciprocity Method and the classical domain discretization method for this particular case). Although it is difficult to directly compare the performance of our approach and the ADM based on these results, different computers, different boundary conditions, etc, nevertheless these results indicate that our approach is accurate and efficient.

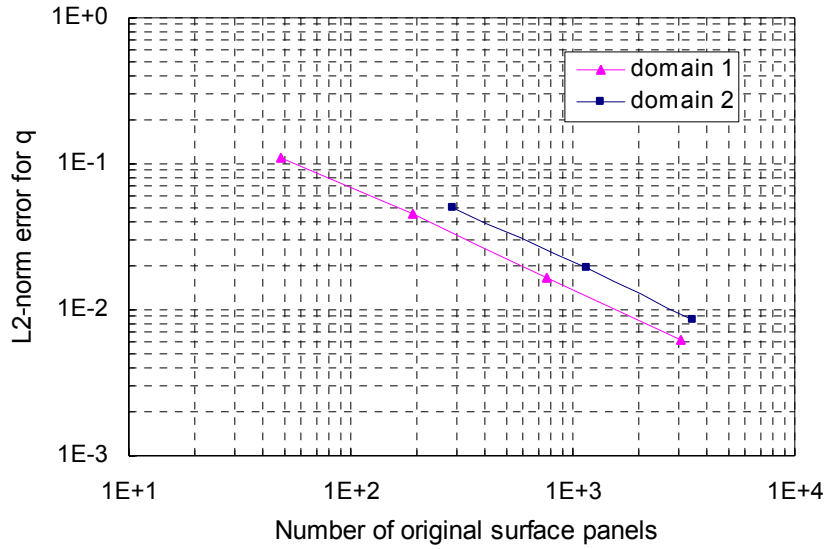


Figure 43. L_2-norm error as function of surface discretization for Poisson problem 2 on two domains.

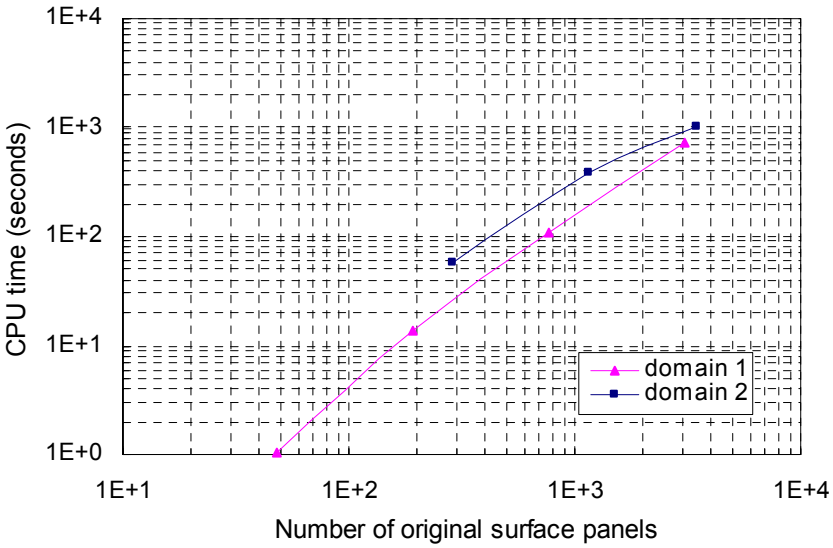


Figure 44. CPU time as function of surface discretization for Poisson problem 2 on two domains.

Also from Figure 43 and Figure 44, we notice that the solver achieved better accuracy on domain 1 in less CPU time. The reason is that domain 2 is larger than domain 1 in size.

Therefore, for same number of surface elements, lower accuracy of surface integration is expected. More cubes is used for domain 2 which results in a longer CPU time.

3.5.2 Helmholtz problem and nonlinear problem

3.5.2.1. Problem 3: *Helmholtz equation* $\nabla^2 u = -u + h(x, y, z)$

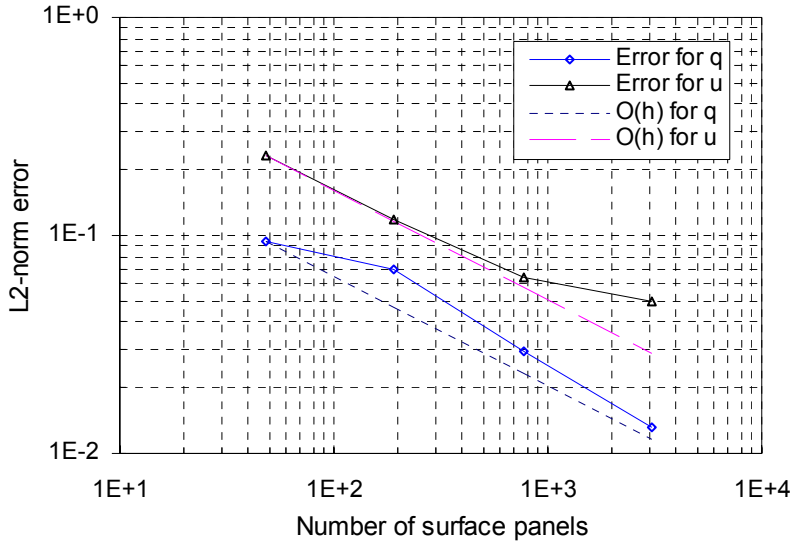
This problem is also chosen from the paper written by Hsiao *et al.* [15] so that we can compare the efficiency and accuracy of our solver. The analytical solution for this equation is shown in Equation (3.23). As before, u is used to apply boundary conditions, and q is used to obtain the L_2-norm error.

$$\begin{aligned} &u = 3x^3y + 2x^2y^2 - xy^3 + z, \quad h = 4x^2 + 4y^2 + 12xy + 3x^3y + 2x^2y^2 - xy^3 + z, \\ &q = -\frac{\partial u}{\partial n} = -(9x^2y + 4xy^2 - y^3)n_x - (3x^3 + 4x^2y - 3xy^2)n_y - n_z. \end{aligned} \tag{3.23}$$

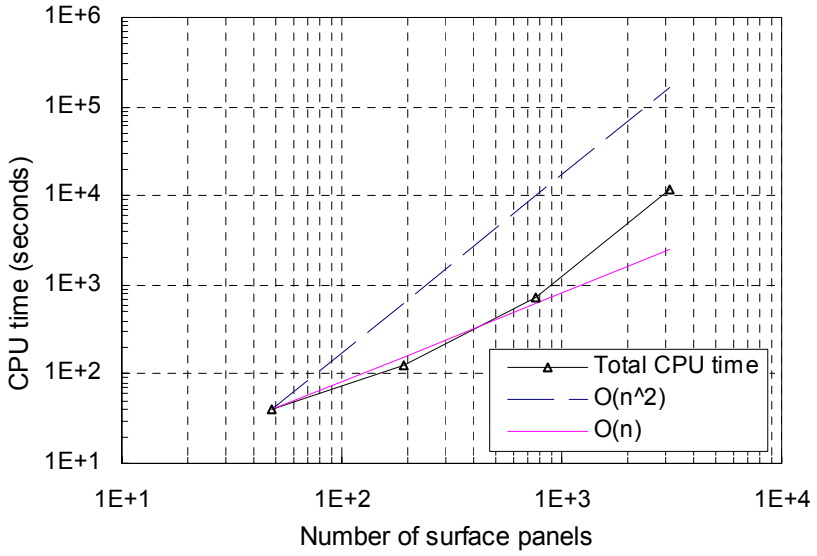
The two domains used for problem 2 are tested on this problem. The results on the solid sphere domain, domain 1, are shown in Table 17 and Figure 45. The results on the 3-D multiply-connected domain, domain 2, are shown in Table 18 and Figure 46. For this problem, it took 3 iterations for domain 1 to converge for all discretizations and 5 iterations for domain 2 to converge for all discretizations.

Table 17. Results for the Helmholtz problem on domain 1.

Number of surface panels	CPU time for each iteration (seconds)	Total CPU time (seconds)	L_2-norm error for q	L_2-norm error for u
48	10.13	39.39	9.36%	23.04%
192	26.78	123.859	6.93%	11.79%
768	111.91	733.734	2.93%	6.44%
3072	1072.07	11896.1	1.31%	4.99%



(a)

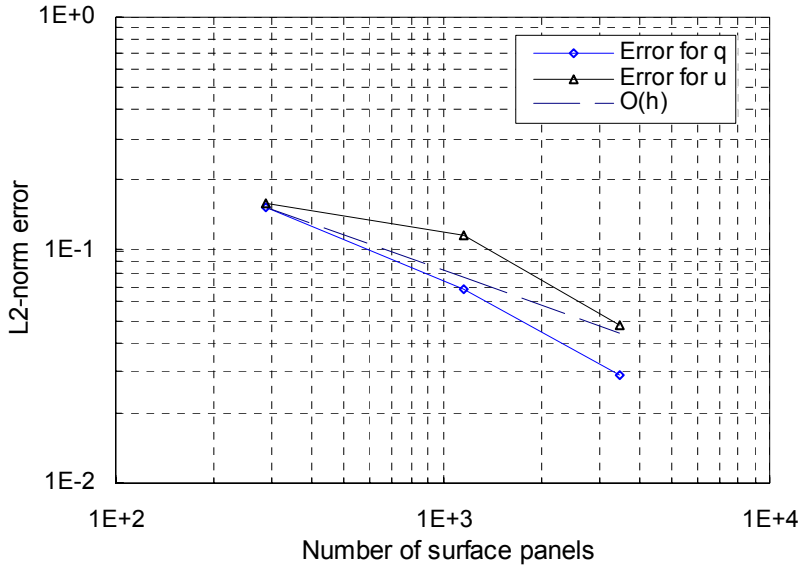


(b)

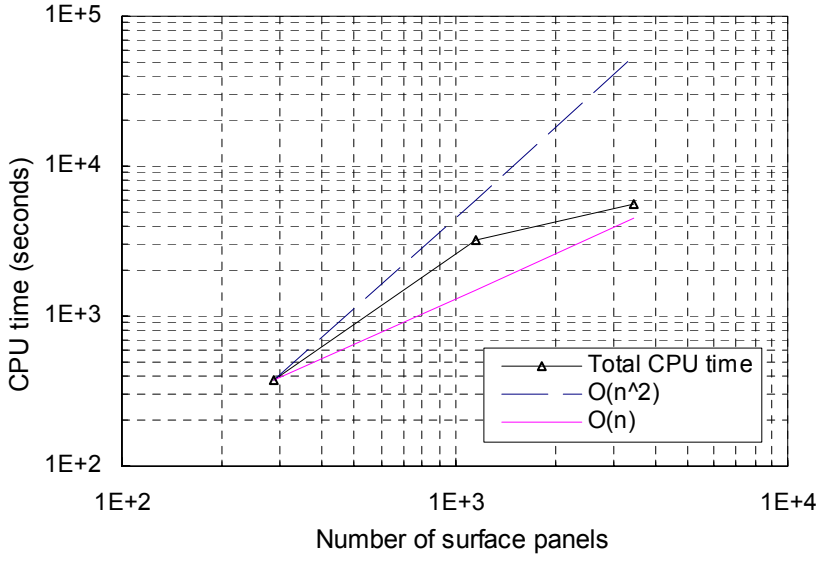
Figure 45. Results for the Helmholtz problem on domain 1: (a) L_2-norm errors, (b) CPU time.

Table 18. Results for the Helmholtz problem on domain 2.

Number of surface panels	CPU time for each iteration (seconds)	Total CPU time (seconds)	L_2-norm error for q	L_2-norm error for u
288	58.99	376.39	15.29%	15.79%
1152	367.66	3235.14	6.84%	11.61%
3456	717.25	5578.73	2.91%	4.81%



(a)



(b)

Figure 46. Results for the Helmholtz problem on domain 2: (a) L_2-norm errors, (b) CPU time.

From the results for the Helmholtz problem, good convergence is observed with the refinement of surface discretization in all cases. The convergence rates for u and q are of $O(h)$ for both domains and are better on domain 1 than on domain 2. One reason is that although the size of the surface panels is reduced by half, the size of the volume elements is not. The size of cubes is determined by the system to achieve the best efficiency of the solver and may not be proportional to the size of surface mesh. For example, for domain 1, 6 cubes per side are used to get the last two data points. Although the surface mesh is refined by half, the cubes are not refined proportionally so the accuracy of u is not improved as much as for q. For domain 2, 12 cubes per side are used to get the middle data point while 24 cubes per side are used to get the last data point. The error for u improved a lot with the refinement of cubes in this case. Therefore, the accuracy of u can be improved by using more cubes. The CPU time on both domains is somewhere between $O(n^2)$ and $O(n)$.

3.5.2.2. Problem 4: *Nonlinear equation* $\nabla^2 u = \frac{3}{32}(u + u^3)$

This problem is chosen because the analytical solution is available. We are interested in the solution of this problem in two domains described. The analytical solution for this nonlinear equation is shown in Equation (3.24). u is used to prescribe the boundary condition.

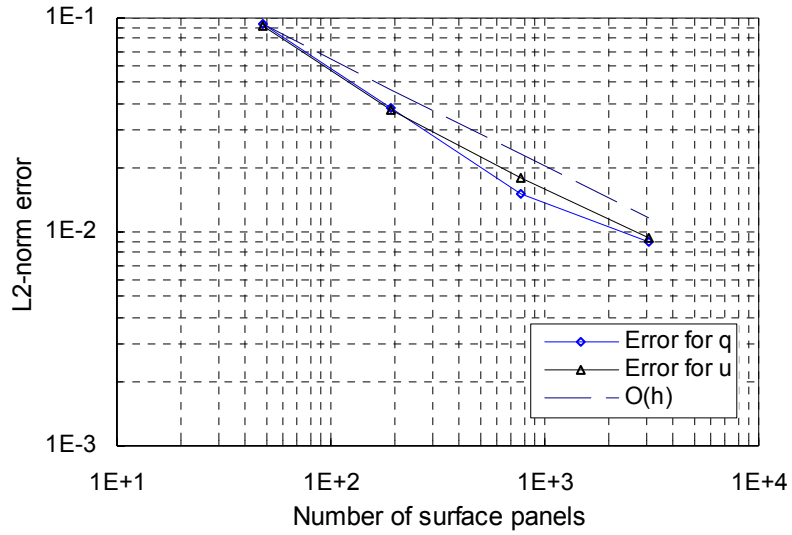
$$u = \tan(\frac{x + y + z}{8}), \quad q = -\frac{\partial u}{\partial n} = -\frac{1 + u^2}{8}(n_x + n_y + n_z). \tag{3.24}$$

The results on the two domains are shown in Table 19 and Figure 47 and the results on domain 2 are shown in Table 20 and Figure 48. It took 2 iterations for domain 1 to converge and 3 iterations for domain 2 to converge.

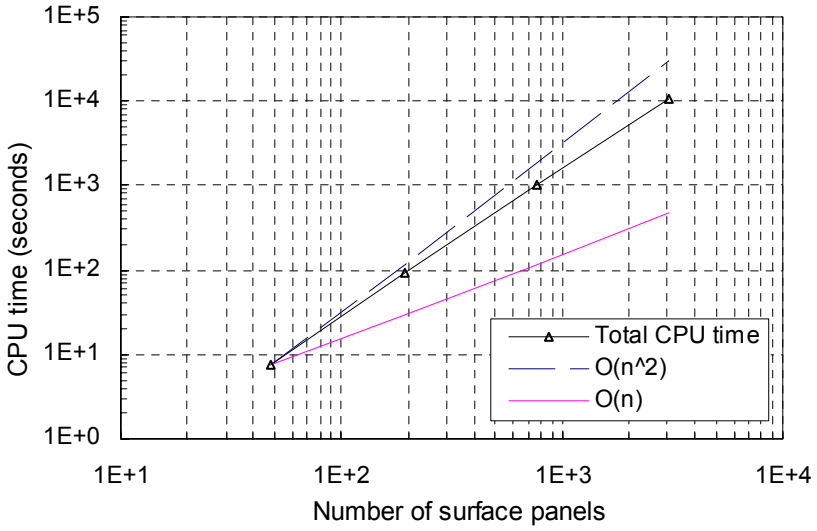
From the error plots shown in Figure 47(a) and Figure 48(a), the convergence rates for u and q in both domains are of $O(h)$. The CPU time for both domains still remains somewhere between $O(n^2)$ and $O(n)$.

Table 19. Results for the nonlinear problem on domain 1.

Number of surface panels	CPU time for each iteration (seconds)	Total CPU time (seconds)	L_2-norm error for q	L_2-norm error for u
48	0.86	7.328	9.29%	9.15%
192	6.80	90.546	3.83%	3.68%
768	161.52	1046.31	1.52%	1.77%
3072	822.46	10471.3	0.89%	0.93%



(a)

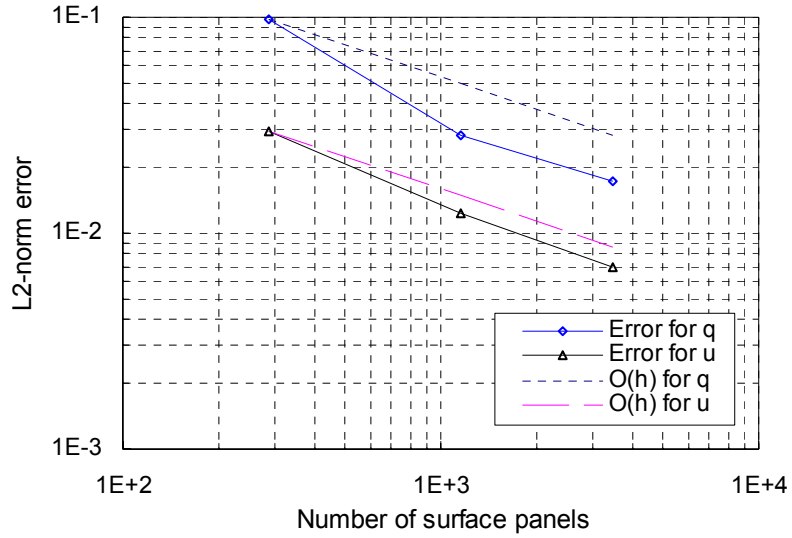


(b)

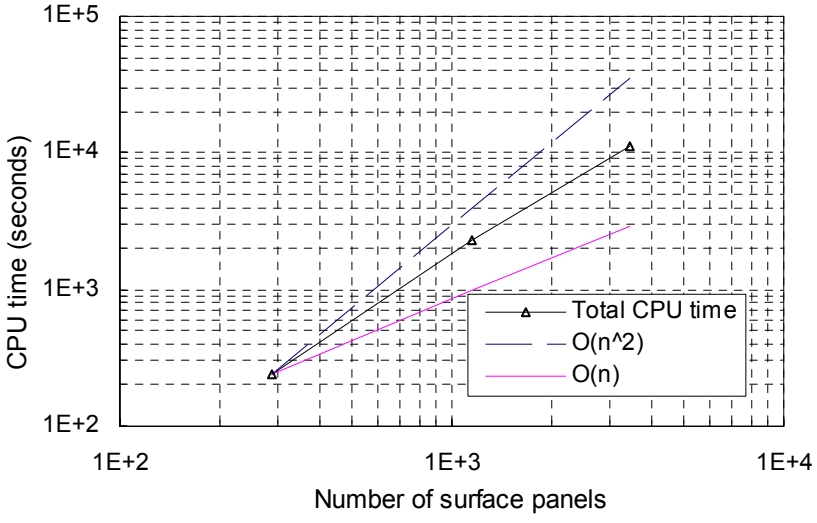
Figure 47. Results for the nonlinear problem on domain 1: (a) L_2-norm errors, (b) CPU time.

Table 20. Results for the nonlinear problem on domain 2.

Number of surface panels	CPU time for each iteration (seconds)	Total CPU time (seconds)	L_2-norm error for q	L_2-norm error for u
288	52.70	243.547	9.84%	2.99%
1152	303.83	2307.3	2.84%	1.24%
3456	985.07	11013.5	1.74%	0.70%



(a)



(b)

Figure 48. Results for the nonlinear problem on domain 2: (a) L_2-norm errors, (b) CPU time.

By comparing the results of the nonlinear problem and the Helmholtz problem we can see that the accuracy from the nonlinear problem is a little better than that for the Helmholtz problem (Error for q is 0.89% vs. 1.31% for domain 1) in about the same amount of CPU time. To explain this, we studied the effect of volume integration by examining the contribution of the volume integral in the integral formulation in two problems. We evaluated the values of three integrals, i.e., $\int_{\partial\Omega} G(\mathbf{x},\mathbf{y})q(\mathbf{y})ds(\mathbf{y})$, $\overset{cpv}{\int_{\partial\Omega}} F(\mathbf{x},\mathbf{y})u_{\partial\Omega}(\mathbf{y})ds(\mathbf{y})$, $\int_{\Omega} G(\mathbf{x},\mathbf{y})b(u(\mathbf{y}),\mathbf{y})dv(\mathbf{y})$ using analytical solutions of q, u and b and calculated their contribution to the total value. The weights of three integrals for two problems are shown in Table 21. For the specific nonlinear problem chosen, the contribution from the volume integral is less so the overall accuracy is a little better than in the Helmholtz problem.

Table 21. Weight of different integrals in step 1 for sphere domain

Equation	Number of surface panels	$\int_{\partial\Omega} G(\mathbf{x},\mathbf{y})q(\mathbf{y})ds(\mathbf{y})$	$\overset{cpv}{\int_{\partial\Omega}} F(\mathbf{x},\mathbf{y})u_{\partial\Omega}(\mathbf{y})ds(\mathbf{y})$	$\int_{\Omega} G(\mathbf{x},\mathbf{y})b(\mathbf{y})dv(\mathbf{y})$
$\Delta u = -u+h$	768	52.55%	15.02%	32.43%
$\Delta u = 3/32(u+u^3)$	768	67.63%	31.16%	1.21%

3.5.3 Comparison with existing approach

The second Poisson problem $\nabla^2 u = (2y^3 + 6y)e^{(x+z)}$ and the Helmholtz problem $\nabla^2 u = -u + h(x,y,z)$ are chosen from [15] so that we can compare the results from our approach to their results where volume discretization approach is used. The comparison is listed in Table 22.

Table 22. Comparison with existing approach

Equation	Results from Reference [15]	Results from our approach
$\nabla^2 u = (2y^3 + 6y)e^{(x+z)}$	2.3% in 750 seconds	0.9% in 1000 seconds
$\Delta u = -u+h$	3.2% in 10^4 seconds	2.91% in 5578 seconds

Since they did not report the computational resources they used to obtain these results in the paper, we cannot make direct comparison about the efficiency. However, the good performance of our approach is nevertheless clearly demonstrated.

3.6 CONCLUSIONS

In this chapter, we described an efficient numerical approach for computing volume integrals present in the boundary integral formulations of non-homogeneous and nonlinear problems with 3-D complex domains. In this approach, the 3-D uniform grid constructed initially to accelerate surface integration is used as the baseline mesh for the evaluation of volume integrals. The boundary cubes are further partitioned by projection of surface panels onto the boundary of corresponding cubes. No additional volume discretization of the interior cubes is necessary. Therefore, complicated volume discretization for the interior problem domain is avoided. Moreover with the uniform 3-D grid, the matrix sparsification techniques (such as the precorrected-FFT technique used in this work) can be extended to accelerate volume integration in addition to surface integration, thus greatly reducing the computational time. Based on this approach, accelerated BEM solvers for nonlinear problems are developed and tested using four partial differential equations.

Several techniques have been developed to handle issues arising from the implementation. In particular a 'projection + transformation' scheme is used to accurately evaluate singular volume integrals and to perform integration on partially filled cubes. The accuracy and the efficiency of these techniques are demonstrated through several examples. The criterion for nearby interactions is defined through numerical study which showed that direct calculation could only include nearest neighbors. It was found that the errors caused by the acceleration are very small, and that more efficiency can be achieved when the discretization is finer, indicating the developed approach is suitable for large-scale problems.

Accelerated BEM solvers for Poisson equations, a Helmholtz equation, and a nonlinear equation have been developed based on our approach. The only required input for the solvers is the surface mesh of the problem domain. The accuracy, efficiency and convergence of this solver are first demonstrated through an example with a spherical domain. A 3-D multiply-connected problem was then solved to illustrate the capability of the solver for handling problems with complex geometries. The results were compared with those obtained using the ADM on two problems, i.e., Poisson problem 2 and the Helmholtz problem. Although it is difficult to judge which method performs better, due to the differences in domain between the two studied cases, the good performance of our approach is nevertheless clearly demonstrated. In addition the approach developed in this work is applicable to any domain geometry in theory. Our approach also does not depend on the choice of radial basis functions as in the auxiliary domain method. The convergence rate achieved in all the test problems is about $O(h)$ and reasonable accuracy

for all the problems is achieved in this study. The efficiency of the developed approach is between $O(n^2)$ and $O(n)$ in all cases.

The accuracy of the proposed approach for non-homogeneous and nonlinear problems depends critically on the volume integration. The current scheme for volume integration, i.e., the "projection + transformation" scheme, has demonstrated some success in the cases presented. These examples, however, have relatively simple geometry, so more tests should be performed on problems with more complex geometries. For non-homogeneous and nonlinear problems with hyper-singular kernels, techniques to deal with the hyper-singular volume integral may be needed. In addition, this scheme is relatively inefficient in dealing with volume integrals involving boundary cubes. A better scheme should be sought to improve the efficiency.

REFERENCES

[1] Ahmad S., Banerjee PK, "Free vibration analysis of BEM using particular integrals", *Journal of Engineering Mechanics*, 1986; **112**:682-695.

[2] Aluru, N. R., and White, J., *"*A Fast integral equation technique for analysis of microflow sensors based on drag force calculation," *Technical Proceedings of the 1998 International Conference on Modeling and Simulation of Microsystem*, Santa Clara, 1998; 283 – 286, (available at *http://www.cr.org/publications/index.html*).

[3] Banerjee, P., *The Boundary Element Methods in Engineering*, MaGraw-Hill, London, 1994.

[4] Beskok, A., and Karniadakis, G., *"*Simulation of heat and momentum transfer in complex microgeometries*"*, *J. Thermophysics and Heat Transfer* 1994; **8(4)**: 647-655.

[5] Beskok, A., Karniadakis, G. and Trimmer, W., "Rarefaction and compressibility effects in gas microflows", *Trans. ASME* 1996; **118**: 448-456.

[6] Oscar P. Bruno and Leonid A. Kunyansky, "A fast, high-order algorithm for the solution of surface scattering problems: basic implementation, tests and applications", *Journal of Computational Physics*, 2001; **169(1)**: 80-110.

[7] Jian Ding and Wenjing Ye, "A Fast Integral Approach for Drag force Calculation Due to Oscillatory Slip Stokes Flows", *International Journal for Numerical Methods in Engineering*, 2004, **60**(9):1535-1567.

[8] Jian Ding, "An Analysis of Fisher's Equation", *project paper for Math6640*, 2001.

[9] Gad-el-hak, M., "The fluid mechanics of microdevices—The freeman scholar lecture", *J. Fluids Engineering,* 1999, ASME; **121**: 5-33.

[10] Xiao-Wei Gao, "The radial integration methods for evaluation of domain integrals with boundary-only discretization", *Engineering Analysis with Boundary Elements*, 2002, 26, pp. 905-916.

[11] J.E. Gómez and H. Power, "A multipole direct and indirect BEM for 2D cavity flow at low Reynolds number", *Engineering Analysis with Boundary Elements,* Volume 19, Issue 1, January 1997, Pages 17-31.

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

[13] Greengard L and Rokhlin V, "New Version of the Fast Multipole Method for the Laplace Equation in Three Dimensions". *Acta Numerica*, 1997: 229-269.

[14] Happel, J. and Brenner, H., *Low Reynolds Number Hydrodynamics*, Noordhoff International Publishing, 1973.

[15] Hsiao SC, Mammoli AA and Ingber MS, "The Evaluation of Domain Integrals in Complex Multiply-Connected Three-Dimensional Geometries for Boundary Element Methods", *Computational Mechanics*, 2003; **32**: 226-233.

[16] Marc S. Ingber, A. A. Mammoli, and M. J. Brown, "A comparison of domain integral evaluation techniques for boundary element methods", *International journal for numerical methods in engineering*, 2001, **52(4)**, pp. 417-432.

[17] Kanwal, R. P., "Drag on an Axially Symmetric Body Vibrating Slowly along its Axis in a Viscous Fluid", *J. Fluid Mech.* 1964; **19**: 631-636.

[18] J. T. Katsikadelis and M.S. Nerantzaki, "The boundary element methods for nonlinear problems", *Engineering Analysis with Boundary Elements*, 1999, 23, pp. 365-373.

[19] A. A. Mammoli and M. S. Ingber, "Stokes flow around cylinders in a bounded two-dimensional domain using multipole-accelerated boundary element methods", *Int. J. Numer. Meth. Engng*., 1999; **44**: 897-917.

[20] Mammoli AA, "Solution of Non-Linear Integral Equations in Complex Geometries with Auxiliary Auxiliary Integral Subtraction". *International journal for numerical methods in engineering*, 2002; **55**: 1115-1128.

[21] Maxwell, J.C., "On stresses in rarefied gases arising from in equalities of temperatures," *Philosophical Transactions of the Royal Society Part I* 1879; **170**: 231 – 256.

[22] Nabors K and White J, Fastcap: "A Multipole Accelerated 3-D Capacitance Extraction Program", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and System*, 1991; **10**(11): 1447-1459.

[23] Nagarajan A., and Mukherjee, S., "A Mapping Method for Numerical Evaluation of Two-Dimensional Integrals with 1/r Singularity," *Computational Mechanics*, 1993: **12**: 99-126.

[24] Nardini D and Brebbia CA, "A New Approach to Free Vibration Analysis Using Boundary Elements", *Applied mathematical modelling*, 1983; **7**(3): 157-162.

[25] Newman, J.N., "Distributions of sources and normal diploes over a quadrilateral panel*"*, *J. Engineering Mathematics* 1986; **20**: 113-126.

[26] Nowak AJ, "The multiple reciprocity method of solving transient heat conduction problems", *Advances in Boundary Elements*, vol. 2, Brebbia CA, Connor JJ (eds), Springer-Verlag: Berlin, 1989; 81-93.

[27] Nowak AJ, Brebbia CA, "The multiple reciprocity method: a new approach of transforming BEM domian integrals to the boundary", *Engineering Analysis*, 1989; **6**(3):81-93.

[28] O'Connor, L., "MEMS: Microelectromechanical Systems", *Mechanical Enginnering Journal* 1992; **114(2)**: 40-47.

[29] P. W. Partridge, C. A. Brebbia, and L. C. Wrobel, *The Dual Reciprocity Boundary Element Method*, Elsevier Applied Mechanics: London, 1992.

[30] Phillips, J. R., *Rapid Solution of Potential Integral Equations in Complicated 3-Dimensional Geometries*, Ph.D. Thesis, MIT, 1997.

[31] Phillips, J. R. and White, J., "A Precorrected-FFT Method for Electrostatic Analysis of Complicated 3-D Structures," *IEEE Transactions On Computer Aided Design,* 1997; **16(10)**: 1059 – 1072.

[32] Pierce A. P. and Napier J. A. L., "A Spectral Multipole Method for Efficient Solution of Large-Scale Boundary Element Models in Electrostatics", *International journal for numerical methods in engineering,* 1995; **38**: 4009-4034.

[33] Pozrikidis, C., *Boundary integral and singularity methods for linearized viscous flow*, Cambridge University Press, Cambridge, 1992.

[34] Saad, Y. and M. Schultz, *"*GMRES: A generalized minimal residual algorithm for solving symmetric linear systems", *SIAM journal on scientific and statistical computing*, 1986; **7**: 856 - 869.

[35] Schaaf, S. A. and P. L. Chambre, *Flow of Rarefied Gases*, Princeton University Press, Princeton, NJ, 1961.

[36] Shephard MS, Georges MK, "Automatic three-dimensional mesh generation by the finite octree technique", *International Journal for Numerical Methods in Engineering*, 1991; **32**:709-749.

[37] Shotsko A, Löhner R, "Three-dimensional parallel unstructured grid generation", *International Journal for Numerical Methods in Engineering*, 1995; **38**:905-925.

[38] Silveira LM, Elfadel IM, White JK, Chilukuri M and Kundert KS, "Efficient Frequency-Domain Modeling and Circuit Simulation of Transmission Lines", *IEEE Transactions on Components, Hybrids, and Manufacturing Technology-Part B: Advanced Packaging, special issue on Electrical Performance of Electrical Packaging*, 1994; **17**(4): 505-513.

[39] M. Tanaka, V. Sladek, and J. Sladek, "Regularization techniques applied to boundary element methods", Appl. Mech. Rev., 1994;**47(10)**:457-499.

[40] M. Tanaka, T. Matsumoto, and Q. F. Yang, "Time-stepping boundary element method applied to 2-D transient heat conduction problems", *Appl. Math. Modeling*, 1994; **18**:569-576.

[41] Veijola, Timo and Marek Turowski, "Compact Damping Models for Laterally Moving Microsturctures with Gas-Rarefaction Effects", *J. MEMS* June 2001; **10(2)**: 263-273.

[42] Ye, Wenjing, J. Kanapka, X. Wang and J. White, *"*Efficiency and Accuracy improvements for FastStokes, A precorrected-FFT accelerated 3-D Stokes Solver*"*, *Technical Proceedings of the 1998 International Conference on Modeling and Simulation of Microsystem*, Puerto Rico, 1998; 502 – 505, ((available at *http://www.cr.org/publications/index.html*).

[43] Ye W, Wang X and White J, "A Fast 3D Solver for Unsteady Flow with Applications to Micro-Electro-Mechanical Systems", *International Conference on Modeling and Simulation of Microsystems*, 1999, 518-521. (Available at *http://www.cr.org/publications/index.html*).

[44] Ye W, Wang X and White J, "Fast Stokes Solver for Generalized Flow Problems", *International Conference on Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators*, 2000, 524-527.

[45] W. Ye, X. Wang, W. Hemmert, D. Freeman and J. White, "Air Damping in Lateral Oscillating Micro Resonators: a Numerical and Experimental Study", *Journal of Microelectromechanical Systems*, 2003, Vol. 12, No. 5, pp. 1-10.

[46] J.J. Zhang, F.F. Afagh, and C.L. Tan, "Boundary integral transformation of body-force volume integral in BEM for 2D anisotropic elasticity", *Mechanics Research Communications*, 1996, Vol. 23, No. 2, pp. 227-232.

[47] Zhang, Xia and William C. Tang, "Viscous Air Damping in Laterally Driven Microresonators", *Proceedings of IEEE Workshop on Micro Electro Mechanical Systems*, MEMS '94, 1994; 199-204.

VITA

The author was born in Shandong, China. In September 1990, she attended Shandong Polytechnic University at Ji'nan, China and graduated with a Bachelor of Science degree in Material Science in 1994. After her graduation, she joined Nanjing University of Aeronautics & Astronautics, Nanjing, China and received a Master of Engineering degree in the College of Mechanical and Electrical Engineering in April 1997. Then she worked as a Material engineer in Zhejiang Electric Power Test & Research Institute, Hangzhou, China before she went to the Department of Mechanical & Production Engineering in National University of Singapore in July 1998. After she got her second Master of Engineering degree in December 1999, she enrolled at Georgia Institute of Technology in the George W. Woodruff School of Mechanical Engineering to pursue her Ph.D. degree. During her stay in Georgia Tech, she held Graduate Research Assistantship. She loves music and travel.