I, Milind Shrikant Bapat
hereby submit this work as part of the requirements for the degree of:
Master of Science
in:
Department of Mechanical Engineering
It is entitled:
Fast Multipole Boundary Element Method For Solving Two-Dimensional Acoustic Wave Problems

This work and its defense approved by:
Chair: Dr. Yijun Liu
Dr. Jay Kim
Dr. Kumar Vemaganti
Fast Multipole Boundary Element Method
For Solving Two-Dimensional Acoustic Wave Problems

A thesis submitted to the
Graduate School
of the University of Cincinnati
in partial fulfillment of the
requirements for the degree of
Master of Science
In the Department of Mechanical, Industrial and Nuclear Engineering
of the College of Engineering
in December 2006

by

Milind Bapat

B.E. Mechanical Engineering
Maharashtra Institute of Technology, Pune, India.
August 2002.

Committee Chair: Dr. Yijun Liu
ABSTRACT

The boundary element method (BEM) is a numerical method for solving boundary value problems. The boundary element method has a clear advantage over other techniques like finite element method (FEM) in problems involving infinite domains. Hence the boundary element method has found many applications in the field of acoustics which often exist in infinite domains. The traditional approach for finding solutions to acoustic problems using the boundary element method has a computational complexity of the order $O(N^2)$. This makes the computation very slow as the number of nodes increase. A new technique called fast multipole method (FMM) has emerged in the last decade. Replacing the normal matrix-vector multiplication with the fast multipole method reduces the computational time to order $O(N)$. In this thesis the fast multipole method has been used to accelerate the boundary element method for 2-D acoustic wave problems. The relevant formulae are derived. It is shown that the computational time is of the order $O(N)$ for this formulation. It is also observed that the memory required is much lesser and hence larger models can be solved. The formulation is a very basic one and gives good results as shown by the numerical examples. Use of higher-order elements and hypersingular formulation will result in a very capable and robust solver in the future.
First and foremost, I would like to thank Dr. Yijun Liu for his able guidance and support in this work. This work could be completed only through his suggestions and help. I would like to thank Dr. Jay Kim for his help in topics related to the field of acoustics. I also thank Dr. Kumar Vemaganti for serving on the committee and his suggestions which have made the report better. I would also like to express my gratitude to my colleagues at the CAE Laboratory, Liang Shen and Jian Sun for their insights into the topic of fast multipole methods.

I would also like to thank Nilesh Billade, a colleague and friend for his help in presenting this thesis. I also thank my other colleagues Manas Phadke, Shardool Chirputkar and Nimish Jagtap for their help. I would like to express my appreciation for the necessary infrastructure provided by the CAE laboratory, OCC laboratories and the libraries. I am grateful to my friends, Kalpak Gatne, Aniruddha Palsule, Ajit Pharne, Rahul Patki, Amit Kulkarni and Koustubh Ashtekar, without whose support this work would not have been completed.

Last but not the least; I thank my parents and my sister who have stood by me in this venture. I also appreciate each and every contribution made to this thesis by everyone whose names I could not mention here.
Table of Contents

List of Figures .......................................................................................................................... 4
List of Tables ........................................................................................................................... 5

Chapter 1 Introduction ........................................................................................................ 6
  1.1 Introduction .................................................................................................................. 6
  1.2 The Boundary Element Method (BEM) ...................................................................... 6
  1.3 BEM for the Acoustics Problem .................................................................................. 7
  1.4 Fast Multipole BEM for Acoustics Problems .............................................................. 8
  1.5 Structure of This Thesis ............................................................................................... 9

Chapter 2 Conventional Boundary Element Method for the Acoustics Problem ....... 10
  2.1 Introduction ................................................................................................................ 10
  2.2 Helmholtz equation and BIE Formulation for 2-D Acoustics ................................. 11
  2.3 BEM Formulation with Constant Element ............................................................... 18
  2.4 Solution to the System of Equations .......................................................................... 22

Chapter 3 Fast Multipole Method ..................................................................................... 23
  3.1 Introduction ................................................................................................................ 23
  3.2 Fast Matrix-Vector Multiplication ............................................................................. 23
  3.3 Iterative Solvers and Fast Matrix-Vector Multiplication ........................................... 26
  3.4 GMRES Method ........................................................................................................ 26
  3.5 Fast Multipole Method ............................................................................................... 27
  3.6 Mathematical Formulation for FMM ......................................................................... 29
    a. Decomposition for G kernel .................................................................................... 29
b. M2M Translation ..................................................................................................... 31

c. M2L Translation ..................................................................................................... 32

d. L2L Translation ..................................................................................................... 33

e. Translations for the F kernel .................................................................................... 34

3.7 The Fast Multipole Method Algorithm ................................................................... 34

3.8 Application to the BEM Problem ............................................................................ 37

Chapter 4 Code Development .................................................................................... 39

4.1 C++ Solver for Conventional BEM ........................................................................ 39

4.2 C++ GMRES Solver .............................................................................................. 39

4.3 Tree traversal in FMM BEM Solver ........................................................................ 39

a. Algorithm for creation of the tree ........................................................................ 40

b. Algorithm for upward pass ................................................................................... 41

c. Algorithm for downward pass ............................................................................. 41

4.4 Brief overview of the main classes and their functions for FMM BEM ................. 41

Chapter 5 Results ....................................................................................................... 43

5.1 Results for 2-D Acoustics problems ...................................................................... 43

a. 2-D Acoustics Interior Domain Problems .............................................................. 43

b. 2-D Acoustics Exterior Domain Problems .............................................................. 46

c. 2-D Acoustics Scattering Problems ...................................................................... 50

5.2 Comparison with the conventional BEM code ..................................................... 52

Chapter 6 Discussions and Future Work .................................................................. 54

References .................................................................................................................. 56
Appendix A. Derivation of the Multipole Coefficients and Functions..........................62

Appendix B. Code listing for the FMBEM solver............................................................79
List of Figures

Figure 1 Point P inside the boundary ................................................................. 15
Figure 2 Point P on the boundary S surrounded by a small semicircle ............... 15
Figure 3 A boundary S divided into m constant elements ................................... 18
Figure 4 A constant element .............................................................................. 21
Figure 5 Multiple node to node interaction ....................................................... 28
Figure 6 Single cell to cell interaction ............................................................... 28
Figure 7 Decomposition for G kernel ................................................................. 30
Figure 8 M2M Translation ............................................................................... 31
Figure 9 M2L Translation ................................................................................ 32
Figure 10 L2L Translation ................................................................................ 33
Figure 11 Domain enclosed in a box ................................................................. 34
Figure 12 Division of any cell ........................................................................... 34
Figure 13 Tree structure for the domain ............................................................ 36
Figure 14 Interior Acoustics Problem ................................................................. 42
Figure 15 Results for an interior acoustics problem ........................................... 45
Figure 16 Exterior Acoustics Problem ............................................................... 46
Figure 17 Exterior acoustics problem ............................................................... 46
Figure 18 Pressure Plot for Exterior Acoustic case k=1 .................................... 48
Figure 19 Results for an Exterior acoustics problem ....................................... 49
Figure 20 Results for an acoustic scattering problem ...................................... 50
Figure 21 Computational time for the FMM solver .......................................... 51
List of Tables

Table 1 Comparison of the values along the centerline of the duct………………………..41
Table 2 Convergence of the pressure value at the center…………………………………..41
Table 3 Comparison of the values along a radial line of the cylinder……………………..44
Chapter 1

Introduction

1.1 Introduction

In many engineering problems, the solutions to partial differential equations (PDEs) are sought. The boundary conditions are specified on the physical boundary of the domain in case of some problems. These are called the boundary value problems (BVPs). The process of finding the solutions analytically to BVPs becomes increasingly difficult with the increasing complexity of the given geometries and boundary conditions. Many numerical techniques have been devised to solve these problems. Finite element method (FEM) and Finite Difference Method are two well-known methods. Boundary element method (BEM), based on the boundary integral equation (BIE) is another numerical method for solving BVPs. Engineering problems in acoustics, soil mechanics, and fracture mechanics can be solved using the boundary element method. The method is very elegant and gives very accurate results. The conventional way of solving the BEM problems takes considerable computational time. The introduction of fast multipole method (FMM), for solving the BEM problems has accelerated the calculation time. The fast multipole boundary element method (FMBEM) is very robust in nature and can solve very large models. The method also consumes lesser memory and has an accuracy similar to the conventional approach.

1.2 The Boundary Element Method

The boundary element method consists of solving an integral equation that is mathematically equivalent to the original partial differential equation (PDE). This integral equation is termed the boundary integral equation (BIE). BIE is defined on the spatial
boundary of the domain alone. To obtain the solution to the integral equation for a complex
gometry, the boundary is discretized into elements. On expressing the integral equation
for each element a linear system of equations is obtained.

Some basic work of integral equation techniques was done by Fredholm [1].
the application of integral equations to problems in mechanics. Kupradze [4] discussed the
solution to potential problems using the integral equations which used potential theory by
Kellogg [5]. Jaswon [6], also independently applied integral equations in the potential
theory. Cruse and Rizzo [7] applied this method to elastostatics. This approach is similar to
the current approach (or sometimes called ‘Direct’ method). The boundary element method
emerged as a computationally viable technique during the 1970’s and has been developed
substantially in the following years. Since, only the spatial boundaries are involved for
solving, a \((N-1)\) dimension model (of the spatial/physical boundary) needs to be created for
solving the same problem, where an \(N\) dimension model is created for FEM. Also it can be
applied to infinite domains. This is the main advantage of BEM. Today the BEM has found
applications in many fields of computational mechanics, such as the wave propagation,
heat transfer, diffusion and convection, fluid flow, fracture mechanics, electric problems,
geomechanics, plates and shells, inelastic problems, contact problems, design sensitivity
and optimization problems.

1.3 BEM for the Acoustic Problem

The governing equation for the acoustic problems is the Helmholtz equation. The earliest
attempts to solve acoustic problems using BEM were by Chen [8]. He discussed the
radiation from an arbitrary body in an infinite medium. Chertock [9] discussed the method for predicting the field generated by a body, on whose boundary the pressure is known. Copley [10] discussed the radiation problem for vibrating bodies. The above formulations do not correctly predict the field at certain frequencies called the eigen frequencies. Schenk [11] improved this formulation for solving the problem at eigen frequencies. This approach is widely accepted as the CHIEF method. Shaw [12] applied the developed method to some ocean engineering problems. Burton and Miller [13] proposed a hypersingular formulation for the problem of eigen frequencies. Shaw and Tai [14] made a study of the modes of these eigen frequencies. Beyond this point a lot of research has been carried out and the scope of these applications has been expanded.

1.4 Fast Multipole BEM for Acoustic Problems

FMM finds its roots in tree-based algorithms. Barnes-Hut algorithm [15] is one such algorithm and is very closely related to the FMM. These tree-based algorithms were originally developed to deal with multibody problems in astrophysics or simulations in molecular dynamics etc. Rokhlin [16] first introduced FMM to solve the Laplace equation. Rokhlin used a binary tree structure. Greengard [17] applied this method to multibody problems. He also introduced the use of quad-trees for solving 2-D problems. The use of a quad-tree for 2-D problems can be compared to the Barnes-Hut algorithm. Mammoli & Ingber [18] applied fast multipole BEM using an indirect BEM for solving Stokes flow around cylinders. Yoshida [19] applied the FMM to solve crack propagation problems. FMM has been applied to a variety of problems. A lot of different aspects of the fast multipole method have also been studied so far. Rokhlin [20] first applied the FMM

8
method for solving the Helmholtz equation. Fukui [21] also solved the Helmholtz equation using FMM. Fukui and Rokhlin discussed the formulation for 3-D acoustic problems. Similar attempts were made by Zhao and Chew [22] to solve the 2-D acoustic problem. Nishimura [23] reviewed the works till then and discussed these methods. Chen [24] used a hypersingular formulation in his FMBEM code to solve 2-D acoustic problem.

1.5 Structure of This Thesis

The remaining chapters of this thesis are as follows: In Chapter 2, the boundary integral equation for the 2-D Acoustic problem is formulated. The BEM formulation for the problem is provided using constant elements. Briefly the iterative methods for solving the problem are discussed. In Chapter 3, the fundamentals of the fast multipole method are discussed. The use of fast multipole method in association with iterative solvers is explained. The fast multipole method formulation for the 2-D acoustic problem is developed. In Chapter 4, a C++ code for the fast multipole method is discussed. In Chapter 5, the numerical results obtained by using the code are presented. In Chapter 6, the thesis concludes with discussions and the scope for future work.
Chapter 2  

Boundary Element Method for Acoustic Problem  

2.1 Introduction  

The phenomenon of the sound wave propagation has been studied in depth for many engineering applications. The starting point for most discussions is the Helmholtz equation. In the following section, a brief derivation of the Helmholtz equation is provided. 

A sound wave passing through any point in a compressible fluid creates a change of pressure at that point given by:

\[ \phi_r = \phi_0 + \phi \]  

\( \phi_r \) is the total pressure  
\( \phi_0 \) is the ambient value of pressure  
\( \phi \) is the acoustic pressure (i.e. change produced in the pressure)  

If  \( c \) is the velocity of the wave,  
\( V \) the specific volume,  
\( \rho \) the density of the compressible fluid, and  
\( \rho_0 \) the density of the fluid at that point,

then the continuity equation is given by

\[ \frac{\partial \phi}{\partial t} + \rho_0 c^2 \nabla V = 0. \]  

(2)  

The Euler’s equation is given by

\[ \rho_0 \frac{\partial V}{\partial t} = -\nabla \phi. \]  

(3)  

Combining equations (2) and (3) results in
\[ \rho_0 \nabla \cdot \left( \frac{1}{\rho_0} \nabla \phi \right) - \frac{1}{c^2} \frac{\partial^2 \phi}{\partial t^2} = 0 \]  

(4)

Assuming a homogenous media, equation (4) can be written as

\[ \nabla^2 \phi - \frac{1}{c^2} \frac{\partial^2 \phi}{\partial t^2} = 0 \]  

(5)

Considering that time harmonic waves are being solved, the pressure can be expressed as

\[ \phi = \tilde{\phi}e^{i\omega t} \]  

(6)

where \( \tilde{\phi} \) is the amplitude of the wave (a function of space and frequency)

On substituting equation (6) into equation (5), the following result is obtained

\[ \nabla^2 \tilde{\phi} + k^2 \tilde{\phi} = 0 \]  

(7a)

where \( k = \frac{\omega}{c} \) is the wave number.

(7b)

This is the Helmholtz equation. Here \( k \) is the wave number and \( \tilde{\phi} \) the pressure at any point. In the following discussions, \( \tilde{\phi} \) will be replaced by \( \phi \). The equation (6) can be applied to find the pressure at any given frequency.

### 2.2 Helmholtz Equation and BIE Formulation for 2-D Acoustic

It was shown in the previous section that for a homogenous medium the mathematical formulation of the acoustic problem in frequency domain is given by the Helmholtz equation

\[ \nabla^2 \phi + k^2 \phi = 0 \]  

(7a)

where \( \phi \) is the acoustic pressure.
Now analogous to the potential problem, we can say that the fundamental solution $G$ will be given by the equation

$$\nabla^2 G + k^2 G = -\delta(P, P_0)$$  \hspace{1cm} (8)

In this case the point $P_0$ is a singular point source and $\delta$ is the Dirac-delta function.

$G(P, P_0)$ is a function of two points and represents the pressure field at point $P$ due to a source at point $P_0$.

In a very basic formulation we will consider a symmetrical radial solution to this equation i.e. $G$ varies only with respect to $r$ (in polar co-ordinate system in 2-D)

Rewriting the equation (8) for the entire plane (except $P_0$) in polar co-ordinate system

$$\frac{d^2 G}{dr^2} + \frac{1}{r} \frac{dG}{dr} + k^2 G = 0$$  \hspace{1cm} (9)

Substitute $\chi = kr$, we obtain the standard Bessel equation of order zero

$$\chi^2 \frac{d^2 G}{d\chi^2} + \chi \frac{dG}{d\chi} + \chi^2 G = 0$$  \hspace{1cm} (10)

The general solution for the above equation, in terms of the first kind and second kind Hankel functions of the nth order $H^{(1)}_n(\cdot)$ and $H^{(2)}_n(\cdot)$, can be written as

$$G = A H^{(1)}_\omega(kr) + B H^{(2)}_\omega(kr)$$  \hspace{1cm} (11)

Wu [26] has discussed the method to find the constants $A$ and $B$.

Integrating the Helmholtz equation over a tiny circular area $\partial \Omega$ bounding $P_0$ and applying divergence theorem, the equation (8) can be written as
\[
\lim_{\Omega \to 0} \int_{\Omega} \frac{\partial \psi}{\partial n} d\Gamma = -1
\]  

(12)

Wu [26] has shown that \(A=0\) and \(B=-i/4\)

Hence the fundamental solution is given by

\[
G = -\frac{i}{4} H_0^{(2)}(kr)
\]  

(13)

The normal derivative of the fundamental solution is given by

\[
\frac{\partial G}{\partial n} = \frac{ik}{4} H_1^{(2)}(kr) \frac{\partial r}{\partial n}
\]  

(14)

Using these fundamental solutions, the boundary integral equation for the problem is derived.

The Green’s second identity as shown below will be applied:

\[
\int \int \left[ u \nabla^2 v - v \nabla^2 u \right] dV = \int \int \left[ u \frac{\partial v}{\partial n} - v \frac{\partial u}{\partial n} \right] dS
\]  

(15)

The equation (8) for fundamental solution can also be written as

\[
\nabla^2 G + k^2 G + \delta(P, P_0) = 0
\]  

(16)

Rearranging the terms,

\[
\nabla^2 G = -k^2 G - \delta(P, P_0)
\]  

(17)

Rearranging the terms of equation (7),

\[
\nabla^2 \phi = -k^2 \phi
\]  

(18)

Substituting \(u = \phi\) and \(v = G\) in equation (15),

\[
\int \left[ \phi \nabla^2 G - G \nabla^2 \phi \right] dV = \int \left[ \phi \frac{\partial G}{\partial n} - G \frac{\partial \phi}{\partial n} \right] dS
\]  

(19)
Substituting the equations from (17) and (19),

\[ \int \phi (-k^2G - \delta(P, P_0)) - G(-k^2\phi) \, dV = \int \left[ \phi \frac{\partial G}{\partial n} - G \frac{\partial \phi}{\partial n} \right] \, dS \]  \hspace{1cm} (20)

Hence,

\[ \int \phi (-k^2G - \delta(P, P_0)) + Gk^2\phi \, dV = \int \left[ \phi \frac{\partial G}{\partial n} - G \frac{\partial \phi}{\partial n} \right] \, dS \]  \hspace{1cm} (21)

Hence,

\[ \int [-\phi k^2G - \phi \delta(P, P_0) + Gk^2\phi] \, dV = \int \left[ \phi \frac{\partial G}{\partial n} - G \frac{\partial \phi}{\partial n} \right] \, dS \]  \hspace{1cm} (22)

Therefore,

\[ \int [-\phi \delta(P, P_0)] \, dV = \int \left[ \phi \frac{\partial G}{\partial n} - G \frac{\partial \phi}{\partial n} \right] \, dS \]  \hspace{1cm} (23)

By the definition of Dirac-Delta function

\[ \phi(P_0) = \int \left[ G(P, P_0) \frac{\partial \phi(P)}{\partial n} - \phi(P) \frac{\partial G(P, P_0)}{\partial n} \right] \, dS(P) \]  \hspace{1cm} (24)

This can be written in a simplified form as

\[ \phi(P_0) = \int [Gq - F\phi] \, dS \]  \hspace{1cm} (25)

where

\[ q = \frac{\partial \phi}{\partial n} \quad \text{and} \quad F = \frac{\partial G}{\partial n} \]

The equation (25) has converted all the terms in the equation to boundary value terms. In a well posed boundary value problem only one of \( q \) or \( \phi \) or a relating equation is given. Hence this equation cannot be used directly. To find the boundary variables first we
consider a point $P_0$. Now we shall consider the limiting case as the point is moved to the boundary (Fig. 1).

![Figure 1. Point $P$ inside the boundary](image1)

In other words, the point $P_0$ can be assumed to be lying on the boundary and a small semicircle $S_\varepsilon$ is assumed (Fig. 2).

![Figure 2. Point $P$ on the boundary $S$ surrounded by a small semicircle](image2)

The limit is found as $\varepsilon$ tends to 0 (Fig. 2). Applying to the equation (25) we have

$$\lim_{r_0 \to S} \phi(P_0) = \lim_{r_0 \to S} \int_S [Gq - F\phi]dS$$

(26)

Evaluating the first term on the right hand side first,
\[
\lim_{\varepsilon \to 0} \int_S G(P, P_0) q(P) dS = \lim_{\varepsilon \to 0} \int_S G(P, P_0) q(P) dS \quad (27)
\]

Separating the integral into two domains \( S - S_\varepsilon \) and \( S_\varepsilon \)

\[
= \lim_{\varepsilon \to 0} \int_{S - S_\varepsilon} G(P, P_0) q(P) dS + \lim_{\varepsilon \to 0} \int_{S_\varepsilon} G(P, P_0) q(P) dS \quad (28)
\]

Substituting the equations (13) and (14), we have

\[
= \lim_{\varepsilon \to 0} \int_{S - S_\varepsilon} G(P, P_0) q(P) dS + \lim_{\varepsilon \to 0} \int_{S_\varepsilon} -\frac{i}{4} H_0^{(2)}(kr) q(P) r d\theta 
= \lim_{\varepsilon \to 0} \int_{S - S_\varepsilon} G(P, P_0) q(P) dS + \lim_{\varepsilon \to 0} \int_{0}^{\theta} -\frac{i}{4} H_0^{(2)}(k\varepsilon) q(\varepsilon) \varepsilon d\theta 
\]

But \( \lim_{\varepsilon \to 0} (\varepsilon H_0^{(2)}(k\varepsilon)) = 0 \) \( (31) \)

Therefore,

\[
\lim_{\varepsilon \to 0} \int_S G(P, P_0) q(P) dS = \lim_{\varepsilon \to 0} \int_{S - S_\varepsilon} G(P, P_0) q(P) dS \quad (32)
\]

No additional term is added for the domain \( S_\varepsilon \) (i.e. when we reach the boundary). That is, there is no jump term here.

Consider the second term on the right hand side

\[
\lim_{\varepsilon \to 0} \int_S F(P, P_0) \phi(P) dS = \lim_{\varepsilon \to 0} \int_S F(P, P_0) \phi(P) dS \quad (33)
\]

Again separating the integral into two domains \( S - S_\varepsilon \) and \( S_\varepsilon \)

\[
= \lim_{\varepsilon \to 0} \int_{S - S_\varepsilon} F(P, P_0) \phi(P) dS + \lim_{\varepsilon \to 0} \int_{S_\varepsilon} F(P, P_0) \phi(P) dS \quad (34)
\]

Substituting the equations (13) and (14),
\[
= \lim_{\varepsilon \to 0} \int_{S-S_c} F(P, P_0)\phi(P)dS + \lim_{\varepsilon \to 0} \int_{S_c} \frac{ik}{4} H_1^{(2)}(kr) \frac{\partial \phi}{\partial n} \phi(P)rd\theta
\]  
(35)

\[
= \lim_{\varepsilon \to 0} \int_{S-S_c} F(P, P_0)\phi(P)dS + \lim_{\varepsilon \to 0} \int_{S_c} \frac{z}{4} H_1^{(2)}(k\varepsilon)\phi(P)\varepsilon d\theta
\]  
(36)

But \( \lim_{\varepsilon \to 0} (\varepsilon H_1^{(2)}(k\varepsilon)) = -\frac{2i}{k\pi} \)  
(37)

Putting this value in equation (36) we have

\[
= \lim_{\varepsilon \to 0} \int_{S-S_c} F(P, P_0)\phi(P)dS + \lim_{\varepsilon \to 0} \int_{S_c} \frac{z}{4} \frac{2i}{k\pi} \phi(P)d\theta
\]  
(38)

\[
= \lim_{\varepsilon \to 0} \int_{S-S_c} F(P, P_0)\phi(P)dS + \lim_{\varepsilon \to 0} \int_{S_c} -\frac{1}{2\pi} \phi(P)d\theta
\]  
(39)

\[
= \lim_{\varepsilon \to 0} \int_{S-S_c} F(P, P_0)\phi(P)dS - \frac{1}{2} \phi(P_0)
\]  
(40)

This is the jump term. The integral is not continuous at \( P \) when \( P_0 \) is on the boundary.

On substituting equations (32) and (40) in equation (25) the result is

\[
C(P_0)\phi(P_0) = \int_S \left[ G(P, P_0)q(P) - F(P, P_0)\phi(P) \right]dS(P)
\]  
(41)

Here \( C(P_0) = \frac{1}{2} \) if \( P_0 \) is on the boundary

\[
C(P_0) = 1 \quad \text{if } P_0 \text{ is inside the domain.}
\]

This equation is applicable for both interior and exterior problems.

In the case of scattering problems an additional term for the incident wave is introduced.
\[ C(P_0)\phi(P_0) = \int_S \left[ G(P, P_0)q(P) - F(P, P_0)\phi(P) \right] dS(P) + \phi^i(P_0) \] (42)

where \( \phi^i \) is the incident wave.

### 2.3 BEM Formulation with Constant Elements

Consider a boundary \( S \) (Fig.3). Let it be divided into \( m \) constant elements. Constant elements are defined as those on which both the potential function \( \phi \) and its normal derivative \( q \) is constant over the entire element [42].

![Figure 3. A boundary S divided into m constant elements](image)

Assuming \( S \) to be divided into \( m \) constant elements

On any element \( S_i \),

\[ \phi = \phi_i \text{ (a constant)} \]

\[ q = q_i \text{ (a constant)} \]

The boundary integral equation (41) is used to derive the formulation.

For every source point (on every element \( S_i \)), we have
\[ C_{ij} = \int_S [G_{ij} - F_{ij}]dS \] \hspace{1cm} (43)

First term on the right hand side is evaluated as:

\[ \int_{S_j} G_{ij}dS = \sum_{j=1}^{m} \int_{S_j} G_{ij}dS \] \hspace{1cm} (44)

\[ \int_{S_j} G_{ij}dS = \sum_{j=1}^{m} q_j \int_{S_j} G_{ij}dS \] \hspace{1cm} (45)

\[ \int_{S_j} G_{ij}dS = \sum_{j=1}^{m} q_j g_{ij} \] \hspace{1cm} (46)

where

\[ g_{ij} = \int_{S_j} G_{ij}dS \] \hspace{1cm} (47)

Similarly the second term can be evaluated as

\[ \int_{S_j} F_{ij}dS = \sum_{j=1}^{m} \int_{S_j} F_{ij}dS \] \hspace{1cm} (48)

\[ \int_{S_j} F_{ij}dS = \sum_{j=1}^{m} \phi_j \int_{S_j} F_{ij}dS \] \hspace{1cm} (49)

\[ \int_{S_j} F_{ij}dS = \sum_{j=1}^{m} \phi_j \hat{f}_{ij} \] \hspace{1cm} (50)

where

\[ \hat{f}_{ij} = \int_{S_j} F_{ij}dS \] \hspace{1cm} (51)

Thus equation (43) can be written as
\[ c_i \phi_i = \sum_{j=1}^{m} q_j g_{ij} - \sum_{j=1}^{m} \phi_j \tilde{f}_{ij} \]  
(52)

The equation (52) is modified using the term

\[ f_{ij} = c_i \delta_{ij} + \tilde{f}_{ii} \]  
(53)

The final equation becomes

\[ \sum_{j=1}^{m} q_j g_{ij} = \sum_{j=1}^{m} \phi_j f_{ij} \]  
(54)

Putting in terms of matrices, we obtain:

\[
\begin{bmatrix}
    g_{11} & g_{12} & \cdots & g_{1m} \\
    g_{21} & g_{22} & \cdots & g_{2m} \\
    \vdots & \vdots & \ddots & \vdots \\
    g_{m1} & g_{m2} & \cdots & g_{mm}
\end{bmatrix}
\begin{bmatrix}
    q_1 \\
    q_2 \\
    \vdots \\
    q_m
\end{bmatrix} =
\begin{bmatrix}
    f_{11} & f_{12} & \cdots & f_{1m} \\
    f_{21} & f_{22} & \cdots & f_{2m} \\
    \vdots & \vdots & \ddots & \vdots \\
    f_{m1} & f_{m2} & \cdots & f_{mm}
\end{bmatrix}
\begin{bmatrix}
    \phi_1 \\
    \phi_2 \\
    \vdots \\
    \phi_m
\end{bmatrix}
\]

(55)

This equation is used for solving the unknown boundary values at the nodes.

Evaluation of \( f_{ij} \) and \( g_{ij} \) cannot be done analytically and is done using Gaussian quadrature. On the other hand, \( f_{ii} \) and \( g_{ii} \) are singular in nature and their values need to be found out.

Consider an element of length \( 2R_i \) (Fig. 4). The Cauchy Principal value (CPV) is taken as the value of the integral.
Now,

\[ g_{ii} = \int_{S_i} G_i dS \]  

(56)

To find the CPV,

\[ g_{ii} = \lim_{\varepsilon \to 0} \int_{S_i \rightarrow S_e} G_i dS \]  

(57)

\[ g_{ii} = \lim_{\varepsilon \to 0} \int_{S_i \rightarrow S_e} -\frac{i}{4} H_0^{(2)}(kr) dS \]  

(58)

\[ g_{ii} = \lim_{\varepsilon \to 0} \int_{r_i \rightarrow r_e} -\frac{i}{4} H_0^{(2)}(kr) dS \]  

(59)

\[ g_{ii} = \frac{\pi R}{2} \left[ S_0(kR_i)H_1^{(2)}(kR_i) - S_1(kR_i)H_0^{(2)}(kR_i) \right] + R_i H_0^{(2)}(kR_i) \]  

(60)

where \( S_n() \) denotes the Struve function of the nth order.

This value is used in the BEM formulation.

Similarly for finding \( f_{ii} \),
\[ \hat{f}_{ii} = \int_{S_i} F_i dS \]  

(61)

\[ \hat{f}_{ii} = \lim_{\varepsilon \to 0} \int_{S_i - S_e} F_i dS \]  

(62)

\[ \hat{f}_{ii} = \lim_{\varepsilon \to 0} \int_{S_i - S_e} \frac{ik}{4} H_1^{(2)} (kr) \frac{\partial r}{\partial n} dS \]  

(63)

\[ \hat{f}_{ii} = 0 \]  

(64)

\[ f_{ii} = c_i + \hat{f}_{ii} \]  

(65)

\[ f_{ii} = 0.5 \]  

(66)

2.4 Solution to the System of Equations

The equation (55) is used for solving the matrix equation

\[
\begin{bmatrix}
g_{11} & g_{12} & \cdots & g_{1m} \\
g_{21} & g_{22} & \cdots & g_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
g_{m1} & g_{m2} & \cdots & g_{mm}
\end{bmatrix}
\begin{bmatrix}
q_1 \\
q_2 \\
\vdots \\
q_m
\end{bmatrix}
=
\begin{bmatrix}
f_{11} & f_{12} & \cdots & f_{1m} \\
f_{21} & f_{22} & \cdots & f_{2m} \\
\vdots & \vdots & \ddots & \vdots \\
f_{m1} & f_{m2} & \cdots & f_{mm}
\end{bmatrix}
\begin{bmatrix}
\phi_1 \\
\phi_2 \\
\vdots \\
\phi_m
\end{bmatrix}
\]

In a well-posed boundary value problem, one of the values of \( q \) or \( \phi \) is known for every boundary node. In the solution for such a BVP, all the unknown values are shifted to the left and the known values to the right. This converts the above system into a matrix system

\[ \mathbf{Ax} = \mathbf{b} \]  

where \( \mathbf{x} \) is the vector of all the unknowns. \( \mathbf{A} \) is unsymmetrical and also dense.

The solution to \( \mathbf{x} \) can be found using various solvers. Direct solvers like Gaussian elimination give high accuracy. These solvers take very large time though \( (O(N^3)) \). Iterative solvers can also be utilized to solve the system. The accuracy of these solvers is controlled by a tolerance. These solvers are much faster and have an efficiency of \( (O(N^2)) \).
Chapter 3

Fast Multipole Method

3.1 Introduction

The BEM formulation of the 2-D acoustic problem was discussed in the earlier chapter. The output of the numerical formulation is a matrix equation of the form $Ax = b$. Solving this system using techniques like Gaussian elimination, the time taken for computing the solution varies directly with the cube of the number of degrees of freedom (i.e. $O(N^3)$). The memory requirement is also directly proportional to the square of the number of degrees of freedom ($O(N^2)$). One can reduce the time taken by the solver by using iterative schemes like Gauss-Siedel, BiCG, and GMRES etc. By using iterative schemes the time taken is reduced to the order $O(N^2)$. But the memory requirement variation stays of the order $O(N^2)$. The memory requirement for each scheme changes as the overheads are different for each scheme. The fast multipole method, used to carry out matrix vector multiplication (amidst the usage of the iterative solver) reduces the solution time to the order of $O(N)$. The memory requirement is also smaller ($O(N)$).

3.2 Fast Matrix-Vector Multiplication

Matrix-vector multiplication is an essential component of any iterative scheme which finds solution to the system $Ax = b$. Setting up the matrix-vector multiplication in the simplest form, an algorithm can be written like the one below:

Consider a matrix $A$, which has to be multiplied by a vector $x$

Let $v$ be the resultant vector.
for i = 1:1:n 
    for j = 1:1:n 
        \( v(i) = v(i) + A(i,j)x(j); \)
    end 
end 

Thus there are two nested loops. The number of operations performed in this calculation is proportional to \( n^2 \) (\( O(N^2) \)).

But if there is knowledge about the matrix the calculation can be done using algorithms which involve fewer calculations. For example if the matrix is a diagonal matrix the algorithm can be written as:

for i = 1:1:n 
    \( v(i) = A(i,i)x(i); \)
end 

Similarly if the matrix \( A \) can be given by \( A(i,j) = (y_i - z_j)^2 \) where \( y \) and \( z \) are two vectors.

Then,

\[
A(i,j)x(j) = (y_i - z_j)^2 x(j) \\
= y_i^2 x(j) - 2y_i z_j x(j) + z_j^2 x(j)
\]

Therefore,

\[
v_i = y_i^2 \sum x_j - 2y_i \sum z_j x_j + \sum z_j^2 \phi_j
\]

\[
v_i = Py_i^2 - 2Qy_i + R
\]

Thus every \( v(i) \) can be written as a function of \( y(i) \) and three constants \( P, Q \) and \( R \). The algorithm can be written as:
for \( i = 1:1:n \)
\[
P = P + x(i); \\
Q = Q + z(i)x(i); \\
R = R + (z(i))^2x(i); \\
\]
end

for \( i = 1:1:n \)
\[
v(i) = P(y(i))^2 - 2Qy(i) + R; \\
\]
end

There are two loops here but they are not nested. The number of operations performed in this calculation is proportional to \( n \ (O(N)) \). Note that in the calculation \( A(i,j) \) is not used and hence the evaluation of the entire matrix is not required. The vectors \( x \) and \( y \) are used instead of the matrix \( A \).

Fast multipole method is one algorithm to do the fast matrix-vector multiplication. The above example of fast matrix-vector multiplication has some key concepts. In general they can be summarized as:

1) The matrix \( A \) is not present in the calculation.

2) \( A \) is expressed as a function of two vectors \( y \) and \( z \). In general, the matrix can be written as a function of one or more vectors.

3) In the resultant vector \( v \), every \( v(i) \) can be written as a function of \( y(i) \) alone and does not depend on any other term from \( y \).
3.3 Iterative Solvers and Fast Matrix-Vector Multiplication

Analytical or exact methods for finding solution to the system \( Ax = b \) require a long time to get results \( (O(N^3)) \). Iterative or numerical methods to solve the same system were devised to reduce this large computational time. Many iterative schemes have been developed. All the schemes can be explained by the following steps:

1) Start with an initial guess for the solution vector \( x \) (say \( x_0 \))
2) Compute the residual vector \( r_0 = Ax_0 - b \).
3) Create another vector \( \delta x = f(r) \). The computation for the vector \( \delta x \) is based on the iterative scheme. A common way of doing this is \( M\delta x = r \). Here \( M \) is a matrix, called the preconditioner matrix. If \( M \) is identical to \( A \), a single iteration is required to solve the problem. In general, the preconditioner matrix \( M \) is similar to \( A \).
4) Update the solution vector \( x_n = x_{n-1} + \delta x \)
5) Find the residual vector \( r_n = Ax_n - b \)
6) If \( r_n \) lies within the given tolerance, \( x_n \) is the solution, else go back to step 3.

In the above method at every step the computation \( Ax_n \) is performed. The multiplication of matrix and vector generally takes \( O(N^2) \) operations. All the other computations are of the nature \( O(N) \). Hence if the matrix-vector multiplication matrix can be done by using an algorithm which takes \( O(N) \) steps, the entire operation of solving the system becomes an \( O(N) \) process.

3.4 GMRES Method

In the code for solving the 2-D Acoustic problem, the iterative scheme called Generalized Minimal Residual Method (GMRES) has been employed. Saad and Schultz [27] developed
this method for solving non-symmetric linear systems. The GMRES method is based on the Krylov subspaces. The residuals $r_0$, $Ar_0$, $A^2r_0$ form an orthogonal space span. In the code developed in this research, the GMRES algorithm which has been used is based on the algorithm developed by Barrett et al [29]. A large amount of memory is required as it involves storing the residual at each step. The GMRES algorithm uses restarts for this reason. The number of iterations required for convergence is not very large for systems which are positive definite or nearly positive definite.

The algorithm for GMRES is as follows

1) Start with an initial guess $x_0$.

2) Compute the residual vector $Mr_0 = Ax_0 - b$

3) The basis $v$ is formed explicitly using Gram-Schmidt orthogonalization. This is also called Arnoldi process. Vector $x$ is created using this basis. A least square technique is applied to update the vector $x$.

4) Check for convergence and continue.

3.5 Fast Multipole Method

The fast multipole method was developed by Greengard and Rokhlin [2] to do fast matrix-vector multiplication method. The algorithm falls into the wide category of tree-based algorithms. Algorithms like Barnes-Hut were developed to solve multi-body gravity problems. Algorithms like fast Fourier transform etc. were developed to solve problems in vibrations etc. All of them involve creating a tree and doing common calculations higher up in the tree structure. Fast multipole method also is based on a similar concept. The points (nodes) are stored in a tree structure in a method which is based on the spatial location. A
number of points are stored in one cell. Every point to point interaction is no longer necessary here. Cell to cell interactions and cell to point interactions are performed. Consider \( n \) points (Fig. 5). The total number of interactions in this case is \( n(n-1)/2 \).

![Figure 5. Multiple node to node interaction](image)

Now if they are divided spatially into 2 sets with \( n1 \) and \( n2 \) points respectively, there are \( n1 \) cell to point interactions in the first cell and \( n2 \) cell to point interactions in the second cell. The number of cell to cell interactions is 1. The interactions thus reduce to \( n2+n1+1 \). This is the single level formulation (Fig. 6).

![Figure 6. Single cell to cell interaction](image)
On further enclosing multiple cells into larger cells hierarchically, the number of interactions gets reduced even more significantly. This is the multi level formulation.

To use the fast multipole method, the node to node interaction must be decomposed into equivalent combination of cell to cell interaction and point to cell interactions. This depends on the system (i.e. nature of the matrix A).

The important mathematical requirements are as below:

1) Far field expansion should exist.
2) Near field expansion should exist

Duraiswami [28] has listed the other requirements for doing the fast multipole method multiplication.

3.6 Mathematical Formulation for FMM

As discussed in Chapter 2, the numerical formulation of the problem leads to a matrix equation \( Gq = F\phi \). Based on the prescribed boundary condition the system is converted to the form \( Ax = b \).

a. Decomposition for G kernel

The G kernel for the 2-D Helmholtz problem is given by

\[
G = -\frac{i}{4} H_0^2(kr).
\]

The G kernel can be split using Graf’s equation (Refer Appendix A).

\[
G = -\frac{i}{4} \left[ O_j^{\cos}(r_i) I_j^{\cos}(r_2) + O_j^{\sin}(r_i) I_j^{\sin}(r_2) \right]
\]  

(66)
Here,

\( \alpha_1 \) = polar angle of \( z_0 \) with respect to \( z_c \)

\( \alpha_2 \) = polar angle of \( z \) with respect to \( z_c \)

\( r_1 \) = distance between \( z_0 \) and \( z_c \)

\( r_2 \) = distance between \( z \) and \( z_c \)

\[
O_j^{\cos}(r_1) = H_j^{(2)}(kr_1) \cos(j \alpha_1)
\]

(67)

\[
O_j^{\sin}(r_1) = H_j^{(2)}(kr_1) \sin(j \alpha_1)
\]

(68)

\[
I_j^{\cos}(r_2) = J_j(kr_2) \cos(j \alpha_2)
\]

(69)

\[
I_j^{\sin}(r_2) = J_j(kr_2) \sin(j \alpha_2)
\]

(70)

But each term of the matrix is \( \int_{S_0} GqdS \)

Therefore, the multipole expansion is

\[
\int_{S_0} GqdS = -\frac{i}{4} \sum_{j=-\infty}^{\infty} \left( O_j^{\cos}(r_1) M_j^{\cos}(kr_2) + O_j^{\sin}(r_1) M_j^{\sin}(kr_2) \right)
\]

(71)

where,
These two are the two formulae for the two sets of moments.

The points \( z_0 \) and \( z \) are separated by the introduction of the point \( z_c \). The moments in the equations (72) and (73), need to be computed only once. This is similar to the calculation of \( P \), \( Q \) and \( R \) in the example in section 3.2.

**b. M2M Translation**

Now consider that the collocation point has changed from \( z_c \) to \( z_{c'} \).

![Figure 8. M2M Translation](image)

The formulae for M2M translations are given by (Refer Appendix A):
\[ M^\cos_j (r_4) = \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_4) \cos((l+j)\alpha_4) M^\cos_l (r_2) + J_{l+j}(kr_4) \sin((l+j)\alpha_4) M^\sin_l (r_2) \right) \] (74)

\[ M^\sin_j (r_4) = \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_4) \sin((l+j)\alpha_4) M^\cos_l (r_2) - J_{l+j}(kr_4) \sin((l+j)\alpha_4) M^\sin_l (r_2) \right) \] (75)

where,

\[ \alpha_4 = \text{polar angle of } z_0 \text{ with respect to } z_c. \]

\[ r_4 = \text{distance between } z_0 \text{ and } z_c. \]

c. M2L Translation

Consider a point \( z_i \) which is close to the source point.

![Figure 9. M2L Translation](image)

The formulae for M2L translation are as derived in Appendix A

\[ L^\cos_j (r_6) = \sum_{l=-\infty}^{\infty} (-1)^l \left( H^{(2)}_{l+j}(kr_6) \cos((l+j)\alpha_4) M^\cos_l (r_2) + H^{(2)}_{l+j}(kr_6) \sin((l+j)\alpha_4) M^\sin_l (r_2) \right) \] (76)

\[ L^\sin_j (r_6) = \sum_{l=-\infty}^{\infty} (-1)^l \left( H^{(2)}_{l+j}(kr_6) \sin((l+j)\alpha_4) M^\cos_l (r_2) + H^{(2)}_{l+j}(kr_6) \sin((l+j)\alpha_4) M^\sin_l (r_2) \right) \] (77)
where,

\[ \alpha_6 = \text{polar angle of } z_0 \text{ with respect to } z_l \]

\[ \alpha_5 = \text{polar angle of } z_l \text{ with respect to } z_c \]

\[ r_6 = \text{distance between } z_0 \text{ and } z_l \]

\[ r_5 = \text{distance between } z_l \text{ and } z_c \]

The M2L translation converts the expression of a matrix vector product as far field expansion to that as a near field expansion. The near field formulation can be used to compute the product.

d. L2L Translation

Now consider a point \( z_f \) which is close to the source point than \( z_l \).

![Figure 10. L2L Translation](image)

The formulae for L2L translations are:
\[ L_j^{\cos}(r_8) = \sum_{l=-\infty}^{\infty} (-1)^j \left( J_{l+j}(kr_7) \cos((l + j)\alpha_7) L_i^{\cos}(r_6) + J_{l+j}(kr_7) \sin((l + j)\alpha_7) L_i^{\sin}(r_6) \right) \]  

(78)

\[ L_j^{\sin}(r_8) = \sum_{l=-\infty}^{\infty} (-1)^j \left( J_{l+j}(kr_7) \sin((l + j)\alpha_7) L_i^{\cos}(r_6) - J_{l+j}(kr_7) \cos((l + j)\alpha_7) L_i^{\sin}(r_6) \right) \]  

(79)

where,

\[ \alpha_7 = \text{polar angle of } z_p \text{ with respect to } z_i \]

\[ \alpha_8 = \text{polar angle of } z_0 \text{ with respect to } z_i \]

\[ r_7 = \text{distance between } z_p \text{ and } z_i \]

\[ r_8 = \text{distance between } z_p \text{ and } z_0 \]

Both, M2M and L2L translations are done to utilize the hierarchical tree structure. The use of tree structure is a key aspect of the FMM.

e. Translations for the F kernel

The formulation for the translations for the F kernel is similar to that of G kernel. The detailed formulae for the F kernel are derived in Appendix A.

3.7 The Fast Multipole Method Algorithm

The fast multipole method algorithm can be described as follows:
1.) **Creation of the tree:** The entire domain is enclosed in a square box. (Fig. 11)

![Figure 11. Domain enclosed in a box](image)

This is also the root level cell or level 0 cell. This cell is subdivided into four more equal cells (Fig. 12).

![Figure 12. Division of any cell](image)

These are the level 1 cells. Every subdivided cell is said to be a child cell of the original cell. The original cell is said to be the parent cell of these child cells. All the cells of level 1 which contain elements are subdivided into cells at level 2. A cell is said to
contain an element if the midpoint of the element lies within the cell. Every domain has to be subdivided till level 2 because M2L interactions cannot be performed on level 1 cells. Every cell which contains elements is continued to be subdivided into four cells until each one has fewer elements than the preset maximum number of elements per cell (Fig. 13). A cell is said to be a leaf cell if the cell has no child cells.

Figure 13. Tree structure for the domain

2.) **Upward pass**: In the next step the upward pass is performed. The moments are calculated for every leaf using the direct formula for $M_j$ (equations 72 and 73). The moments of a single leaf is the sum of the corresponding moments of all the elements below it. The moments for the other cells are carried out by doing the M2M translation
(equations 74 and 75). For every cell the moment is the summation of the M2M translations of its child cells. These moments are found for all cells whose level \( l \geq 2 \).

3.) **Downward Pass**: The actual product of \( Ax \) is calculated in this step. The cells of level \( l \) which share at least a common vertex are said to be adjacent cells at level \( l \). M2L translations cannot be performed on adjacent cells. Two cells whose parents are adjacent but the cells themselves are not adjacent are called well separated cells. The list of all well separated cells of a particular cell is called the interaction list of that cell. M2L interactions can be carried out with each cell of the interaction list. At level 2, the downward moment is computed by adding the corresponding moments formed from all of the cells in the interaction list of that cell. From level 3 onwards, the L2L interaction with its parent is added to this sum to get the downward moment at that point. This process is carried out till one reaches the leaf.

4.) **Evaluation of the integral**: The near field formula is used to evaluate the integral inside the leaf. The direct formula for computation of the product (\( Fq \) or \( G\phi \)) is used for elements in the leaf and elements in the adjacent cells. The sum is obtained by adding the near field expansion to the summation from the direct formula.

3.8 **Application to the BEM Problem**

The process of finding the matrix-vector product is carried out for both the \( F \) and \( G \) kernels. The resultant vector is obtained as the difference \( v = Gq - F\phi \). The known quantities are input in the first case and the approximation \( x \) is used in the next case.

This vector \( v \) is passed back to the GMRES solver. The resultant vector \( v \) is used to find the next approximation to the solution vector for the next iteration. Use of preconditioner accelerates the process of finding the solution to the equation system. In the code
developed, the use of a block Jacobi preconditioner is employed. Elements in one leaf are used to form a single block in the block matrix. The vector $v$ is calculated cell by cell. The use of preconditioner can be done immediately after finding the value of the components of $v$ for each cell level.
Chapter 4

Code Development

4.1 C++ Solver for Conventional BEM

The code for both the methods, conventional BEM and FMBEM, has been written using the C++ language. C++ is a middle level language and is useful for writing code involving data structures and mathematical calculations. Most of the mathematical calculation routines are standard library functions. Evaluation of Bessel functions has been done using standard library code available on www.netlib.org.

The input is provided to the solver in the form of a file input.dat. The points inside the domain, for which the pressure is to be found out, are input as points.dat. The solver writes the output in a file output.dat.

4.2 C++ GMRES Solver

A critical part of the solver is the iterative solver. The iterative scheme used is the GMRES. The GMRES solver used in this particular code has been downloaded from www.netlib.org. Some of the routines have been modified to match the overall flow of the solver developed in this work. This is possible because the GMRES solver code uses the paradigm of cpp templates. The above solver was created at the Oak Ridge National Laboratory at Tennessee.

4.3 Tree Traversal in FMM BEM Solver
Fast multipole method is a tree based algorithm. The algorithm starts with creation of a tree. The upward and downward passes involve tree traversals. The code for the tree data structure is contained in the *class tree* and *class treenode*. Very simple routines for creating, traversing and deleting the tree are included in the code for these two classes.

**a. Algorithm for Creation of the Tree**

The algorithm for creating the tree is based on the serial input of the elements to the tree creation logic. This fact can be used for extending the tree if additional elements are added. The algorithm starts with creation of a root cell. This cell corresponds to the bounding box of the entire given boundary. A quad-tree is used and hence every parent has four child cells. Each child cell corresponds to a level 1 cell. The algorithm for fast multipole method requires at least level two cells. The four child cells are created for the root. Child cells are created further for the level 1 cells. The initial subdivision of the tree cells into child cells stops at level 2. The algorithm for every new element is as below:

1) The coordinates of the new element are read. The level 2 cell for the element is determined.

2) Insert the element in that cell’s element list if it is a leaf. A leaf is a cell with no child cells. If the cell has child cells determine the child cell for the element. Repeat this procedure till the element is placed in a list.

3) If the elements in the leaf exceed a preset maximum, the leaf is subdivided into four child cells. All the elements are then distributed in these next level child cells. The parent cell’s element list is emptied. This process is repeated till every child cell has fewer elements than the preset maximum.
b. Algorithm for Upward Pass

The tree traversal algorithm is a simple recursive algorithm. The class `momentinfo` stores the upward moments for each cell.

c. Algorithm for Downward Pass

The tree traversal algorithm is also a simple recursive algorithm. The class `momentinfo` stores the downward moments for each cell also. The interaction list of each cell is stored. This helps in expediting the calculation. Similarly the list of adjacent cells for every cell is also saved. The adjacent cells may be at a different level. The calculation for such cases is generally lengthy and storing this list also expedites the solver code.

4.4 Brief Overview of the Main Classes and Their Functions for the FMM BEM

1) `Class tree` – This class is the container class for the tree data structure. Algorithms for tree traversal use this class.

2) `Class treenode` – This class contains the information about each cell like its coordinates, its children etc.

3) `Class momentinfo` - This class is the data structure used for storing moments. The code for all the translations are functions of this class.

4) `Class element` – This class contains information about the constant element used in each case.

5) `Class node` – This class is the data structure for the BEM node.
6) **Class Preconditioner1** – The class has the code for preconditioning for the GMRES solver.

7) **Class Vector1 and Class Operator1** – These classes contain the implementation for the GMRES solver.

8) **Bessel1, Bessel2, HankelSecond, Gaussintegrator** and **Struve** are classes which do some mathematical operations. These classes have been created to have easier readability for the code.

9) The file **main.cpp** has the routines for reading the input file, writing the output file. It also maintains the overall flow for the program.

10) **Gmres.h, yn.c, jn.c** are standard library files.
Chapter 5

Numerical Results

5.1 Results for 2-D Acoustic Problems

a. 2-D Acoustic Interior Domain Problems

To test the accuracy of the BIE formulation and the codes, a very simple test case has been utilized. The test is that of a cross section of an acoustic duct. The geometry of the duct is square with side length 1.00 (Fig. 14). The boundaries parallel to x axis are the walls of the duct. In the input for the solver a mixed boundary condition is defined. Velocity boundary conditions are prescribed on the two walls of the duct. Pressure boundary conditions are prescribed on the other two boundaries.

The analytical solution for this problem is a simple 1-D solution of the nature:

\[ q = 0 \]

\[ \phi = 100 + 0i \]

\[ \phi = 0 + 100i \]

Figure 14. Interior Acoustics Problem

The analytical solution for this problem is a simple 1-D solution of the nature:
The numerical results for both the conventional code and FMBEM code agree very well with this analytical value. The values for pressure at 9 equally spaced points along the centerline of the duct are found out. Here we compare the value of real part of the pressure using both the codes with the analytical value, for the wave number $k=1.5707$. The error for these values is calculated. The number of nodes used is 400. It is found that the L2 norm of the error for the conventional code is 0.000114 while the L2 norm of the error for the FMBEM code is 0.000244. The errors in both the cases are of the similar order.

<table>
<thead>
<tr>
<th>Point</th>
<th>Analytical</th>
<th>Conventional BEM code value</th>
<th>% error</th>
<th>FMBEM code value</th>
<th>% error</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>y</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>0.5</td>
<td>98.7688</td>
<td>0.0016%</td>
<td>98.7692</td>
<td>0.0004%</td>
</tr>
<tr>
<td>0.2</td>
<td>0.5</td>
<td>95.1057</td>
<td>0.0028%</td>
<td>95.1071</td>
<td>0.0016%</td>
</tr>
<tr>
<td>0.3</td>
<td>0.5</td>
<td>89.1007</td>
<td>0.0041%</td>
<td>89.1030</td>
<td>0.0026%</td>
</tr>
<tr>
<td>0.4</td>
<td>0.5</td>
<td>80.9017</td>
<td>0.0056%</td>
<td>80.9047</td>
<td>0.0037%</td>
</tr>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>70.7107</td>
<td>0.0074%</td>
<td>70.7142</td>
<td>0.0050%</td>
</tr>
<tr>
<td>0.6</td>
<td>0.5</td>
<td>58.7785</td>
<td>0.0095%</td>
<td>58.7822</td>
<td>0.0063%</td>
</tr>
<tr>
<td>0.7</td>
<td>0.5</td>
<td>45.3991</td>
<td>0.0120%</td>
<td>45.4025</td>
<td>0.0076%</td>
</tr>
<tr>
<td>0.8</td>
<td>0.5</td>
<td>30.9017</td>
<td>0.0152%</td>
<td>30.9042</td>
<td>0.0080%</td>
</tr>
<tr>
<td>0.9</td>
<td>0.5</td>
<td>15.6435</td>
<td>0.0206%</td>
<td>15.6441</td>
<td>0.0040%</td>
</tr>
</tbody>
</table>

The study for convergence of the results has also been conducted. The same problem has been modeled with different number of nodes. The value of the real part of the pressure at

$$\phi = \phi_0 e^{ikx}$$

$$\phi_0 = 100$$
the center of the square is evaluated. In the table below, the % error with the increasing number of nodes is shown. The accuracy increases with the increasing number of nodes.

Table 2 Convergence of the pressure value at the center

<table>
<thead>
<tr>
<th>No. of nodes</th>
<th>Conventional BEM code value</th>
<th>% error</th>
<th>FMBEM code value</th>
<th>% error</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>70.8012</td>
<td>0.1280%</td>
<td>70.7914</td>
<td>0.1141%</td>
</tr>
<tr>
<td>400</td>
<td>70.7159</td>
<td>0.0074%</td>
<td>70.7149</td>
<td>0.0060%</td>
</tr>
<tr>
<td>1000</td>
<td>70.7114</td>
<td>0.0010%</td>
<td>70.7112</td>
<td>0.0008%</td>
</tr>
</tbody>
</table>

The interior pressure distribution for a constant value of \( \phi =100 \) and varying values of \( k=1.5707, 7.8539, 39.25 \) is plotted (Fig. 15). For the case of \( k=1.570796 \), the resultant domain has a quarter wavelength. This has been called the ‘Quarter Wavelength Problem’ in the following discussion.

![a) Interior Acoustic Problem for k=1.5707](image-url)
b) Interior Acoustic Problem for $k=7.8539$

c) Interior Acoustic Problem for $k=39.25$

Figure 15. Results for an interior Acoustic problem

b. 2-D Acoustic Exterior Domain Problems

The next case considered is an exterior domain problem. The boundary conditions are specified and the values of pressure are found in the exterior domain.
The numerical example uses a pulsating cylinder of radius =1.0. The analytical solution for the problem is

$$\phi = \phi_0 H_0^2(kr)$$

This solution is in polar coordinates. A constant pressure condition is prescribed in the problem (Fig. 16). The numerical results agree with the analytical results. In this case we use a pressure condition of $\phi = 100$ on the entire boundary. The value of pressure is calculated at some points in the domain for the value of wave number $k=1$. The plot is only of a region of the infinite domain.
The problem has been solved using both conventional BEM and FMBEM code. In the table below the cylinder is modeled using 100 nodes. A wave number of $k=1$ has been used.

Solving the case analytically we obtain $\phi_0 = 128.97 + 14.875\,i$

The comparison of the values is given in table 3.

**Table 3 Comparison of the values along a radial line of the cylinder**

<table>
<thead>
<tr>
<th>Point</th>
<th>Analytical value</th>
<th>FMBEM code value</th>
<th>Conventional BEM code value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>x</td>
<td>y</td>
<td>Real part</td>
</tr>
<tr>
<td>1.1</td>
<td>0</td>
<td>95.2215</td>
<td>-10.2096</td>
</tr>
<tr>
<td>1.2</td>
<td>0</td>
<td>89.9484</td>
<td>-19.4326</td>
</tr>
<tr>
<td>1.3</td>
<td>0</td>
<td>84.2344</td>
<td>-27.7304</td>
</tr>
<tr>
<td>1.4</td>
<td>0</td>
<td>78.1333</td>
<td>-35.1461</td>
</tr>
<tr>
<td>1.5</td>
<td>0</td>
<td>71.6992</td>
<td>-41.7107</td>
</tr>
<tr>
<td>1.6</td>
<td>0</td>
<td>64.9869</td>
<td>-47.4481</td>
</tr>
<tr>
<td>1.7</td>
<td>0</td>
<td>58.0519</td>
<td>-52.3776</td>
</tr>
<tr>
<td>1.8</td>
<td>0</td>
<td>50.9498</td>
<td>-56.5168</td>
</tr>
<tr>
<td>1.9</td>
<td>0</td>
<td>43.7363</td>
<td>-59.8825</td>
</tr>
</tbody>
</table>

The plot for the above table is as below:
Figure 18. Pressure Plot for Exterior Acoustic case k=1

The 2-D pressure variation in the external domain has been plotted for different values of $k = 2, 5$ and 10 respectively (Fig. 19).

a) Exterior Acoustic Problem k=2
b) Exterior Acoustic Problem $k=5$

c) Exterior Acoustic Problem $k=10$

Figure 19. Results for an Exterior Acoustic problem

c. 2-D Acoustic Scattering Problems

A scattering problem by a cylinder has been considered in this discussion. The cylinder serves as an obstacle to the passing plane wave. The velocity condition is prescribed on the cylinder boundary $q = 0$. The incident pressure wave is a traveling wave given by the equation $\phi' = \phi_0 e^{ikx}$. The wave number $k=0.5$

The use of conventional BEM solver and FMBEM solver was done to find the solution.
The comparison has been done with the analytical solution for the pressure at a field point (4, 0). The analytical value is given by [50]:

$$
\phi = \phi' \left( 1 - \sum_{n=0}^{\infty} \varepsilon_n i^n \frac{J_n'(ka)H_n'(kr)}{H_n^{(1)}(ka)} \cos(n\phi) \right)
$$

where $a$ = radius of the cylinder

and $\varepsilon_n = 1$ for $n=0$

=2 otherwise.

The value -75.008784 by the conventional BEM solver and -75.001032 by the FMBEM solver compares well with the analytical value of -75.03 at the point (4,0).

The pressure for the field points for the same problem is plotted. (Refer Fig. 20.)

![Figure 20. Results for an acoustic scattering problem](image-url)
5.2 Comparison with the conventional BEM code

In this section, the CPU time for the computation will be studied. The conventional BEM code using the same GMRES solver is taken as a guideline. Test cases of the quarter wavelength problem (discussed in section 5.1) with different number of nodes for the problem were created. The tests were run on a PC with 3.6 GHz CPU and, 2.8 GB RAM. The computational times required using both the solvers are plotted (Refer Fig. 21).

![Plot of computational time Vs. number of nodes](image)

**Figure 21. Computational time for the FMM solver**

The conventional solver can only solve cases with up to 4000 nodes. The FMBEM solver can solve cases with a maximum of 100,000 nodes. The memory required for the conventional BEM solver is of the order $O(N^2)$. The memory required for the FMM solver is of the order $O(N)$. Hence only small models can be solved by the conventional BEM
code. The slope for the FMBEM solver is 1.1360 while slope of the conventional solver was 1.9053. The slopes have been calculated by using a least square approximation for the curves. This is quite close to the expected slope of 1.0. The conventional solver is faster in the cases with fewer numbers of nodes (about 250). The FMM solver is faster for cases with more than about 250 nodes.
Chapter 6

Discussions and Future Work

In the previous chapters, the fast multipole method for the 2-D acoustic problem was discussed. The fast multipole method gives accurate results for the different classes of Acoustic problems. The fast multipole method also is much faster for larger sized models.

The graph of computational time vs. number of nodes for the FMM solver has a slope of 1.1360. In other words, the computational time is directly proportional to \( N^{1.1} \). This is very close to the theoretical variation of \( O(N) \). There are some overheads involved in the solver like the formation of the tree structure. The convergence time also varies due to the GMRES solver used. The GMRES solver used requires more steps for convergence as number of nodes increases.

The conventional solver is faster than the FMBEM solver for small number of nodes. The calculation of moments uses a \(+p\) to \( -p\) approximation. This results in \( p^2 \) calculations being performed at each step. Consequently \( kp^2N \) calculations are performed for finding the product where \( k \) depends on the number of levels in the tree structure. Hence the crossover point (number of nodes for which both the solvers take same time) depends largely on the value of \( p \). The crossover occurs at about 250 nodes.

The number of elements per leaf is a critical value. Increasing the number of elements per leaf decreases the number of levels in the tree. This does not necessarily decrease the computational time since direct calculation needs to be done for all the elements in a leaf and thus it increases the corresponding computational time. The optimum number of elements per leaf was found to be 25 for the cases studied.
Exterior acoustic problems have better conditioned matrix than interior acoustic problems. Hence the number of iterations for convergence is much lesser in the exterior case. Despite being better than the regular BEM code, there is scope for further improvement of the FMBEM solver. The decomposition of the $F$ and $G$ kernels has been made using two sets of moments. Since the two sets of moments are so closely related, it must be possible to use a single set by the use of complex numbers. The current formulation is simplistic and does not consider the fictitious eigen frequencies. The hypersingular formulation for the same problem with the use of Burton-Miller BIE should solve the problem of fictitious frequencies for exterior acoustic problems. The formulation for higher-order elements should also be possible. The higher-order elements reduce the total number of elements for the same accuracy.

In all, this is the first step of the formulation of FMM BEM for 2-D Acoustics. It can be extended to encompass a large number of other problems with the same or better efficiency.
References

1) I Fredholm (1903) "Sur une classe d’équations fonctionnelles". Acta Mathematica.


Topics in Ocean Engineering 2 :143-163


Nature 324: 446–449.


29) Barrett "Templates for the Solution of Linear Systems Building Blocks for Iterative Methods."


Appendix A.

Derivation of the Multipole Coefficients and Functions

Graf’s equation

\[ C_v(w) \cos(v \chi) = \sum_{j=-\infty}^{\infty} C_{v+j}(u) J_j(v) \cos(j \alpha) \]  

(A.1)

Here,

\( C_v() \) is any Bessel function or Hankel function of integer order \( v \)

\( J_v() \) is any Bessel function of integer order \( v \)

The equation has been taken in the form discussed in Abramowitz and Stegun [51].

Substitute \( v = 0 \) and \( C = H^{(2)} \) (Hankel function of the second kind)

\[ H_0^{(2)}(w) = \sum_{j=0}^{\infty} H_j^{(2)}(u) J_j(v) \cos(j \alpha) \]  

(A.2)

\[ G = -\frac{i}{4} \sum_{j=-\infty}^{\infty} H_j^{(2)}(k r_1) J_j(k r_2) \cos(j \alpha) \]  

(A.3)
\[ G = \frac{-i}{4} \sum_{j=-\infty}^{\infty} H_j^{(2)}(kr_1) J_j(kr_2) \cos(j(\alpha_1 - \alpha_2)) \quad (A.4) \]

\[ = \frac{-i}{4} \sum_{j=-\infty}^{\infty} \left( H_j^{(2)}(kr_1) J_j(kr_2) \cos(j\alpha_1) \cos(j\alpha_2) + \sin(j\alpha_1) \sin(j\alpha_2) \right) \]

\[ = \frac{-i}{4} \left[ \sum_{j=-\infty}^{\infty} H_j^{(2)}(kr_1) \cos(j\alpha_1) J_j(kr_2) \cos(j\alpha_2) + H_j^{(2)}(kr_1) \sin(j\alpha_1) J_j(kr_2) \sin(j\alpha_2) \right] \]

\[ = \frac{-i}{4} \left[ \sum_{j=-\infty}^{\infty} H_j^{(2)}(kr_1) \cos(j\alpha_1) J_j(kr_2) \cos(j\alpha_2) + \sum_{j=-\infty}^{\infty} H_j^{(2)}(kr_1) \sin(j\alpha_1) J_j(kr_2) \sin(j\alpha_2) \right] \quad (A.5) \]

Substituting the terms

\[ I_j^{\cos}(r_2) = J_j(kr_2) \cos(j\alpha_2) \quad \text{(A.6)} \]

\[ I_j^{\sin}(r_2) = J_j(kr_2) \sin(j\alpha_2) \quad \text{(A.7)} \]

Hence,

\[ G = \frac{-i}{4} \left[ \sum_{j=-\infty}^{\infty} H_j^{(2)}(kr_1) \cos(j\alpha_1) I_j^{\cos}(r_2) + \sum_{j=-\infty}^{\infty} H_j^{(2)}(kr_1) \sin(j\alpha_1) I_j^{\sin}(r_2) \right] \]

\[ = \frac{-i}{4} \left[ H_j^{(2)}(kr_1) \cos(j\alpha_1) I_j^{\cos}(r_2) + H_j^{(2)}(kr_1) \sin(j\alpha_1) I_j^{\sin}(r_2) \right] \quad (A.8) \]

Substituting the terms

\[ O_j^{\cos}(r_1) = H_j^{(2)}(kr_1) \cos(j\alpha_1) \quad \text{(A.9)} \]
\[ O_j^{\sin}(r_1) = H_j^{(2)}(kr_1) \sin(j\alpha) \]  
(A.10)

we get,

\[ G = -\frac{i}{4} \left[ O_j^{\cos}(r_1) I_j^{\cos}(r_2) + O_j^{\sin}(r_1) I_j^{\sin}(r_2) \right] \]  
(A.11)

So the multipole expansion is

\[
\int G \, q \, dS = -\frac{i}{4} \sum_{j=-\infty}^{\infty} \left( O_j^{\cos}(r_1) M_j^{\cos}(kr_2) + O_j^{\sin}(r_1) M_j^{\sin}(kr_2) \right) 
\]  
(A.12)

where,

\[ M_j^{\cos}(r_2) = \int_{S_0} J_j(kr_2) \cos(j\alpha_2) \, q \, dS \]  
(A.13)

\[ M_j^{\sin}(r_2) = \int_{S_0} J_j(kr_2) \sin(j\alpha_2) \, q \, dS \]  
(A.14)

\[ M_j^{\cos}(r_2) = \int_{S_0} I_j^{\cos}(r_2) \, q \, dS \]  
(A.15)

\[ M_j^{\sin}(r_2) = \int_{S_0} I_j^{\sin}(r_2) \, q \, dS \]  
(A.16)
Now consider another collocation point $z'_c$.

\[
M_j^{\cos}(r_3) = \int_{S_0} I_j^{\cos}(r_3)qdS \tag{A.17}
\]

\[
M_j^{\cos}(r_3) = \int_{S_0} J_j(r_3) \cos(k\alpha_3)qdS \tag{A.18}
\]

\[
M_j^{\sin}(r_3) = \int_{S_0} J_j(r_3) \sin(k\alpha_3)qdS \tag{A.19}
\]

where,

\[
\alpha_3 = \text{polar angle of } z \text{ with respect to } z'_c
\]

\[
r_3 = \text{distance between } z \text{ and } z'_c
\]
Now consider the term \( J_j(kr_z) \cos(j\alpha_z) \)

\[
J_j(kr_z) \cos(j\alpha_z) = J_j(kr_z) \cos(j(\alpha_4 + \chi))
\]

where,

\[
\alpha_4 = \text{polar angle of } z_c \text{ with respect to } z_c \\
r_4 = \text{distance between } z_c \text{ and } z_c
\]

\[
J_j(kr_z) \cos(j\alpha_z) = J_j(kr_z)(\cos(j\alpha_4) \cos(j\chi) - \sin(j\alpha_4) \sin(j\chi))
\]

\[
J_j(kr_z) \cos(j\alpha_z) = J_j(kr_z) \cos(j\chi) \cos(j\alpha_4) - J_j(kr_z) \sin(j\chi) \sin(j\alpha_4)
\]  \(\text{(A.20)}\)

\[
= \sum_{l=-\infty}^{\infty} J_{l+j}(kr_z) J_j(kr_z) \cos(l(\alpha_4 - \alpha_2 + 180)) \cos(j\alpha_4)
\]

\[- \sum_{l=-\infty}^{\infty} J_{l+j}(kr_z) J_j(kr_z) \sin(l(\alpha_4 - \alpha_2 + 180)) \sin(j\alpha_4)
\]

\[
= \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_z) J_j(kr_z) \cos(l\alpha_4) \cos(l\alpha_2) \cos(j\alpha_4) \right)
\]

\[- \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_z) J_j(kr_z) \sin(l\alpha_4) \cos(l\alpha_2) \sin(j\alpha_4) \right)
\]

\[- \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_z) J_j(kr_z) \cos(l\alpha_4) \sin(l\alpha_2) \cos(j\alpha_4) \right)
\]

\[- \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_z) J_j(kr_z) \sin(l\alpha_4) \sin(l\alpha_2) \sin(j\alpha_4) \right)
\]

\[
= \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_z) J_j(kr_z) \cos((l+j)\alpha_4) \cos(l\alpha_2) + J_{l+j}(kr_z) J_j(kr_z) \sin((l+j)\alpha_4) \sin(l\alpha_2) \right)
\]

\[
= \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_z) \cos((l+j)\alpha_4) J_j \cos(r_2) + J_{l+j}(kr_z) \sin((l+j)\alpha_4) J_j \sin(r_2) \right)
\]  \(\text{(A.21)}\)

Substituting into the equation (A.17) we get,
\[ M_j^{\cos}(r_j) = \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_3) \cos((l+j)\alpha_4) M_l^{\cos}(r_2) + J_{l+j}(kr_3) \sin((l+j)\alpha_4) M_l^{\sin}(r_2) \right) \]

(A.22)

Similarly,

\[ J_j(kr_5) \sin(j\alpha_3) = J_j(kr_5) \sin(j(\alpha_4 + \chi)) \]

\[ J_j(kr_5) \sin(j\alpha_3) = J_j(kr_5)(\sin(j\alpha_4) \cos(j\chi) + \cos(j\alpha_4) \sin(j\chi)) \]

\[ J_j(kr_5) \cos(j\alpha_3) = J_j(kr_5) \cos(j\chi) \sin(j\alpha_4) + J_j(kr_5) \sin(j\chi) \cos(j\alpha_4) \]  

(A.23)

\[ = \sum_{l=-\infty}^{\infty} J_{l+j}(kr_3) J_i(kr_2) \cos(l(\alpha_4 - \alpha_2 + 180)) \sin(j\alpha_4) \]

\[ + \sum_{l=-\infty}^{\infty} J_{l+j}(kr_3) J_i(kr_2) \sin(l(\alpha_4 - \alpha_2 + 180)) \cos(j\alpha_4) \]

\[ = \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_3) J_i(kr_2) \cos(l\alpha_4) \cos(l\alpha_2) \sin(j\alpha_4) \right) \]

\[ + \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_3) J_i(kr_2) \sin(l\alpha_4) \sin(l\alpha_2) \sin(j\alpha_4) \right) \]

\[ = \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_3) J_i(kr_2) \sin((l+j)\alpha_4) \cos(l\alpha_2) - J_{l+j}(kr_3) J_i(kr_2) \cos((l+j)\alpha_4) \sin(l\alpha_2) \right) \]

\[ = \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_3) \sin((l+j)\alpha_4) J_i(kr_2) \cos(l\alpha_2) - J_{l+j}(kr_3) \cos((l+j)\alpha_4) J_i(kr_2) \sin(l\alpha_2) \right) \]

\[ = \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_3) \sin((l+j)\alpha_4) I_i^{\cos}(r_2) - J_{l+j}(kr_3) \cos((l+j)\alpha_4) I_i^{\sin}(r_2) \right) \]  

(A.24)

Substituting into the equation (A.19) we get,

\[ M_j^{\sin}(r_j) = \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_3) \sin((l+j)\alpha_4) M_l^{\cos}(r_2) - J_{l+j}(kr_3) \sin((l+j)\alpha_4) M_l^{\sin}(r_2) \right) \]
Now by equation (A.9),

\[
O_j^{\cos}(r_i) = H_i^{(2)}(kr_i) \cos(j\alpha_i) \tag{A.9}
\]

\[
O_j^{\cos}(r_i) = H_i^{(2)}(kr_i) \cos(j(\alpha_5 + \chi))
\]

\[
O_j^{\cos}(r_i) = H_i^{(2)}(kr_i)(\cos(j\alpha_5) \cos(j\chi) - \sin(j\alpha_5) \sin(j\chi))
\]

\[
O_j^{\cos}(r_i) = H_i^{(2)}(kr_i) \cos(j\chi) \cos(j\alpha_5) - H_i^{(2)}(kr_i) \sin(j\chi) \sin(j\alpha_5) \tag{A.25}
\]

\[
= \sum_{l=-\infty}^{\infty} H_{i+j}^{(2)}(kr_5) J_l(kr_6) \cos(l(\alpha_5 - \alpha_6 + 180)) \cos(j\alpha_5)
\]

\[
- \sum_{l=-\infty}^{\infty} H_{i+j}^{(2)}(kr_5) J_l(kr_6) \sin(l(\alpha_5 - \alpha_6 + 180)) \sin(j\alpha_5)
\]

\[
= \sum_{l=-\infty}^{\infty} (-1)^l \left( H_{i+j}^{(2)}(kr_5) J_l(kr_6) \cos(l\alpha_5) \cos(l\alpha_6) \cos(j\alpha_5) + H_{i+j}^{(2)}(kr_5) J_l(kr_6) \sin(l\alpha_5) \sin(l\alpha_6) \cos(j\alpha_5) \right)
\]

\[
+ \sum_{l=-\infty}^{\infty} (-1)^l \left( H_{i+j}^{(2)}(kr_5) J_l(kr_6) \sin(l\alpha_5) \cos(l\alpha_6) \sin(j\alpha_5) - H_{i+j}^{(2)}(kr_5) J_l(kr_6) \cos(l\alpha_5) \sin(l\alpha_6) \sin(j\alpha_5) \right)
\]

\[
= \sum_{l=-\infty}^{\infty} (-1)^l \left( H_{i+j}^{(2)}(kr_5) J_l(kr_6) \cos((l + j)\alpha_5) \cos(l\alpha_6) + H_{i+j}^{(2)}(kr_5) J_l(kr_6) \sin((l + j)\alpha_5) \sin(l\alpha_6) \right)
\]
\[
\sum_{l=-\infty}^{\infty} (-1)^l \left( H_{1+j}^{(2)}(kr) \cos((l + j)\alpha_6) J_{l}(kr) \cos(l\alpha_6) + H_{1+j}^{(2)}(kr) \sin((l + j)\alpha_6) J_{l}(kr) \sin(l\alpha_6) \right)
\]

\[
= \sum_{l=-\infty}^{\infty} (-1)^l \left( H_{1+j}^{(2)}(kr) \cos((l + j)\alpha_6) I_l^\cos(r_6) + H_{1+j}^{(2)}(kr) \sin((l + j)\alpha_6) I_l^\sin(r_6) \right)
\]

\[
= \sum_{l=-\infty}^{\infty} (-1)^l \left( O_{i+j}^\cos(r_6) I_l^\cos(r_6) + O_{i+j}^\sin(r_6) I_l^\sin(r_6) \right)
\]

\[
L_j^\cos(r_6) = \sum_{l=-\infty}^{\infty} (-1)^l \left( H_{1+j}^{(2)}(kr) \cos((l + j)\alpha_6) M_l^\cos(r_6) + H_{1+j}^{(2)}(kr) \sin((l + j)\alpha_6) M_l^\sin(r_6) \right)
\]

By equation (A.10)

\[
O_j^\sin(r_1) = H_{j}^{(2)}(kr_1) \sin(j\alpha_1) \tag{A.10}
\]

\[
O_j^\sin(r_1) = H_{j}^{(2)}(kr_1) \sin(j(\alpha_2 + \chi))
\]

\[
O_j^\sin(r_1) = H_{j}^{(2)}(kr_1)(\sin(j\alpha_5) \cos(j\chi) + \cos(j\alpha_5) \sin(j\chi))
\]

\[
O_j^\sin(r_1) = H_{j}^{(2)}(kr_1) \cos(j\chi) \sin(j\alpha_5) + H_{j}^{(2)}(kr_1) \sin(j\chi) \cos(j\alpha_5) \tag{A.27}
\]

\[
= \sum_{l=-\infty}^{\infty} H_{1+j}^{(2)}(kr_1) J_{l}(kr) \cos(l(\alpha_5 - \alpha_6 + 180)) \sin(j\alpha_5)
\]

\[
+ \sum_{l=-\infty}^{\infty} H_{1+j}^{(2)}(kr_1) J_{l}(kr) \sin(l(\alpha_5 - \alpha_6 + 180)) \cos(j\alpha_5)
\]

\[
= \sum_{l=-\infty}^{\infty} (-1)^l \left( H_{1+j}^{(2)}(kr_1) J_{l}(kr) \cos(l\alpha_5) \sin(l\alpha_6) \sin(j\alpha_5) \right)
\]

\[
+ \sum_{l=-\infty}^{\infty} (-1)^l \left( H_{1+j}^{(2)}(kr_1) J_{l}(kr) \sin(l\alpha_5) \cos(l\alpha_6) \sin(j\alpha_5) \right)
\]

\[
- \sum_{l=-\infty}^{\infty} (-1)^l \left( H_{1+j}^{(2)}(kr_1) J_{l}(kr) \sin(l\alpha_5) \cos(l\alpha_6) \cos(j\alpha_5) \right)
\]

\[
- H_{1+j}^{(2)}(kr_1) J_{l}(kr) \cos(l\alpha_5) \sin(l\alpha_6) \cos(j\alpha_5) \right)
\]

\[
= \sum_{l=-\infty}^{\infty} (-1)^l \left( H_{1+j}^{(2)}(kr_1) J_{l}(kr) \sin((l + j)\alpha_5) \cos(l\alpha_6) + H_{1+j}^{(2)}(kr_1) J_{l}(kr) \cos((l + j)\alpha_5) \sin(l\alpha_6) \right)
\]
\[
= \sum_{l=-\infty}^{\infty} (-1)^l \left( H_{i+j}^{(2)}(kr_2) \sin((l + j)\alpha_5)J_1(kr_6)\cos(l\alpha_6) - H_{i+j}^{(2)}(kr_2) \cos((l + j)\alpha_5)J_1(kr_6)\sin(l\alpha_6) \right)
\]

\[
= \sum_{l=-\infty}^{\infty} (-1)^l \left( H_{i+j}^{(2)}(kr_2) \sin((l + j)\alpha_5)I_l^\cos(r_6) - H_{i+j}^{(2)}(kr_2) \cos((l + j)\alpha_5)I_l^\sin(r_6) \right)
\]

\[
= \sum_{l=-\infty}^{\infty} (-1)^l \left( O_{i+j}^\sin(r_5)I_l^\cos(r_6) - O_{i+j}^\cos(r_5)I_l^\sin(r_6) \right)
\]

\[
L_j^\sin(r_6) = \sum_{l=-\infty}^{\infty} (-1)^l \left( H_{i+j}^{(2)}(kr_2) \sin((l + j)\alpha_5)M_l^\cos(r_2) + H_{i+j}^{(2)}(kr_2) \sin((l + j)\alpha_5)M_l^\sin(r_2) \right)
\]

(A.28)
\[
J_j(kr_\theta) \cos(j\alpha_\theta) = J_j(kr_\theta) \cos(j(\alpha_\gamma - \chi + 180))
\]
\[
= (-1)^j J_j(kr_\theta) \cos(j\alpha_\gamma) \cos(j\chi) + \sin(j\alpha_\gamma) \sin(j\chi)
\]
\[
= (-1)^j J_j(kr_\theta) \cos(j\chi) \cos(j\alpha_\gamma) + (-1)^j J_j(kr_\theta) \sin(j\chi) \sin(j\alpha_\gamma)
\]
\[
= (-1)^j \sum_{l=-\infty}^{\infty} J_{i+j}(kr_\gamma) J_i(kr_\theta) \cos(l(\alpha_\theta - \alpha_\gamma)) \cos(j\alpha_\gamma)
\]
\[
+ (-1)^j \sum_{l=-\infty}^{\infty} J_{i+j}(kr_\gamma) J_i(kr_\theta) \sin(l(\alpha_\theta - \alpha_\gamma)) \sin(j\alpha_\gamma)
\]
\[
= \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{i+j}(kr_\gamma) J_i(kr_\theta) \cos(l\alpha_\theta) \cos(l\alpha_\gamma) \cos(j\alpha_\gamma) \right)
\]
\[
+ J_{i+j}(kr_\gamma) J_i(kr_\theta) \sin(l\alpha_\theta) \sin(l\alpha_\gamma) \cos(j\alpha_\gamma)
\]
\[
+ \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{i+j}(kr_\gamma) J_i(kr_\theta) \sin(l\alpha_\theta) \cos(l\alpha_\gamma) \sin(j\alpha_\gamma) \right)
\]
\[
- J_{i+j}(kr_\gamma) J_i(kr_\theta) \cos(l\alpha_\theta) \sin(l\alpha_\gamma) \sin(j\alpha_\gamma)
\]
\[
= \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{i+j}(kr_\gamma) J_i(kr_\theta) \cos((l + j)\alpha_\gamma) \cos(l\alpha_\theta) + H_{i+j}^{(2)}(kr_\gamma) J_i(kr_\theta) \sin((l + j)\alpha_\gamma) \sin(l\alpha_\theta) \right)
\]
\[
\sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_{7}) \cos((l+j)\alpha_{7})J_{l}(kr_{6}) \cos(l\alpha_{6}) + H^{(2)}_{l+j}(kr_{7}) \sin((l+j)\alpha_{7})J_{l}(kr_{6}) \sin(l\alpha_{6}) \right)
\]

\[
= \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_{7}) \cos((l+j)\alpha_{7})I^{\cos}_{l}(r_{6}) + J_{l+j}(kr_{7}) \sin((l+j)\alpha_{7})I^{\sin}_{l}(r_{6}) \right)
\]

Thus we get,

\[
L^{\cos}_{j}(r_{6}) = \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_{7}) \cos((l+j)\alpha_{7})L^{\cos}_{l}(r_{6}) + J_{l+j}(kr_{7}) \sin((l+j)\alpha_{7})L^{\sin}_{l}(r_{6}) \right)
\] (A.30)

Similarly,

\[
J_{j}(kr_{8}) \sin(j\alpha_{8}) = J_{j}(kr_{8}) \sin(j(\alpha_{7} - \chi + 180))
\]

\[
= (-1)^j J_{j}(kr_{8}) (\sin(j\alpha_{7}) \cos(j\chi) - \cos(j\alpha_{7}) \sin(j\chi))
\]

\[
= (-1)^j J_{j}(kr_{8}) \cos(j\chi) \sin(j\alpha_{7}) - J_{j}(kr_{8}) \sin(j\chi) \cos(j\alpha_{7})
\]

\[
= \sum_{l=-\infty}^{\infty} (-1)^l J_{l+j}(kr_{7})J_{l}(kr_{6}) \cos(l(\alpha_{6} - \alpha_{7})) \sin(j\alpha_{7})
\]

\[
- \sum_{l=-\infty}^{\infty} (-1)^l J_{l+j}(kr_{7})J_{l}(kr_{6}) \sin(l(\alpha_{6} - \alpha_{7})) \cos(j\alpha_{7})
\]

\[
= \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_{7})J_{l}(kr_{6}) \cos(l\alpha_{7}) \cos(l\alpha_{6}) \sin(j\alpha_{7}) + J_{l+j}(kr_{7})J_{l}(kr_{6}) \sin(l\alpha_{6}) \sin(l\alpha_{7}) \sin(j\alpha_{7}) \right)
\]

\[
- \sum_{l=-\infty}^{\infty} (-1)^l \left( -J_{l+j}(kr_{7})J_{l}(kr_{6}) \cos(l\alpha_{7}) \cos(l\alpha_{6}) \cos(j\alpha_{7}) \right)
\]

\[
= \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_{7})J_{l}(kr_{6}) \sin(l\alpha_{6}) \cos(l\alpha_{7}) \cos(j\alpha_{7}) - H^{(2)}_{l+j}(kr_{7})J_{l}(kr_{6}) \sin((l+j)\alpha_{7}) \sin(l\alpha_{6}) \right)
\]

\[
= \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_{7}) \sin((l+j)\alpha_{7})J_{l}(kr_{6}) \cos(l\alpha_{6}) - H^{(2)}_{l+j}(kr_{7}) \cos((l+j)\alpha_{7}) J_{l}(kr_{6}) \sin(l\alpha_{6}) \right)
\]

\[
= \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_{7}) \sin((l+j)\alpha_{7})I^{\cos}_{l}(r_{6}) - J_{l+j}(kr_{7}) \cos((l+j)\alpha_{7})I^{\sin}_{l}(r_{6}) \right)
\]
Thus we get

\[
L^\sin_j(r_\gamma) = \sum_{l=-\infty}^{\infty} (-1)^l \left( J_{l+j}(kr_\gamma) \sin((l + j)\alpha_\gamma) L^\cos_l(r_\gamma) - J_{l+j}(kr_\gamma) \cos((l + j)\alpha_\gamma) L^\sin_l(r_\gamma) \right) \quad (A.31)
\]
Moments for the $F$ kernel

The same notation will be used for the derivations of the $F$ kernel.

\[
H_1^{(2)}(kr) \cos \beta = H_1^{(2)}(kr) \cos(180 - (\alpha + \chi - \gamma)) \\
= -H_1^{(2)}(kr) \cos(\alpha + \chi - \gamma) \\
= -H_1^{(2)}(kr)(\cos(\alpha - \gamma) \cos \chi - \sin(\alpha - \gamma) \sin \chi) \\
= -H_1^{(2)}(kr) \cos \chi \cos(\alpha - \gamma) + H_1^{(2)}(kr) \sin \chi \sin(\alpha - \gamma)
\]

Again by Graf’s equation,

\[
= - \sum_{j=-\infty}^{\infty} H_{j+1}^{(2)}(kr) J_j(kr) \cos j \alpha \cos(\alpha - \gamma) + \sum_{j=-\infty}^{\infty} H_{j+1}^{(2)}(kr) J_j(kr) \sin j \alpha \sin(\alpha - \gamma) \quad (A.32)
\]

\[
= - \sum_{j=-\infty}^{\infty} H_{j}^{(2)}(kr_1) J_j(kr_2) \cos j \alpha \cos(\alpha \cos \gamma + \sin \alpha \sin \gamma) \\
+ \sum_{j=-\infty}^{\infty} H_{j+1}^{(2)}(kr_1) J_j(kr_2) \sin j \alpha \sin(\alpha \cos \gamma - \cos \alpha \sin \gamma)
\]

\[
= - \sum_{j=-\infty}^{\infty} H_{j}^{(2)}(kr_1) J_j(kr_2) \cos(\cos \alpha \cos j \alpha - \sin \alpha \sin j \alpha) \\
- \sum_{j=-\infty}^{\infty} H_{j+1}^{(2)}(kr_1) J_j(kr_2) \sin(\sin \alpha \sin j \alpha + \cos \alpha \sin j \alpha)
\]

\[
= - \sum_{j=-\infty}^{\infty} H_{j}^{(2)}(kr_1) J_j(kr_2) \cos \gamma \cos((j + 1)\alpha) - \sum_{j=-\infty}^{\infty} H_{j+1}^{(2)}(kr_1) J_j(kr_2) \sin \gamma(\sin(j + 1)\alpha)
\]

\[
= - \sum_{j=-\infty}^{\infty} H_{j}^{(2)}(kr_1) J_j(kr_2) \cos \gamma \cos((j + 1)(\alpha_1 - \alpha_2)) \\
- \sum_{j=-\infty}^{\infty} H_{j+1}^{(2)}(kr_1) J_j(kr_2) \sin \gamma(\sin(j + 1)(\alpha_1 - \alpha_2))
\]

\[
= - \sum_{j=-\infty}^{\infty} H_{j}^{(2)}(kr_1) J_j(kr_2) \cos \gamma \cos((j + 1)\alpha_1)(\cos((j + 1)\alpha_2) + \sin((j + 1)\alpha_1)\sin((j + 1)\alpha_2))
\]
\[- \sum_{j=-\infty}^{\infty} H_{j+1}^{(2)}(kr_1)J_j(kr_2) \sin \gamma [\sin((j+1)\alpha_1) \cos((j+1)\alpha_2) - \cos((j+1)\alpha_1) \sin((j+1)\alpha_2)] \]

\[= - \sum_{j=-\infty}^{\infty} H_{j+1}^{(2)}(kr_1)J_j(kr_2) \cos((j+1)\alpha_1)[\cos((j+1)\alpha_2) \cos \gamma - \sin \gamma \sin((j+1)\alpha_2)] \]

\[= - \sum_{j=-\infty}^{\infty} H_{j+1}^{(2)}(kr_1)J_j(kr_2) \sin((j+1)\alpha_1)[\cos((j+1)\alpha_2) \sin \gamma + \sin((j+1)\alpha_2) \cos \gamma] \]

\[= - \sum_{j=-\infty}^{\infty} H_{j+1}^{(2)}(kr_1)J_j(kr_2) \cos((j+1)\alpha_1) \cos((j+1)\alpha_2 + \gamma) \]

\[- \sum_{j=-\infty}^{\infty} H_{j+1}^{(2)}(kr_1)J_j(kr_2) \sin((j+1)\alpha_1) \sin((j+1)\alpha_2 + \gamma) \quad (A.33) \]

Substitute the terms

\[I_j^{\cos}(r_2) = J_j(kr_2) \cos((j+1)\alpha_2 + \gamma) \quad (A.34) \]

\[I_j^{\sin}(r_2) = J_j(kr_2) \sin((j+1)\alpha_2 + \gamma) \quad (A.35) \]

we get,

\[\frac{\partial G}{\partial n} = \frac{ik}{4} \left[ \sum_{j=-\infty}^{\infty} \left( H_{j+1}^{(2)}(kr_1) \cos((j+1)\alpha_1) I_j^{\cos}(r_2) \right) + \sum_{j=-\infty}^{\infty} \left( H_{j+1}^{(2)}(kr_1) \sin((j+1)\alpha_1) I_j^{\sin}(r_2) \right) \right] \quad (A.36) \]

Substitute the terms

\[O_j^{\cos}(r_1) = H_{j+1}^{(2)}(kr_1) \cos((j+1)\alpha_1) \quad (A.37) \]

\[O_j^{\sin}(r_1) = H_{j+1}^{(2)}(kr_1) \sin((j+1)\alpha_1) \quad (A.38) \]

we get,

\[\frac{\partial \psi}{\partial n} = -\frac{ik}{4} \left[ O_j^{\cos}(r_1) I_j^{\cos}(r_2) + O_j^{\sin}(r_1) I_j^{\sin}(r_2) \right] \]

So the multipole expansion is

\[\int_{S_o} \frac{\partial \psi}{\partial n} \phi dS = -\frac{ik}{4} \sum_{j=-\infty}^{\infty} \left( O_j^{\cos}(r_1) M_j^{\cos}(kr_2) + O_j^{\sin}(r_1) M_j^{\sin}(kr_2) \right) \quad (A.39) \]
where,

\[ M_j^{\cos}(r_2) = \int_{S_o} J_j(kr_2) \cos((j + 1)\alpha_2 + \gamma) \phi dS \]  \hspace{1cm} (A.40)

\[ M_j^{\sin}(r_2) = \int_{S_o} J_j(kr_2) \sin((j + 1)\alpha_2 + \gamma) \phi dS \]  \hspace{1cm} (A.41)

\[ M_j^{\cos}(r_2) = \int_{S_o} I_j^{\cos}(r_2) \phi dS \] \hspace{1cm} (A.42)

\[ M_j^{\sin}(r_2) = \int_{S_o} I_j^{\sin}(r_2) \phi dS \] \hspace{1cm} (A.43)
\[ J_j(kr_3 \cos((j+1)\alpha_3 + \gamma)) = J_j(kr_3) \left( \cos((j+1)\alpha_3) \cos \gamma - \sin((j+1)\alpha_3) \sin \gamma \right) \]
\[ = J_j(kr_3 \cos((j+1)\alpha_3)) \cos \gamma - J_j(kr_3 \sin((j+1)\alpha_3)) \sin \gamma \]
\[ = J_j(kr_3 \cos((j+1)(\alpha_4 + \chi))) \cos \gamma - J_j(kr_3 \sin((j+1)(\alpha_4 + \chi))) \sin \gamma \]
\[ = J_j(kr_3 \cos(j\chi + (j+1)\alpha_4 + \chi)) \cos \gamma - J_j(kr_3 \sin(j\chi + (j+1)\alpha_4 + \chi)) \sin \gamma \]
\[ = J_j(kr_3 \cos(j\chi)[\cos((j+1)\alpha_4 + \chi) - \sin((j+1)\alpha_4 + \chi)] \]
\[ - J_j(kr_3 \sin(j\chi)[\sin((j+1)\alpha_4 + \chi) + \cos((j+1)\alpha_4 + \chi)] \]
\[ = J_j(kr_3 \cos(j\chi)[\cos((j+1)\alpha_4 + \chi) - \sin((j+1)\alpha_4 + \chi)] \]
\[ - J_j(kr_3 \sin(j\chi)[\sin((j+1)\alpha_4 + \chi) + \cos((j+1)\alpha_4 + \chi)] \]
\[ = J_j(kr_3 \cos(j\chi)[\cos((j+1)\alpha_4 + \chi + \gamma)] \]
\[ - J_j(kr_3 \sin(j\chi)[\sin((j+1)\alpha_4 + \chi + \gamma)] \]
\[ = \sum_{l=-\infty}^{\infty} J_{l+j}(kr_4)J_j(kr_2) \cos(l(\alpha_4 - \alpha_2 + 180)) \cos((j+1)\alpha_4 + \chi + \gamma) \]
\[ - \sum_{l=-\infty}^{\infty} J_{l+j}(kr_4)J_j(kr_2) \sin(l(\alpha_4 - \alpha_2 + 180)) \sin((j+1)\alpha_4 + \chi + \gamma) \]
\[ = \sum_{l=-\infty}^{\infty} J_{l+j}(kr_4)J_j(kr_2) \cos(l(\alpha_4 - \alpha_2 + 180)) \cos((j+1)\alpha_4 + \chi + \gamma) \]
\[ - \sum_{l=-\infty}^{\infty} J_{l+j}(kr_4)J_j(kr_2) \sin(l(\alpha_4 - \alpha_2 + 180)) \sin((j+1)\alpha_4 + \chi + \gamma) \]
\[ = \sum_{l=-\infty}^{\infty} J_{l+j}(kr_4)J_j(kr_2) \cos((l(\alpha_4 - \alpha_2 + 180)) + ((j+1)\alpha_4 + \chi + \gamma)) \]
\[
\sum_{l=\infty}^{\infty} j_{l+j}(kr_4) j_{l}(kr_2) \cos[l \alpha_4 - l \alpha_2 + 180l + j \alpha_4 + \alpha_4 + \chi + \gamma]
\]

\[
\sum_{l=\infty}^{\infty} j_{l+j}(kr_4) j_{l}(kr_2) \cos[(l + j)\alpha_4 - l \alpha_2 + \alpha_4 + \chi + \gamma]
\]

\[
\sum_{l=\infty}^{\infty} (-1)^l j_{l+j}(kr_4) j_{l}(kr_2) \cos[(l + j)\alpha_4 - l \alpha_2 - \alpha_2 - \delta]
\]

Thus we get,

\[
M_j^{\cos}(r_3) = \sum_{l=\infty}^{\infty} (-1)^l \left( j_{l+j}(kr_4) \cos((l + j)\alpha_4) M_j^{\cos}(r_2) + j_{l+j}(kr_4) \sin((l + j)\alpha_4) M_j^{\sin}(r_2) \right)
\]

(A.44)

The M2L moments and L2L moments for the \(F\) kernel are very similar to \(G\) kernel because the \(I\) function is same for both kernels.
Appendix B

Code listing for the FMBEM solver

The list of files in the FMBEM code is:

- bessel1.h
- bessel2.h
- element.h
- Gaussintegrator.h
- gmres.h
- HankelSecond.h
- InteractionList.h
- InteractList.h
- leafnodelist.h
- mconf.h
- momentinfo.h
- node.h
- nodeinfo.h
- nodelist.h
- Operator1.h
- Preconditioner1.h
- resource.h
- Struve.h
- templates.h
- tree1.h
- treenode.h
- Vector1.h
- yn.c
- jn.c
• bessel1.cpp
• bessel2.cpp
• element.cpp
• Gaussintegrator.cpp
• HankelSecond.cpp
• InteractionList.cpp
• InteractList.cpp
• leafnodelist.cpp
• main.cpp
• momentinfo.cpp
• node.cpp
• nodeinfo.cpp
• nodelist.cpp
• Operator1.cpp
• Preconditioner1.cpp
• Struve.cpp
• tree1.cpp
• treenode.cpp
• Vector1.cpp