NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

AN INVESTIGATION OF NUMERICAL TECHNIQUES FOR THE FOURIER MATCHING METHOD ACOUSTIC SCATTERING MODEL

by

Matthew K. Henigin

September 2005

Thesis Advisor:	D. Benjamin Reeder
Co-Advisor:	John A. Colosi

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2005	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: An Investigation of Numerical Techniques for the Fourier Matching Method Acoustic Scattering Model			**5. FUNDING NUMBERS**
6. AUTHOR(S) Henigin, Matthew K.			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES)			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

The effects of extended precision computing and other numerical techniques are evaluated for the Fourier matching method (FMM) acoustic scattering model, initially developed by Assistant Professor D. Benjamin Reeder, CDR/USN (NPS), and Professor Timothy K. Stanton (MIT/WHOI). Theory on acoustic scattering, reverberation, scattering models, conformal mapping, scatterer boundary conditions, floating point arithmetic, computational error, and extended precision computing is presented as a foundation for research development. The paper presents an assessment of the effects of numerical techniques on model output with the initial expectation of obtaining a more accurate, converged solution at higher frequencies, higher modal combinations, and greater eccentricities of scatterer shape. Comparisons to results from Reeder and Stanton (2004) demonstrate effects of executed techniques. Analysis includes an evaluation of the relationship between variable precision settings and computational time, gains in the useful frequency regime of the FMM, and numerical analysis benefits. Demonstrated techniques confirm that increased precision has a positive effect on model performance. The utility of other numerical techniques is discussed, and limitations of current computer systems and other shortfalls are illustrated. A feasibility assessment for Navy use of the FMM and recommendations for further improvements to the FMM are included.

14. SUBJECT TERMS Oceanography, Acoustics, Scattering, Reverberation, Models, Fourier Matching Method, Numerical Techniques, Computer Precision			**15. NUMBER OF PAGES** 91
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited.

AN INVESTIGATION OF NUMERICAL TECHNIQUES FOR THE FOURIER MATCHING METHOD ACOUSTIC SCATTERING MODEL

Matthew K. Henigin
Lieutenant, United States Navy
B.S., United States Naval Academy, 2000

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN METEOROLOGY AND PHYSICAL OCEANOGRAPHY

from the

NAVAL POSTGRADUATE SCHOOL
September 2005

Author: Matthew K. Henigin

Approved by: D. Benjamin Reeder
Thesis Advisor

John A. Colosi
Co-Advisor

Mary L. Batteen
Chairman, Department of Oceanography

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The effects of extended precision computing and other numerical techniques are evaluated for the Fourier matching method (FMM) acoustic scattering model, initially developed by Assistant Professor D. Benjamin Reeder, CDR/USN (NPS), and Professor Timothy K. Stanton (MIT/WHOI). Theory on acoustic scattering, reverberation, scattering models, conformal mapping, scatterer boundary conditions, floating-point arithmetic, computational error, and extended precision computing is presented as a foundation for research development. The paper presents an assessment of the effects of numerical techniques on model output with the initial expectation of obtaining a more accurate, converged solution at higher frequencies, higher modal combinations, and greater eccentricities of scatterer shape. Comparisons to results from Reeder and Stanton (2004) demonstrate effects of executed techniques. Analysis includes an evaluation of the relationship between variable precision settings and computational time, gains in the useful frequency regime of the FMM, and numerical analysis benefits. Demonstrated techniques confirm that increased precision has a positive effect on model performance. The utility of other numerical techniques is discussed, and limitations of current computer systems and other shortfalls are illustrated. A feasibility assessment for Navy use of the FMM and recommendations for further improvements to the FMM are included.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. SIMULATIONS OF ACOUSTIC SCATTERING

The design of a capable sonar system requires a thorough understanding of the physical mechanisms related to the propagation of underwater sound. Two of these mechanisms, scattering and reverberation, encompass the reflection of sound from both oceanic boundaries and objects suspended within the water column. Scattering from submerged objects typically dissipates energy away from the source-receiver path and obscures returned echoes through a phenomenon called reverberation, limiting the performance of sonar systems in certain ocean environments. These two phenomena, scattering and reverberation, can be only marginally simulated even in today's most advanced sonar systems, and thus limit operational effectiveness.

A school of fish can be considered a volume of scatterers that produces reverberation when insonified by an incident pulse from an active sonar system. If scattering by the school of fish could be modeled based upon an enhanced understanding of the physical mechanisms that produced the echo or reverberation, then this reverberation could be parameterized within the active sonar equation. The amount of reverberation in the equation could be reduced based upon an understanding of the origin and character of the associated scattering events. By incorporating more finely-tuned scattering simulations, next-generation sonar systems may take advantage of improved signal-to-noise ratios, which will result in the increased probability of detection of real targets.

This research focuses on improving one such modeling approach, known as the Fourier matching method (FMM). The FMM is valid for all frequencies and scatterer shapes but is numerically limited in its utility and application to acoustic scattering by fairly regular shapes, because of several computational constraints. This research aims at addressing and conquering some of the computational limitations that are impeding implementation of the FMM into operational systems.

B. HISTORY OF SCATTERING MODELS AND THEIR APPLICATION

The problem of simulating acoustic scattering phenomena has challenged acousticians and mathematicians for decades. Lord Rayleigh studied scattering from a simple sphere during the late 1800's (Rayleigh, 1945), while Victor Anderson published a paper on the solution of sound scattering from a fluid sphere in 1950 (Anderson, 1950). However, complex shapes present the most difficult challenge to acousticians. Several numerical solutions have been developed for complex shapes, including the boundary element method (Tobacman, 1984; Francis, 1993), the T-matrix method (Waterman, 1968; Varadan *et al.*, 1982; Lakhtakia *et al.*, 1984; Hackman and Todoroff, 1985), and the mode matching method (Yamashita, 1990). However, these numerical methods all have limitations in frequency range, useful boundary conditions, scatterer surface type, dimensions of the scatterer, or mathematical and computational efficiency (Reeder and Stanton, 2004).

Reeder and Stanton (2004) introduced a general scattering formulation for calculating the far-field scattered sound pressure from irregular, axisymmetric, finite-length bodies. Their work incorporated a two-dimensional conformal mapping approach, which they adapted to scattering by finite-length bodies for three boundary conditions—soft, rigid, and fluid. Although their method was numerically efficient in its formulation, the range of parameters for which the solution converged was believed to be limited by certain factors, such as computer precision, which is an inherent limitation of many numerical models. This posed restrictions to the useful frequency range and to eccentricity of scatterer shape (Reeder and Stanton, 2004).

C. IMPORTANCE – THE REVIVAL OF OCEANOGRAPHY

The driving influence behind the choice of this research topic is the renewed interest in the ocean acoustics field within the United States Navy's Naval Meteorology and Oceanography Command. It is imperative that the U.S. Navy improve upon its existing sonar systems to meet future warfighting requirements, in order to ensure undersea dominance. Future conflicts between the U.S. and adversaries with diesel submarines operating in the challenging littoral environment could demand considerable exploitation of underwater capabilities to maintain the acoustic advantage. Advancement

of critical technologies is a present day priority for the U.S. Department of Defense. Further development of the FMM may offer potential improvements in the capabilities of active sonar systems, allowing employment of a superior anti-submarine warfare tool in the oceanic battlespace.

Submarines continue to get quieter and quieter, and sonar systems must be improved to maintain undersea dominance against these threats, particularly in the shallow water environment. Incorporation of scattering models like the FMM into sonar systems may someday enable threat detection in an intensely reverberant environment, which is very difficult with today's systems.

The operational environment requires timely processing of environmental data and acoustic returns during the prosecution of an enemy threat. Time is an important limitation in the real world, especially in combat situations. If the FMM cannot be used in a timely fashion by operational computer systems, then it is still only a research tool. With this in consideration, an analysis of computational time versus precision is incorporated into the results of this paper.

While this is still 6.1 (basic) research, the FMM will hopefully one day be incorporated into an advanced anti-submarine warfare (ASW) or undersea warfare (USW) sonar system. Other potential applications for the FMM include mine hunting, port security systems, monitoring of fish populations, or oceanographic research. The FMM is a progression of science. With continued development, this technology will enable oceanographers and operators alike to simulate the undersea environment more accurately. Tools such as the FMM could one day help operators maintain or assert undersea superiority in ways which are yet to be conceived.

D. THESIS STATEMENT

The two-dimensional Fourier matching method (FMM), introduced by DiPerna and Stanton (1994), established a conformal mapping approach, which maps the space variables to a new coordinate system and matches a constant radial coordinate to the surface of the scattering body. Reeder and Stanton (2004) extended this approach for axisymmetric, finite-length bodies of revolution. The extended approach enabled the simulation and prediction of scattering events by more complex, bounded scattering

surfaces. Reeder and Stanton (2004) obtained results using the Institute for Electrical and Electronics Engineers (IEEE) double precision format in which the floating-point numbers were stored in a 64-bit word, accurately representing numbers to about 16 decimal digits. Results of the FMM at double precision are well-known and can be duplicated with the current model.

The two primary limitations of the FMM are (1) that it requires the scattering body to be an axisymmetric body of revolution, and (2) computer precision (Reeder and Stanton, 2004). The limitations of the FMM bring several research questions to mind:

1. In what ways can the Reeder and Stanton (2004) results be improved?
2. Will increased precision computing lead to an improved converged solution?
3. Are there other potential improvements that could help the FMM produce a more accurate description of scattering phenomena?
4. What are the limiting factors for computational accuracy and efficiency and what numerical techniques can be applied to the FMM to circumvent these limiting factors?
5. How do the results of the improved model compare with previous model results and real data collected in laboratory situations?

The focus of this research is improving the performance of the FMM for application to scatterers of increased eccentricity at higher frequencies. Specifically, this research aims to improve FMM performance by increasing computer precision and by evaluating the numerical techniques that are used within the model in order to achieve greater accuracy and efficiency. Increases in precision are assessed to determine the added value. Various values of precision are utilized for investigations of computational time, because time is an operational constraint. Other possible improvements to the Reeder and Stanton model, such as numerical techniques, are investigated and incorporated into the research model where applicable.

This research will show whether or not computer precision plays an important role in the calculated representation of sound scattering events. The extent to which extended precision may improve model output is currently unknown. The exact relationship between computational precision and increased accuracy for a series solution is not well-understood. This study attempts to characterize this relationship.

The FMM has future applicability to active sonar systems in minimizing volume reverberation by subtracting the unwanted acoustic returns of recognized scatterers from the total reverberation due to active sonar transmissions. Subtracting these known echoes from the reverberation will result in an improvement in signal-to-noise-ratio. Any gains in the operational utility of the FMM demonstrated here will only be amplified as computer precision is extended on the average computer system that will be employed by tomorrow's Navy.

THIS PAGE INTENTIONALLY LEFT BLANK

II. THEORY AND FUNDAMENTALS

In the following paragraphs, theory and background information is presented for the development of the research model. Because of the breadth of topics associated with this research, theory on acoustic scattering, reverberation, scattering mathematics and models, applied computer terms and processes, computational error, numerical analysis, and assumptions and approximations is presented. Each of these subjects was considered in developing the approach for improving the FMM. General scattering formulations are described to show exactly where research efforts are focused.

A. ACOUSTIC SCATTERING – CHARACTERIZING THE PHENOMENON

Acoustic signals travel extremely long distances in the ocean environment in comparison with other forms of energy. For example, sound energy has an extremely small rate of attenuation in the ocean in contrast with light. This makes the acoustic signal a very useful tool to use in the ocean for many applications. However, objects encountered by the propagating acoustic pressure waves cause scattering. The range of propagation depends upon the attenuation rate in a continuous, fluid medium and the presence of inhomogeneities with different material and acoustic properties which cause scattering. The oceans contain many types of inhomogeneities, including schools of fish, bubbles, debris, suspended silt, and submarines. Each of these objects has a physical boundary and thus "intercepts and reradiates" some acoustic energy that interacts with that boundary (Urick, 1983). As the incident energy is reflected and reradiated outward in multiple directions, it is said to be "scattered."

It is important to note that the scattering problem is only one problem of many that are associated with the sonar equations.

> The sea is a moving medium containing inhomogeneities of various kinds, together with irregular boundaries, one of which is in motion. Multipath propagation is the rule. As a result, many of the sonar parameters fluctuate irregularly with time, while others change because of the unknown changes in the equipment and the platform on which it is mounted. Because of these fluctuations, a "solution" of the sonar equations is no more than a best-guess time average of what is to be expected in a basically stochastic problem. (Urick, 1983)

Achieving an accurate scattering simulation will reduce the amount of uncertainty in a complex problem. Scattering events affect the transmission loss and reverberation level components of the active sonar equation differently and are not very-well parameterized in current sonar system models. These poorly described phenomena are only a minute part of a largely stochastic and difficult problem. An accurate scattering simulation will reduce the uncertainty of the reverberation problem at its source—the individual scatterer.

Scattering from simple shapes, like spheres and cylinders, can be solved exactly through separation of variables. However, it is extremely difficult to obtain an exact analytical solution to scattering events from asymmetric and complex shapes. Many acousticians have tried to formulate solutions with limited success.

Some typical applications of the FMM for use in operational sonar systems might include scattering from air bubbles or fish swim bladders, which would have similar acoustical properties and boundary conditions. Fish swim at various depths in the ocean, but most air bubbles occur due to breaking wave processes and ship propellers at relatively shallow depths. Large numbers of microbubbles per unit volume have significant effects on near-surface, or shallow water, sound propagation (Medwin and Clay, 1998).

When a bubble is insonified by an incident acoustic signal, the bubble reacts with a compression and rarefaction in response to the sound wave. The frequency of the acoustic signal and the size of the bubble dictate the response of the bubble to the stimulus. A large fraction of the incident energy is reflected in all directions by the pulsating bubble and the remaining energy is converted into heat. At a certain frequency, a bubble of specified diameter resonates and produces maximum extinction of the incident sound wave by converting it into the largest possible scattered wave and heat energy (Urick, 1983).

B. REVERBERATION AND THE SONAR EQUATIONS

The cumulative sum of all scattering events from all of the scatterers within a given volume is called “reverberation” (Urick, 1983). Ol’Shevskii (1967) defines reverberation as “a process describing the time variation of the total scattered sound field

observed at the point of reception following transmission of a sound signal." Reverberation is generally considered the primary limitation on performance of active sonar systems. Reverberation that takes place within the water column, aside from its boundaries, is called "volume reverberation" (Urick, 1983). This occurs when scattered sound is returned back to its source in a monostatic arrangement. Volume reverberation occurs in association with a field of scatterers.

In the ocean, the scatterers that cause volume reverberation can almost never be directly determined. Usually a speculation has to be made as to what types of fish, invertebrates, bivalves, or other scatterers are present, introducing significant uncertainty into the reverberation level term of the active sonar equation. If the echoes from certain known scatterers can be accurately modeled, then real world scatterers can also be modeled. Information about the underwater environment can be deduced based upon known reverberation characteristics.

Early investigations of reverberation as a stochastic process were conducted by Yu. M. Sukharevskii (Ol'Shevskii, 1967). In 1947, he published a paper which indicated that the reverberation process could be described to an acceptable approximation by the sum of a large number of elementary scattered acoustic signals (Ol'Shevskii, 1967). If the ocean provided homogeneous fields of scatterers, the reverberation problem would be easy to solve. However, reverberation is almost always caused by a heterogeneous field of scatterers. This poses another problem.

It is important to understand the scattering from individual scatterers first before the reverberation problem can be addressed. Then the reverberation research can be directly applied to real world problems. Ol'Shevskii (1967) also said, "Once, however, a definite hypothesis is adopted regarding the distribution of scatterers in the ocean, as well as their possible sizes and acoustical properties, it is possible to carry out a fairly complete analysis of the statistical characteristics of reverberation without analyzing in detail the scattering by all possible types of inhomogeneity." If individual scattering events from scatterers such as fish can be mastered, then there is no need to estimate changing fish populations for acoustic purposes, and the reverberation problem can be solved with more confidence.

This research focuses on individual scattering events. Once the physics of the individual events are modeled correctly and with a certain degree of real-world utility, then the formulation can be applied to the largely stochastic problem of volume reverberation.

C. SCATTERING MATHEMATICS AND MODELS

1. Scattering Formulation – General Solution

The modeling of scattering phenomena begins with the three-dimensional scalar wave equation:

$$\nabla^2 p = (1/c^2)(\partial^2 p / \partial t^2), \tag{1}$$

where p is the acoustic pressure, ∇^2 is the Laplacian operator, c is the speed of sound, and t is time. Exact analytical solutions to the wave equation necessitate that the scatterer's surface coincide with the locus of all points for which the radial coordinate is a constant (Reeder and Stanton, 2004). This requirement can be met for simple geometries, such as a sphere, an infinitely long cylinder, or a prolate spheroid. For these geometries, scattering phenomena can be formulated mathematically with an exact answer.

In order to better understand acoustic scattering from a sphere, the general solution from Victor Anderson's paper "Sound Scattering from a Fluid Sphere" (1950) will be analyzed. Anderson's paper examines scattering from a fluid spherical scatterer, for which viscosity and heat conduction effects are assumed to be negligible. The sphere has dimensions that are similar to the wavelength of the acoustic energy with which it is insonified. The sphere has radius a and contains fluid 1, which has density ρ' and sound velocity c'. The sphere is centered at the origin of a polar coordinate system with radius denoted by r, azimuthal angle φ ranging from 0 to 2π, and polar angle ϑ ranging from 0 to π. The geometry of this sphere is shown in Figure 1. Surrounding this sphere is fluid 2, which has different acoustical properties than the sphere that are designated by ρ and c (Anderson, 1950).

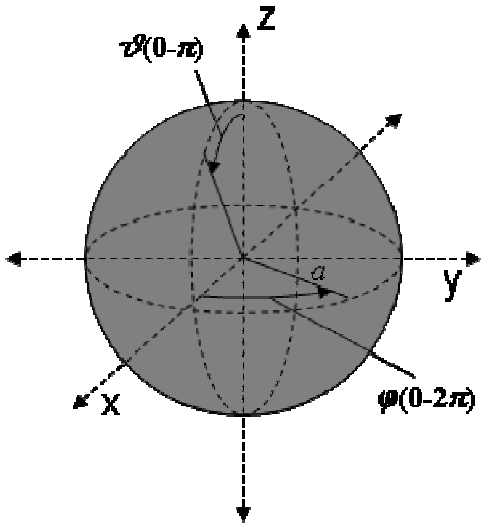


Figure 1. Spherical geometry for the Anderson (1950) solution to scattering from a fluid sphere. Azimuthal angle φ is measured from the positive *x*-axis in the *xy*-plane and ranges from 0-2π. The polar angle ϑ is measured from the positive *z*-axis and ranges from 0-π. The sphere has radius, *a*.

Far-field scattering is assumed, giving the acoustic signal ample time and space to spread significantly from its source. Thus, the incident acoustic signal is insonifying the scatterer in straight and parallel wave fronts. This is known as the plane-wave approximation (Medwin and Clay, 1998). The sphere is insonified by a plane acoustic wave with pressure amplitude P and angular frequency ω traveling parallel to the polar axis in the *z* direction. With this choice of incident wave the dependence on angle φ is removed, so that only *r* and ϑ must be considered (i.e., backscattering from the fluid sphere is identical for all φ). Once the sphere is insonified, the internal pressure becomes p' and a scattered spherical wave with acoustic pressure *p* emanates outward from the sphere. The purpose of this solution, and of the FMM, is to calculate the pressure amplitude *p* of this scattered wave at large ranges from the sphere.

Along the outer edge of the sphere, where $r = a$, the pressure and the normal component of particle velocity, *u*, must be continuous. Under these conditions, the following equations should be considered:

$$p(a) + p_0(a) = p'(a), \text{ and} \tag{2}$$

$$u_r(a) + u_{0,r}(a) = u_r'(a), \tag{3}$$

where p_0 is the incident acoustic pressure and u_r is the radial component of the particle velocity. The radial component of particle velocity has an incident component, $u_{0,r}$, and an internal component, u_r', while the scattered component is denoted by u_r (Anderson, 1950).

The solution for acoustic pressure, p, must satisfy the three-dimensional wave equation given by Equation (1). The solution of the scalar wave equation is assumed to be time harmonic, with $e^{-i\omega t}$, where ω is the angular frequency. With this assumption, the scalar wave equation can be transformed into the three-dimensional Helmholtz differential equation:

$$\nabla^2 p(x,y,z) + k^2 p(x,y,z) = 0, \tag{4}$$

In this equation, k is wave number, and $k = \omega / c = 2\pi / \lambda$, where λ is the wavelength of the acoustic energy. For the spherical coordinate system used in this formulation, the Helmholtz equation is:

$$\nabla^2 p(r,\vartheta,\varphi) + k^2 p(r,\vartheta,\varphi) = 0. \tag{5}$$

In order to solve this equation, it must be expanded into:

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 \frac{\partial p}{\partial r}\right) + \frac{1}{r^2 \sin\vartheta}\frac{\partial}{\partial \vartheta}\left(\sin\vartheta \frac{\partial p}{\partial \vartheta}\right) + \frac{1}{r^2 \sin\vartheta}\frac{\partial^2 p}{\partial \varphi^2} + k^2 p = 0, \tag{6}$$

and then separation of variables must be carried out, with:

$$p(r,\vartheta,\varphi) = R(r)\Theta(\vartheta)\Phi(\varphi), \tag{7}$$

where the φ terms are known to be either $\cos m\varphi$ or $\sin m\varphi$ due to sinusoidal dependence. The term $\frac{\partial^2 p}{\partial \varphi^2}$ can be replaced with $-m^2 p$, where $-m^2$ is a separation constant (Haberman, 1998). In addition, the time harmonic portion of the scattering solution of unit amplitude is $p = e^{-i\omega t}$, which will be seen in the solution further on.

With some simplification, the separated expression becomes:

$$\frac{1}{R}\frac{d}{dr}\left(r^2\frac{dR}{dr}\right)+k^2r^2=\frac{-1}{\Theta\sin\vartheta}\frac{d}{d\vartheta}\left(\sin\vartheta\frac{d\Theta}{d\vartheta}\right)-\frac{m^2}{\sin^2\vartheta}=\mu, \tag{8}$$

where μ is another separation constant (Haberman, 1998). The left side of this equation can be arranged into what is known as Sturm-Liouville form:

$$\frac{d}{dr}\left(r^2\frac{dR}{dr}\right)+(k^2r^2-\mu)R=0 \text{ (Haberman, 1998).} \tag{9}$$

This ordinary differential equation is the eigenvalue problem in *r* which can be solved with spherical Bessel functions.

The remainder of Equation (8) above is:

$$\frac{-1}{\Theta\sin\vartheta}\frac{d}{d\vartheta}\left(\sin\vartheta\frac{d\Theta}{d\vartheta}\right)-\frac{m^2}{\sin^2\vartheta}=\mu, \tag{10}$$

which can be simplified to:

$$\frac{d}{d\vartheta}\left(\sin\vartheta\frac{d\Theta}{d\vartheta}\right)+\left(\mu\sin\vartheta-\frac{m^2}{\sin\vartheta}\right)\Theta=0 \tag{11}$$

where $0<\vartheta<\pi$. This is now the eigenvalue problem in ϑ. Assume boundary conditions $\Theta(0)<\infty$ $(finite)$, $\Theta(\pi)<\infty$ $(finite)$, $R(a)=0$, and $|R(0)|<\infty$ $(finite)$. Let $x=\cos\vartheta$ to simplify the equation in ϑ. The derivatives can be handled with the chain rule:

$$\frac{d}{d\vartheta}=\frac{dx}{d\vartheta}\frac{d}{dx}=-\sin\vartheta\frac{d}{dx} \text{ (Haberman, 1998).} \tag{12}$$

After dividing by $\sin\vartheta$ and recognizing that $\sin^2\vartheta=1-\cos^2\vartheta=1-x^2$, the above equation in ϑ becomes:

$$\frac{d}{dx}\left[(1-x^2)\frac{d\Theta}{dx}\right]+\left(\mu-\frac{m^2}{1-x^2}\right)\Theta=0 \text{ (Haberman, 1998).} \tag{13}$$

In order to get a bounded solution at $x=\pm1$, the simplification $\mu=n(n+1)$ is used. For

$m = 0$, where there is no dependence on φ, the differential equation is then:

$$\frac{d}{dx}\left[(1-x^2)\frac{d\Theta}{dx}\right] + n(n+1)\Theta = 0 \quad \text{(Haberman, 1998).} \tag{14}$$

The bounded solutions to this equation are called Legendre polynomials, which satisfy Rodrigues' formula:

$$P_n(x) = \frac{1}{2^n n!}\frac{d^2}{dx^2}(x^2-1)^n \quad \text{(Haberman, 1998).} \tag{15}$$

The first few Legendre polynomials take the following forms:

$$\begin{aligned} &n = 0 : P_0(x) = 1 \\ &n = 1 : P_1(x) = x = \cos\vartheta \\ &n = 2 : P_2(x) = \frac{1}{2}(3x^2-1) = \frac{1}{4}(3\cos 2\vartheta + 1) \end{aligned} \qquad \text{(Haberman, 1998).} \tag{16}$$

Thus, the Legendre polynomials of order *n*, represented by $P_n(\cos\vartheta)$, are the solution in the ϑ direction for $m = 0$. The associated Legendre functions which are incorporated into the FMM formulation apply when $m > 0$ and are similar to the Legendre polynomials shown here for $m = 0$.

Revisiting the Sturm-Liouville Equation (9) from above, with $\mu = n(n+1)$, the equation becomes:

$$\frac{d}{dr}(r^2\frac{dR}{dr}) + \left(k^2r^2 - n(n+1)\right)R = 0 \tag{17}$$

for $n \geq m$ with fixed *m*. The solution at $r = 0$ must be bounded and $R(a) = 0$, where *a* is the radial coordinate at the scatterer's surface. If *kr* is considered a separate variable and $\zeta = kr$, then there are two solution forms to Equation (17). The first solution is the spherical Bessel function of the first kind:

$$j_n(\zeta) = \sqrt{\frac{\pi}{2\zeta}}J_{n+1/2}(\zeta) \quad \text{(Abromowitz and Stegun, 1965).} \tag{18}$$

The second solution is the spherical Bessel function of the second kind, otherwise known as the spherical Neumann function:

$$n_n(\zeta) = \sqrt{\frac{\pi}{2\zeta}} N_{n+1/2}(\zeta) = (-1)^{n+1} \sqrt{\frac{\pi}{2\zeta}} J_{-n-1/2}(\zeta) \tag{19}$$

(Abromowitz and Stegun, 1965).

The solutions $j_n(\zeta)$ and $n_n(\zeta)$ are the solutions to the partial differential equation for *r*. The addition of the spherical Bessel function and the spherical Neumann function yields the spherical Hankel function of the first kind which is of the form:

$$h_n^{(1)}(\zeta) = j_n(\zeta) + i n_n(\zeta), \tag{20}$$

which gives spherical harmonics of outward radiating acoustic energy. This is the general solution for the eigenvalue problem in the radial direction.

Putting all of the derived components together yields the general solution for the fluid sphere:

$$p = \sum_{n=0}^{\infty} A_n P_n(\cos\vartheta) \begin{bmatrix} j_n(kr) \\ n_n(kr) \end{bmatrix} e^{-i\omega t} \tag{21}$$ (Anderson, 1950),

where p is the acoustic pressure, A_n is the summation coefficient, $P_n(\cos\vartheta)$ is the Legendre polynomial, $j_n(kr)$ is the spherical Bessel function, $n_n(kr)$ is the spherical Neumann function, and $e^{-i\omega t}$ is the sinusoidal time-harmonic portion of the solution. This equation gives spherical harmonics of acoustic pressure.

The solution for the internal wave, where $r < a$, is:

$$p' = \sum_{n=0}^{\infty} B_n P_n(\cos\vartheta) j_n(k'r) e^{-i\omega t}. \tag{22}$$

The scattered wave in the region where $r > a$ is then:

$$p = \sum_{n=0}^{\infty} A_n P_n(\cos\vartheta) \left[j_n(kr) + i n_n(kr) \right] e^{-i\omega t}, \tag{23}$$

which is the portion of the solution that is most applicable to this research. However, all parts, including the incident acoustic wave:

$$p_0 = \mathrm{P}_0 \sum_{n=0}^{\infty} (-i)^n (2n+1) P_n(\cos\vartheta) j_n(kr) e^{-i\omega t}, \tag{24}$$

are required to solve for the coefficients A_n and B_n. Here, P_0 is the incident acoustic pressure amplitude from the source. The equation for the radial component of particle velocity, $u_r = \left(\frac{-i}{\rho c}\right)\frac{\partial(p)}{\partial(kr)}$ can be used to obtain the equations for the particle velocities of each of the three individual waves—incident, internal and scattered. Then, equations for the particle velocities and the acoustic pressure equations can be solved simultaneously to yield the following expression for A_n:

$$A_n = -\mathrm{P}_0 (-i)^n (2n+1)/(1+iC_n). \tag{25}$$

Substituting this into Equation (21) yields:

$$p = -\mathrm{P}_0 \sum_{n=0}^{\infty} \left[(-i)^n (2n+1)/(1+iC_n) \right] \times P_n(\cos\vartheta) \left[j_n(kr) + i n_n(kr) \right] e^{-i\omega t}, \tag{26}$$

which is the general solution for the total acoustic pressure external to the sphere (Anderson, 1950).

The Anderson (1950) solution is directly related to the math incorporated into the FMM, but the FMM contains modifications for mathematical improvements like geometry parameterizations and changes in boundary conditions under different physical circumstances. In the case of backscattering by a fluid sphere, the FMM formulation and predictions are identical to Anderson (1950) (Reeder and Stanton, 2004).

2. Fourier Matching Method

The Fourier Matching Method (FMM) was introduced by DiPerna and Stanton (1994). This method introduced a conformal mapping tactic for the prediction of far-field acoustic scattering for the case of an infinitely long cylinder with non-circular cross section. The method incorporates the conformal mapping of variables to a different coordinate system, in which the constant radial coordinate exactly matches the scatterer's

surface. The conformal mapping method generates a transformed Helmholtz equation, which is solvable even for irregularly shaped objects (Reeder and Stanton, 2004). The new Helmholtz equation is solved with a given set of boundary conditions, and the scattering formulation yields values for backscattering amplitude (f_{bs}), target strength (*TS*), and reduced target strength (*RTS*) for a prescribed array of frequencies. These parameters are discussed along with the FMM formulation presented here.

a. General Solution Formulation of the FMM

The general solution incorporated into the FMM is similar in form to the Anderson (1950) general solution developed above. The formulation for the FMM extended to axisymmetric, finite-length bodies by Reeder and Stanton (2004) is very similar to that used in the two-dimensional solution developed by DiPerna and Stanton (1994). The FMM is formulated with spherical wave functions, including spherical Bessel functions, spherical Neumann functions, spherical Hankel functions, and associated Legendre functions. This formulation, like the Anderson (1950) solution, also starts with the wave equation, which is expressed as the Helmholtz equation in spherical coordinates given in Equation (5). The general solution to this equation is represented in the FMM as:

$$(27)\ p^{ext}(r,\vartheta,\varphi) = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} A_{nm} j_n(kr) P_n^m(\cos(\vartheta)) e^{im\varphi} + \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} B_{nm} h_n^{(1)}(kr) P_n^m(\cos(\vartheta)) e^{im\varphi}$$

(Reeder and Stanton, 2004).

In this equation, $j_n(kr)$ is the spherical Bessel function of order *n*, $P_n^m(\cos(\vartheta))$ is the associated Legendre function of degree *n* and order *m*, and $h_n^{(1)}(kr)$ is the spherical Hankel function of the first kind with order *n*. The total pressure external to the scattering body is denoted by $p^{ext}(r,\vartheta,\varphi)$. The first term in the general solution equation above characterizes the incident acoustic pressure, and the second term corresponds to the scattered acoustic pressure. The scattered field coefficients are denoted by B_{nm}, and they can be calculated by using the known incident field coefficients, A_{nm}:

$$(28)\qquad A_{nm} = i^n \in_m (2n+1)\frac{\Gamma(n-m+1)}{\Gamma(n+m+1)} P_n^m(\cos(\vartheta_0))$$ (Reeder and Stanton, 2004).

In this equation, $\in_m$ is the Neumann factor and Γ is the gamma function. Another difference between the FMM and the Anderson (1950) solution is that the number of modes associated with φ, denoted by *m*, is allowed to be greater than zero.

The formulation incorporates a general spherical coordinate system. However, this coordinate system must be modified in order to accommodate different and more complex scatterer shapes. Conformal mapping is used to transform the Helmholtz equation from one set of coordinates to another in order to accommodate a variety of geometries. This modification is outlined below.

b. Coordinate System

The FMM uses an orthogonal coordinate system that can be created for a three-dimensional body of revolution from a two dimensional conformal mapping. The two dimensional approach was used by DiPerna and Stanton (1994). The orthogonal coordinate system starts from the typical spherical coordinate system shown in Figure 2.

The azimuthal angular coordinate is φ in this case, which ranges from 0 to 2π, and is measured from the positive *x*-axis in the *xy*-plane. The polar angular coordinate is ϑ, which ranges from 0 to π, is measured from the positive *z*-axis.

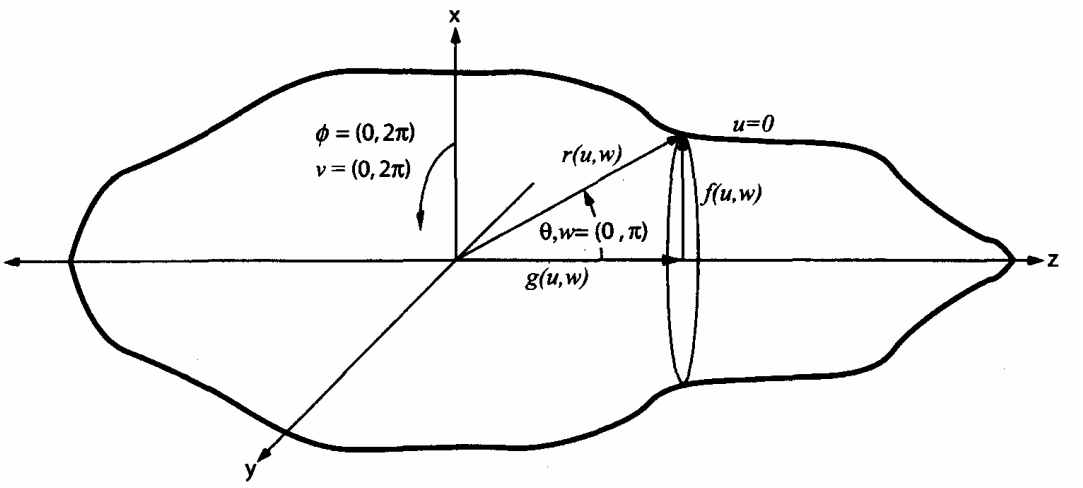


Figure 2. Geometry for an irregular, axisymmetric, finite-length body of revolution. In this case, the body is symmetric about the z-axis. This figure shows the geometry for both spherical coordinates and an orthogonal, conformally mapped coordinate system. (From Reeder and Stanton, 2004)

The radial coordinate, *r*, is a scalar measured from the origin, which ranges from 0 to ∞. The body of revolution is formed by rotating a contour about the *z*-axis in the φ direction.

The new, orthogonal coordinate system has azimuthal angular coordinate *v*, which corresponds to φ and ranges from 0 to 2π. The polar angular coordinate is denoted by *w*, which corresponds to ϑ and spans a range from 0 to π. In the new coordinate system, the radial coordinate that defined the scatterer surface in the old coordinate system is changed. It is now defined by the locus of all points for which the new radial coordinate is a constant, or where $u = 0$ (Reeder and Stanton, 2004). The original body of revolution must also be parameterized in the new coordinate system. Using functions $f(u,w)$ and $g(u,w)$ together with trigonometry, the dimensions of the scatterer in the *x*, *y* and *z* directions become:

$$\begin{aligned} x(u,w,v) &= f(u,w)\cos(v) \\ y(u,w,v) &= f(u,w)\sin(v) \\ z(u,w,v) &= g(u,w) \end{aligned} \qquad (29)$$

(Reeder and Stanton, 2004).

Further, the radial position vector becomes:

$$\mathbf{r}(u,w,v) = x(u,w,v)\hat{i} + y(u,w,v)\hat{j} + z(u,w,v)\hat{k} \qquad (30)$$

(Reeder and Stanton, 2004),

where $\hat{i}$, $\hat{j}$ and $\hat{k}$ are unit vectors in the respective *x*, *y* and *z* directions. The scatterer dimensions from Equation (29) can be substituted into this radial position vector to yield:

$$\mathbf{r} = f(u,w)\cos(v)\hat{i} + f(u,w)\sin(v)\hat{j} + g(u,w)\hat{k} \qquad (31)$$

(Reeder and Stanton, 2004).

The orthogonal coordinate system is important because it aids in the calculation of normal particle velocities on the boundary, which are necessary in satisfying the boundary conditions (Reeder and Stanton, 2004). Orthogonality requires that $\mathbf{r}_u \cdot \mathbf{r}_v = 0$, $\mathbf{r}_w \cdot \mathbf{r}_v = 0$, and $\mathbf{r}_u \cdot \mathbf{r}_w = 0$. These conditions allow a conformal transformation, where a shape that was represented in the (*x*,*y*,*z*) coordinate system can be depicted as a shape in the (*u*,*w*,*v*) coordinate system.

c. Conformal Mapping and Parameterization

In order to use the orthogonal coordinate system, the (x,y,z) representation of the scatterer's surface must be transformed with a mapping function into the new geometry in the (u,w,v) coordinate system. Conformally mapping the old geometry into the new coordinate system guarantees that the new geometry will be mutually orthogonal (Reeder and Stanton, 2004). Reeder and Stanton (2004) wrote "the conformal mapping generates a new set of angular functions which fit the scatterer surface more naturally; that is, points along the surface that change rapidly in (x,y,z) are plotted at a higher spatial rate yet are equally spaced in (u,w,v)."

The mapping function provides the means by which the scatterer shape is mapped to the new orthogonal, axisymmetric coordinate system. A three-dimensional mapping scheme does not yet exist within the field of mathematics, so the mapping must also be for an axisymmetric body of revolution (Reeder and Stanton, 2004). To this point, the FMM has been developed with the two-dimensional mapping approach that was introduced by DiPerna and Stanton (1994). See Reeder and Stanton (2004) for further information and development of the DiPerna and Stanton (1994) mapping function.

The mapping algorithm was not utilized in this research. For a smooth sphere or spheroid, the conformal mapping reduces to:

$$f = a * \sin(w) \qquad g = b * \cos(w)\text{,} \tag{32}$$

where a is the half-width of the body along the semi-minor axis, and b is the half-length of the body along the semi-major axis of the spheroid. Aspect ratio (AR) is defined as:

$$AR = \frac{b}{a}. \tag{33}$$

The shapes that were used in conjunction with this research are shown in Figure 3.

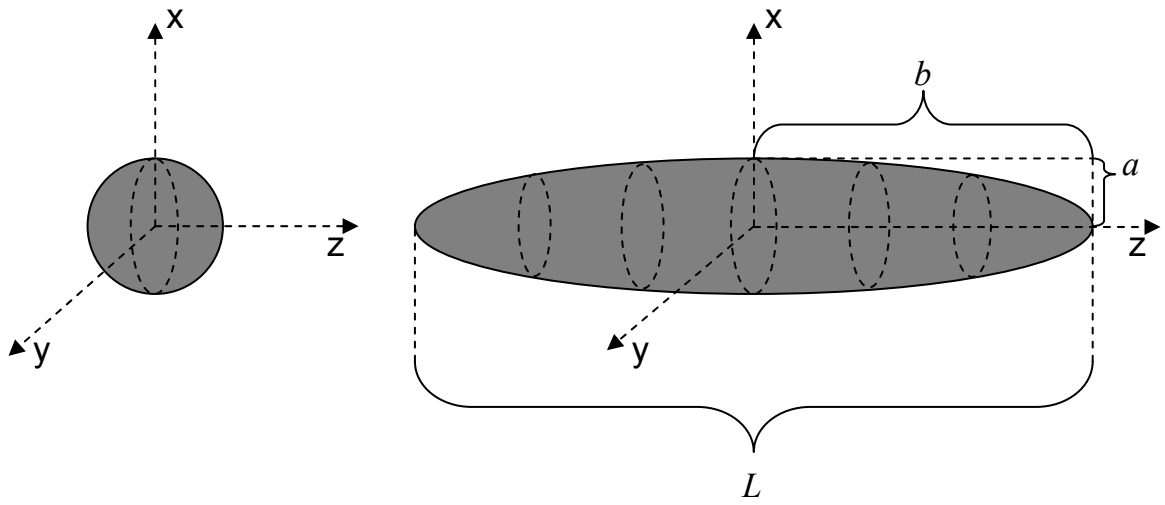


Figure 3. A sphere with aspect ratio of 1:1 and a prolate spheroid with aspect ratio 5:1. These are the objects that are used in FMM simulations for this research. The objects are three-dimensional and are axisymmetric about the *z*-axis.

With the change to the new coordinate system, the following relationships are established:

$$\begin{aligned} &\varphi = v, \\ &r(u,w) = \sqrt{f^2(u,w) + g^2(u,w)}, \\ &\cos(\vartheta(u,w)) = \frac{g(u,w)}{r(u,w)}. \end{aligned} \tag{34}$$

In addition, the Helmholtz equation becomes:

$$\nabla^2 p(u,w,v) + k^2 F(u,w) p(u,w,v) = 0, \tag{35}$$

where (*u*,*v*,*w*) make up the new coordinate system, and $F(u,w)$ is a special function that depends upon the type of transformation (Reeder and Stanton, 2004). With this change, wave number becomes a function of position, but the Helmholtz equation itself is otherwise very similar to its counterpart in Cartesian coordinates (Reeder and Stanton, 2004).

The solution to the new Helmholtz equation is:

$$p^{ext}(u,w,v) = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} A_{nm} j_n(kr(u,w)) P_n^m\left(\frac{g(u,w)}{r(u,w)}\right) e^{imv} + \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} B_{nm} h_n^{(1)}(kr(u,w)) P_n^m\left(\frac{g(u,w)}{r(u,w)}\right) e^{imv} \quad (36)$$

(Reeder and Stanton, 2004).

This equation yields the total far-field acoustic pressure as a sum of the incident and scattered acoustic pressure in the new orthogonal coordinate system. This solution is valid for axisymmetric bodies for all frequencies and for all angles. It also applies to both impenetrable (soft and rigid) and penetrable (fluid) boundary conditions.

d. Target Strength (TS) and Reduced Target Strength (RTS)

At a great distance from the scatterer, the scattered acoustic pressure is given by the limit:

$$p^{scat} \xrightarrow[u\to\infty]{} p^{inc} \frac{e^{ikr}}{r} f_s \quad (37)$$

(Reeder and Stanton, 2004),

where p^{scat} is the scattered pressure given by the second term of Equation (36), and p^{inc} is the incident pressure given by the first term. The scattering amplitude, f_s, is an expression of scattering efficiency and is a function of the scatterer's size, shape, orientation, physical composition, in addition to the wavelength of the incident wave (Reeder and Stanton, 2004). The scattering amplitude is given by:

$$f_s = \sum_{n=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} B_{nm} i^{-n-1} P_n^m\left(\frac{g(u,w)}{r(u,w)}\right) e^{imv} \quad (38)$$

(Reeder and Stanton, 2004).

Acoustic energy that is calculated in the far-field region in the backscatter direction is usually conveyed as target strength (*TS*) in units of decibels (dB) relative to 1 m (Urick, 1983). The expression for target strength is:

$$TS = 10\log\sigma_{bs}, \quad (39)$$

where σ_{bs} is the differential backscattering cross section. This is similar to the more common backscattering cross section, σ, but is multiplied by a factor of 4π so that $\sigma = 4\pi\sigma_{bs}$ (Reeder and Stanton, 2004). The differential backscattering cross section can

also be represented in the form $\sigma_{bs} = |f_{bs}|^2$, where f_{bs} is the scattering amplitude calculated in the backscattering direction (Reeder and Stanton, 2004).

When comparing scattering events from similar objects of different sizes, it is often useful to normalize target strength by the square of an associated dimension. The resulting normalized target strength is then called "reduced" target strength (*RTS*). Using the length (L) of an elongated scatterer as the normalization constant, *RTS* can be expressed as:

$$\begin{aligned} RTS &= 10\log\left(\frac{\sigma_{bs}}{L^2}\right) \\ &= 10\log|f_{bs}|^2 - 10\log(L^2) \\ &= 10\log\left|\frac{f_{bs}}{L}\right|^2 \end{aligned} \qquad (40)$$

(Reeder and Stanton, 2004).

For a sphere, the target strength should be normalized by πa^2 instead of L^2 (Reeder and Stanton, 2004). Plots of *RTS* vs. dimensionless frequency (*ka*) are useful in resolving the scattering behavior of a scatterer, in addition to determining model characteristics and performance. *RTS* is expressed in units of decibels (dB).

e. Benefits and Limitations of the Method

Reeder and Stanton (2004) showed that the FMM is accurate over a wide range of scatterer cross sections, signal frequencies, and boundary conditions. Under many circumstances, the FMM is superior to other acoustic models. Figure 4 shows that the FMM matches the exact solutions for soft, rigid, and fluid spheres formulated by Anderson (1950).

While the FMM has several advantages like enabling simulations of scattering from complex shapes, it also has several limitations. In order for the FMM to be applied to a three-dimensional scatterer, the scatterer itself must be axisymmetric, or symmetric about one of its axes. In other words, the outer boundary of the scatterer must be depicted by a function rotated around the length-wise axis. This is because the FMM was initially formulated by DiPerna and Stanton (1994) for the case of an infinitely-long cylinder, which is a two-dimensional scattering solution. Real scattering events typically

occur with three-dimensional scatterers. The FMM was developed from its original form by Reeder and Stanton (2004) to apply to three-dimensional bodies, with the limitation that the body must be described by a function rotated about the length-wise *z*-axis.

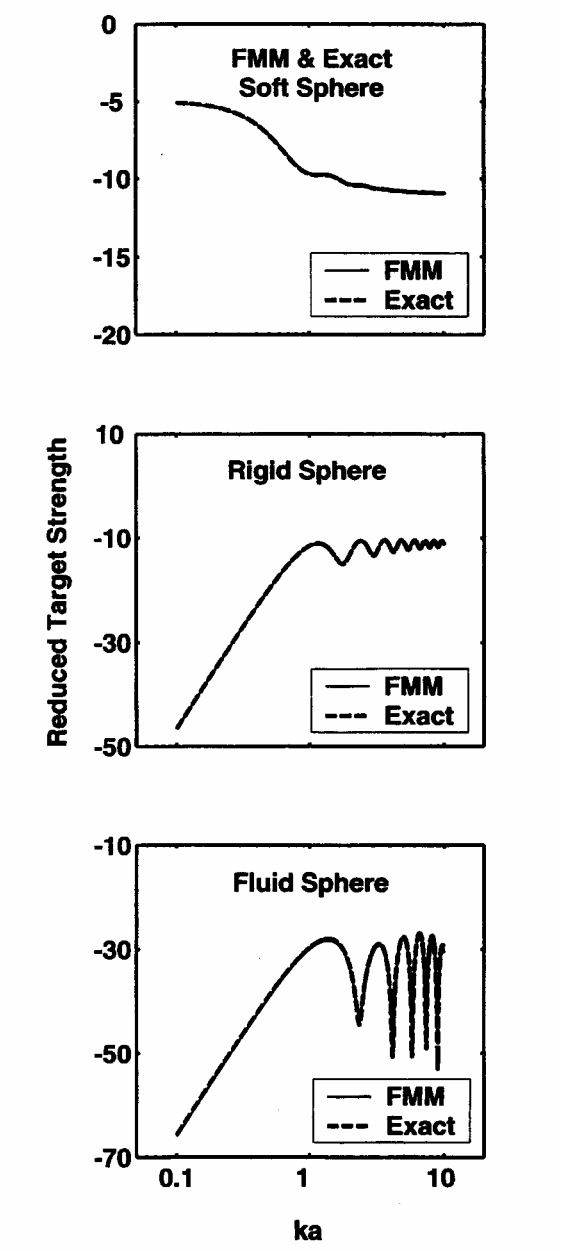


Figure 4. FMM comparisons to exact solutions. This shows reduced target strength in dB plotted as a function of *ka* for the cases of soft, rigid and fluid spheres. The exact solutions shown here are from Anderson (1950). (From Reeder and Stanton, 2004)

In addition, it is single-valued in r—there can be only one value of r for each value w (Reeder and Stanton, 2004). The FMM is also computationally expensive, in that it takes a long time to run on current day computer systems.

In addition to the above listed limitations, the FMM (or any scattering model, for that matter) is only as good as the mathematics that it incorporates. The FMM uses spherical wave functions; however, not all of the scatterers to which this model is applied are actually spherical. For example, most fishes' swim bladders are elongated, irregularly-shaped, prolate spheroids. Although the conformal mapping that is incorporated into the Reeder and Stanton (2004) FMM attempts to provide the details of a scatterer's boundary, the basis functions within the formulation must still be somewhat similar to the object that they are being used to describe. Figure 5 shows the anatomy of an alewife (*Alosa pseudoharengus*) with the outline of the fish's swim bladder from an x-ray image (Reeder, *et al*., 2004). This fish, and specifically its swim bladder and head, is an example of an individual scatterer to which the FMM would directly apply.

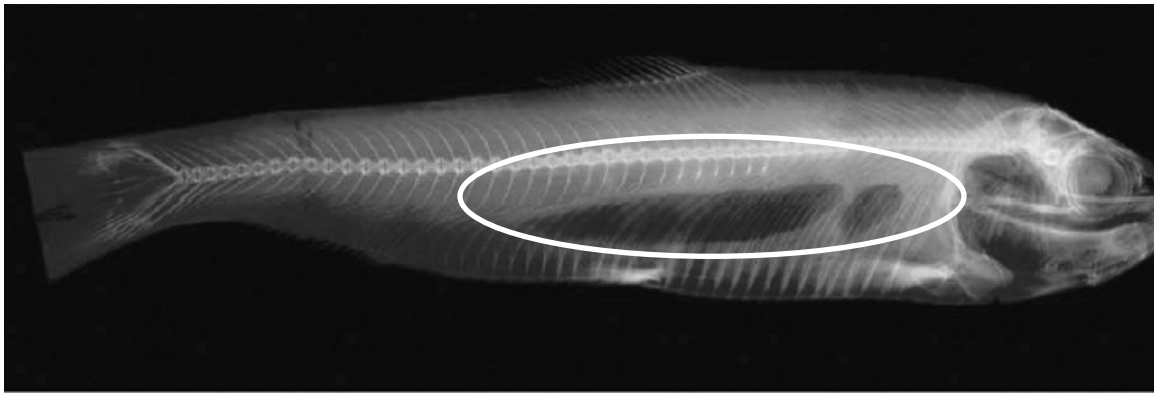


Figure 5. X-ray image of an alewife (*Alosa pseudoharengus*) with swim bladder encircled in white. (After Reeder, *et al*., 2004)

Notice that the swim bladder is not a sphere or a spheroid but a complex shape with an irregular boundary. Although the swim bladder is not spherical or spheroidal, the shape can be better represented by a prolate spheroid than with a sphere. The FMM would be more applicable to scattering from fish and submarines if it was formulated with spheroidal wave functions rather than spherical wave functions.

3. Boundary Conditions

Inhomogeneities in the ocean may be characterized as hard-, soft-, or fluid-type scatterers, depending upon the material properties of which they are composed. A scatterer's characterization depends upon the boundary between it and the ocean, as well as its interior structure. Sound interacts differently with each type of boundary and scattering body. Thus, there is a need for boundary conditions within a mathematical scattering solution, and boundary conditions often do not accurately depict the boundary of a scatterer. Different behavior is forced by each boundary condition, as shown by Figure 4 in the last section. It is important to use the correct boundary conditions, or the entire scattering formulation could yield incorrect results.

Boundary conditions are used to produce the particular solution for a given set of initial conditions, such as scatterer properties. While Reeder and Stanton (2004) evaluated the model with soft, rigid, and fluid boundary conditions for their work, soft boundary conditions were used in this research for purposes of consistency in evaluating the effects of precision and numerical techniques.

The soft boundary conditions are otherwise known as pressure release or Derichlet boundary conditions, where the total pressure vanishes along the surface of the scattering body (Reeder and Stanton, 2004). In other words:

$$p^{ext}(u_0, w, v) = 0 \tag{41}$$

from Equation (36). Thus, the series solution in the FMM general solution above is set equal to zero at the boundary. Reeder and Stanton (2004) outline the development of the boundary conditions for the specific case of a soft boundary, which is used here. The series coefficients, B_{nm}, for the scattered field are:

$$B_{nm} = -(Q_n^m)^{-1} R_n^m A_{nm}, \tag{42}$$

where the inverse notation indicates a matrix inversion and R and Q are integral expressions for summation over w (Reeder and Stanton, 2004). This boundary condition represents the physical case of a bubble or fish's swim bladder, in which the difference in

fluid properties between the exterior and interior of the scatterer is significant and the surface of the scatterer is free to deform when insonified by an acoustic pressure wave.

4. Series Solutions and Convergence

For the FMM, the *m*-modes are the series components that are associated with the azimuthal angular coordinate φ. The *n*-modes are the series components associated with the polar angular coordinate ϑ. In a series solution, the modes are summed until a converged answer is achieved. The individual modes can be analyzed for their separate contributions to the total answer. When modes are added together, the solution should become increasingly converged with the addition of each mode.

For the general formulation and solution to the spherical geometry in the Anderson (1950) paper, the series solution reaches convergence. The sum of the series components is an exact mathematical solution. For simple geometries, the series solution requires only a few modes in the summation to yield a converged solution. However, when irregular scatterer shapes are introduced into the problem, additional modes are required to be calculated in order to approach a converged solution. With more complex shapes, more modes are usually required to represent the scattering phenomena. Scattering from objects with higher aspect ratios or rough outer surfaces is extremely difficult to predict because of the higher number of modal combinations necessary in the computation. Based upon these circumstances, there is an inverse relationship between computational accuracy and computational efficiency.

Convergence is achieved only to a certain point in the solution, however. When other factors such as error are introduced into the mathematical formulation by the higher modal combinations, model calculations become much more cumbersome. At higher modal combinations, error becomes a significant part of the solution. Error starts to dominate some modes for which the contribution to the total solution is only an extremely small number. In this circumstance, the error may outweigh the actual modal contribution, and the model will be inaccurate. At this point, the solution will incorporate numerous erroneous values that are not actually part of the intended solution, and the model can no longer be used for a useful scattering simulation.

5. Truncation

Realistically speaking, a computed series can never give an exact solution because there are an infinite and unattainable number of terms in the solution. At a certain point within an infinite series, the series term being computed requires more decimal places than the computer can actually handle due to precision limits. The smaller terms at higher modes cannot be resolved by the computer.

One method to deal with this problem is to truncate the solution and include only the modes that are known to be correct in the calculated solution. There is not a common stopping point in the series solution for all scatterer shapes. The point at which the solution is stopped, or truncated, is different for various initial conditions and scattering surfaces.

Truncation may also be required when the time needed to calculate the next mode would otherwise extend model runtime beyond the useful limit. In many situations, truncation at a certain point in the series solution could be necessary, in order to avoid computer delays for additional calculations that yield little or no benefit. There is little value added in calculating a mode that changes the solution only by a fraction of a percentage point.

Reeder and Stanton (2004) conducted a study to discern the operational range of the FMM. They tallied the results in the *performance envelope* plot shown in Figure 6. They defined a *converged* solution as "one in which the computation of additional modes does not significantly change the result for a given value of *ka*" (Reeder and Stanton, 2004). The *truncated* approximations are those for which the FMM yields results that leave some question of reliability and accuracy. The *numerically stable* approximations are those that reach beyond the truncated approximations and that still yield results that are not rendered completely useless by computational errors.

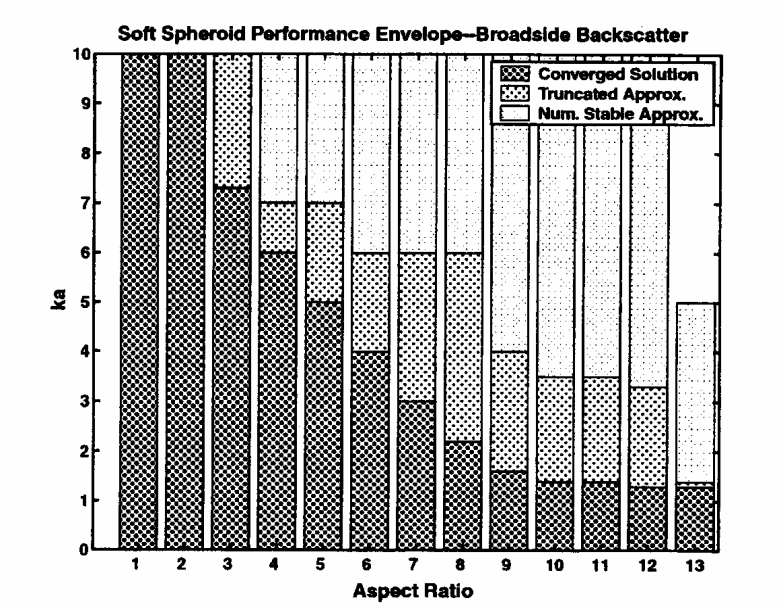


Figure 6. Performance envelope for a broadside incident plane wave for a smooth prolate spheroid under soft boundary conditions. The performance level is indicated by the shading and is plotted as a function of *ka* and aspect ratio. (From Reeder and Stanton, 2004)

Numerical techniques are implemented in this research with two principle goals: (1) extending the aforementioned performance envelope, and (2) making the lower modal combinations more accurate and, therefore, more useful. If either of these goals is achieved, then model performance is improved. The intention is to investigate changes in the performance envelope based on targeted changes in the formulation or calculation of results.

6. Accuracy versus Precision when Applied to Computers

In the course of this research, only a sphere and a prolate spheroid were considered as scattering shapes. For these shapes, the accuracy of the FMM was previously demonstrated by Reeder and Stanton (2004) in comparison to the Anderson (1950) exact solution, as shown in Figure 3, and other models for a prolate spheroid. While Reeder and Stanton (2004) addressed the problem of accuracy, they did not entirely address the issue of precision. Precision is inherent in any formulated computer calculation. Answers do not change from one model run to another, unless the inputs to that model are altered. However, another form of precision is considered here, which is

known as computer precision. With different values of precision, changes in the numbers of binary digits and decimal digits utilized in calculations will craft different answers.

The precision of a particular computer is only one facet of the computing process. The following sections reveal the complexity of computer calculations, while introducing the key computer fundamentals behind this research.

D. APPLIED COMPUTER TERMS AND PROCESSES

Nearly all numerical computing utilizes floating-point arithmetic. Almost all computers use the Institute for Electrical and Electronics Engineers (IEEE) binary floating-point standard to represent numbers for computer storage. In the late 1940s and early 1950s, many people were afraid that the intrinsic rounding errors of floating-point computing would make the results of complex calculations too inaccurate to be useful for science (Overton, 2001). That is still the fear for the FMM at higher frequencies and modes for complex shapes. In order to further understand this problem and to elucidate the scope of this thesis, some introductory material in floating-point, computer precision and other computer terminology is included.

1. Introduction to Floating-Point Arithmetic

A "bit" of computer storage space is a single binary digit. A "byte" is a group of 8 bits. A "word" is 4 consecutive bytes of computer storage space, or 32 bits, while a double-word is 8 consecutive bytes, or 64 bits. All real numbers have binary representations.

Floating-point representation is based primarily upon scientific notation. In floating-point form, the number S is called the *significand* while the number E is called the *exponent*. For example, the decimal number 0.00000625 can be expressed in scientific notation as 6.25×10^{-6} in base 10 notation, where 6.25 is the *significand* and 10^{-6} is the *exponent*. This is called floating-point because the decimal point in the decimal number "floats" to the position immediately after the first nonzero digit in the decimal expansion of the number (Overton, 2001). However, the computer standard is base 2, so the above expression would be written in the form $x = \pm S \times 2^{E}$, where $1 \le S < 2$, and the binary point floats to enable representation in this form. This representation is called the *normalized* form. In order to store a floating-point number, the computer word is divided

into three values, representing the sign of the number (i.e., 0 for a positive number and 1 for a negative number), the significand *S*, and the exponent *E*. For example, a 32-bit word could use 1 bit for the sign, 23 bits for the significand, and 8 bits for the exponent of a floating-point number (Overton, 2001). The binary expansion of the significand would then be:

$$S = (b_0.b_1b_2b_3...b_{23}) \tag{43}$$

where b_0 is always 1.

It is important to note that the bit on the left of the binary point (b_0) will always have the value 1 in binary representation for a floating-point number. In order to avoid using valuable storage space, this value is not stored and it is called the *hidden bit* (Overton, 2001). This is an important gain in precision computing, because it allows the storage of another binary digit.

While all real numbers have a binary representation, not all of them can be stored in a prescribed floating-point form. For a given amount of storage space, not all numbers can be represented within that numerical capacity. For the 32-bit word mentioned above, the exponent must fit in the range $-128 < E < 127$ because all 8 bits of storage space for the exponent would be filled with the binary representation of *E*, which is 01111111 for *E*=127 (Overton, 2001). A number larger than 127 would require more than 32 bits of storage space, which is not allowed by 32-bit computer architecture with only 8 bits prescribed for the exponent. Many numbers must be rounded before they can fit into floating-point form because the binary expansion must contain an exponential power of 2 within a prescribed range. This rounding process introduces error, which will be discussed later. Even the number 1/10 does not have a finite binary representation and will introduce error into a computing process (Overton, 2001).

2. Precision and Machine Epsilon

Within a given bit string, numbers can be prescribed for the significand and exponent. The number of bits stipulated for the significand (plus one for the hidden bit) is called the *precision* of the floating point-number. So, for the system described above

where S=23, the precision is 24. For any floating-point number x with precision p,

$$x = \pm(1.b_1 b_2 ... b_{p-2} b_{p-1})_2 \times 2^E . \tag{44}$$

This means that the smallest floating-point number greater than 1 is then:

$$(1.00...01)_2 = 1 + 2^{-(p-1)} . \tag{45}$$

The difference between this number the number 1 is called *machine epsilon*, which is usually denoted as *eps*, or $\in$. This can be expressed as:

$$\in = (0.00...01)_2 = 2^{-(p-1)} \tag{46}$$

Machine *eps* is the smallest number that can be discerned by a computer, which is also the smallest difference between numbers that can exist on any given machine. With each different value of machine precision, *eps* changes. So, by changing machine precision, which changes the number of binary digits in the storage space of S, the machine *eps* is also specified. However, machine precision cannot be changed on a given computer unless different hardware is installed to facilitate changes in numerical storage capabilities.

Higher values of precision allow a larger range of numbers to be stored within a computer. The largest number that can be stored by a computer is increased, while the smallest number that the computer can store is decreased. Higher precision therefore leads to increased technical capabilities for a computer system.

Precision represents a challenge for certain calculations. Most computations require only a basic level of precision; however, certain computations require a large degree of precision in order to yield results that are accurate enough to be useful. It is believed that the FMM is currently limited by machine precision for higher modes and for shapes of high complexity and eccentricity. According to Reeder and Stanton (2004),

> The FMM generates a transition matrix, much like the T-matrix model (Waterman, 1968), that relates the incident field coefficients to the scattered field coefficients. For a spherical scatterer, the transition matrix is diagonal and each nonzero term on the main diagonal is an eigenvalue for each mode computed. If the scatterer shape deviates from spherical, the matrix contains off-diagonal terms. The additional higher modal terms

> required to represent the scattering become extremely small, sometimes falling below the value that can be accurately represented numerically, resulting in a singular matrix in which the true values of its elements are below the precision of the machine. (Reeder and Stanton, 2004)

The resurgent problem is that the actual elements of the ill-conditioned transition matrix in the FMM require more binary digits and smaller machine epsilon than 32-bit precision, and even 64-bit precision, will allow.

Before proceeding, it is important to note that there are two forms of computer precision—machine precision and software precision. Machine precision is a set value on any given computer, based on how that machine was built, and the value of precision that is most commonly referenced in literature. Machine precision governs how fast a computer will perform a calculation, based on the number of bits used for storage. However, certain software programs introduce another facet to the concept of precision. Tools have been developed to allow software precision changes on the fly that will allow certain computations to be carried out at lower or higher precision than the machine architecture would otherwise allow. Software precision is conceptually identical to machine precision but can be changed with clever computer coding. MATLAB is an example of a software program that can run calculations at a higher precision than that of the machine on which it runs (i.e., MATLAB can run calculations at double precision, even if it is running on a computer that was built with single precision). For example, the version of MATLAB employed in this research uses two 32-bit words to store a single floating-point number, effectively gaining 64-bit precision via software syntax that instructs the computer to do so.

3. Single, Double, Extended, and Variable Precision

The IEEE standard has two fundamental formats, which they call *single* and *double*. The 32-bit word is the typical mode of storage for computers. Single format numbers use a 32-bit word as their storage mechanism. However, single format does not offer enough binary digits for applications where higher precision is needed or where a greater exponent is required. Double format uses a 64-bit word, which offers a larger

number of binary digits for the significand and exponent of the normalized representation. For double format, the exponent may take values within the range:

$$-1022 < E < 1023 . \tag{47}$$

Table 1 displays a summary of the ranges of E and the maximum (N_{max}) and minimum (N_{min}) numbers that single and double formats can handle. Table 2 shows how double precision numbers are stored within the exponent and significand.

Format	E_{min}	E_{max}	N_{min}	N_{max}
Single	-126	127	$2^{-126} \approx 1.2 \times 10^{-38}$	$\approx 2^{128} \approx 3.4 \times 10^{38}$
Double	-1022	1023	$2^{-1022} \approx 2.2 \times 10^{-308}$	$\approx 2^{1024} \approx 1.8 \times 10^{308}$

Table 1. Range of Numerical Values for IEEE Floating Point Single and Double Precision Formats. (From Overton, 2001)

The IEEE standard also has an *extended* format, which holds 15 bits for the exponent, 63 bits for the significand, one digit for sign, and one for the leading (hidden) bit that is not hidden in this format (Overton, 2001). Machines that run the extended format often run more slowly than their single and double precision counterparts, because every numerical value is stored with extended precision. Table 3 shows the precision and *eps* values for the various IEEE standard floating-point formats.

Machine precision cannot be changed without changing the actual hardware storage devices that are built into a computer. In order to circumvent the current limits on machine precision, symbolic representations of variables and variable-precision arithmetic (VPA) can be used to increase accuracy and precision for certain calculations. Symbolic mathematics that enable variable precision and rational representations of numbers are discussed in more detail in Chapter III.

±	$a_1a_2a_3...a_{11}$	$b_1b_2b_3...b_{52}$

If exponent bitstring is $a_1...a_{11}$	Then numerical value represented is
$(00000000000)_2 = (0)_{10}$	$\pm\ (0.b_1b_2b_3\ldots b_{52})_2 \times 2^{-1022}$
$(00000000001)_2 = (1)_{10}$	$\pm\ (1.b_1b_2b_3\ldots b_{52})_2 \times 2^{-1022}$
$(00000000010)_2 = (2)_{10}$	$\pm\ (1.b_1b_2b_3\ldots b_{52})_2 \times 2^{-1021}$
$(00000000011)_2 = (3)_{10}$	$\pm\ (1.b_1b_2b_3\ldots b_{52})_2 \times 2^{-1020}$
↓	↓
$(01111111111)_2 = (1023)_{10}$	$\pm\ (1.b_1b_2b_3\ldots b_{52})_2 \times 2^{0}$
$(10000000000)_2 = (1024)_{10}$	$\pm\ (1.b_1b_2b_3\ldots b_{52})_2 \times 2^{1}$
↓	↓
$(11111111100)_2 = (2044)_{10}$	$\pm (1.b_1b_2b_3\ldots b_{52})_2 \times 2^{1021}$
$(11111111101)_2 = (2045)_{10}$	$\pm (1.b_1b_2b_3\ldots b_{52})_2 \times 2^{1022}$
$(11111111110)_2 = (2046)_{10}$	$\pm\ (1.b_1b_2b_3\ldots b_{52})_2 \times 2^{1023}$
$(11111111111)_2 = (2047)_{10}$	$\pm\ \infty$ if $b_1 = \ldots = b_{52} = 0$, NaN otherwise

Table 2. IEEE Double Format with *E* represented by $a_1a_2a_3...a_{11}$ and *S* represented by $b_1b_2b_3\ldots b_{52}$. The left-hand column shows binary and decimal representation of the exponent strings. The right-hand column shows the normalized representation of the numerical value. (From Overton, 2001)

Format	Precision	Machine Epsilon
Single	p=24	$\in = 2^{-23} \approx 1.2 \times 10^{-7}$
Double	p=53	$\in = 2^{-52} \approx 2.2 \times 10^{-16}$
Extended	p=64	$\in = 2^{-63} \approx 1.1 \times 10^{-19}$

Table 3. Precision of IEEE Floating-Point Representations. The precision, *p*, is the number of binary digits in the significand of the normalized form of the stored value. Machine epsilon is the smallest distinguishable number associated with the given format. Although the precision seems high in binary form, the decimal equivalent of these numbers of binary digits of precision is much lower. For example, double precision floating-point numbers in binary representation with 53 bits of precision must be rounded to their decimal equivalents with only 15 or 16 decimal digits. Thus increasing the number of decimal digits used in computation can make a much greater impact than increasing the number of binary digits by the same value. (From Overton, 2001)

4. Computational Expense and Realistic Limits

The term "expense" refers to the amount of computational time and computer storage that is required for computer operations. Computational storage increases with higher precision, because more bits are required for storage of larger bit strings. Computational time requirements also increase as precision increases, since more 1's and 0's need to be placed into the bit strings with each operation. At a given value of precision, computational time can only be lessened with the use of a faster computer processor or a more efficient mathematical formulation. This means that the potential use of higher precision is reliant upon the amount of time available for computer operations. Higher precision is more expensive to the end user that is cognizant of limited resources.

E. COMPUTATIONAL ERROR

1. Introduction to Computer Error

As mentioned above, rounding takes place when a number cannot be stored within a bit string of prescribed length. Numbers are rounded when converted from decimal to binary for computation, and they are again rounded when they are converted from the binary storage form to decimal screen output. In addition, when numbers are added, subtracted, multiplied or divided, the result might be a number that cannot be represented in floating-point without being rounded. This is called *integer overflow*, which the computer corrects by rounding the number before it can be stored. This compounds the error that may have started with numbers that were not floating-point numbers to begin with and required rounding.

In a simple addition such as:

$$1/2+1/3,$$

there are really three *roundoff errors* associated with the output. Note that 1/2 is actually a power of 2 and does not require rounding prior to the binary storage of the number. The first roundoff error is encountered with the division of 1 by 3 when the computer has to round in order to store the result of the division within the prescribed number of bits in the significand of the associated IEEE format. Another error is introduced with the addition of 1/2 to the result of the aforementioned division, because this result must also be rounded and stored. The final error occurs when the binary result is converted to

decimal format for the printed output. When IEEE double floating-point arithmetic is used to calculate this result, 64 bits are used for the stored result, but only 53 are used in the binary expansion of the significand. This usually gives about 16 decimal digits, when the conversion is made. However, only 15 digits print to screen in this specific case, with the result showing:

0.83333333333333 (The Mathworks, Inc., 2002).

This result is not the exact answer to the problem above because it incorporates the three errors that were introduced during its creation.

2. Compounding Error – The Importance of Precision

As already mentioned, a calculation performed below a required level of precision will incorporate roundoff error into the answer. This occurs because the real values have more decimal or binary digits than the computer can store within its floating-point representation. The mathematical solution becomes limited by machine precision, so that the computer solution does not accurately represent the actual intended computation. If this rounded answer is then used in further calculation, it can propagate and multiply the roundoff error by performing various mathematical operations on an already erroneous number and heavily impact the accuracy of the final solution. This happens frequently in models such as the FMM, in which the smallest of numbers is important to the mathematical computation of an accurate answer.

Small changes in initial conditions within a model should produce small changes within the results. In this situation, the model is said to be *stable*. However, if a small change in initial conditions produces a drastically different answer, then the model is said to be *unstable* (Mathews and Fink, 1999). The errors discussed above may introduce instabilities throughout the course of model calculations.

F. NUMERICAL TECHNIQUES

1. Operator Modifications and Reducing Roundoff Error

Addition, subtraction, multiplication and division all inherently incorporate different amounts of roundoff error. With each calculation, there is an associated error either for storage or output to the computer screen. An efficient mathematical formulation reduces the number of operations that must be performed. When fewer

mathematical operations are performed, fewer incidences of rounding occur which consequently minimizes roundoff error.

2. Singular Value Decomposition

Singular value decomposition (SVD) is a mathematical tool that assists in handling series of equations or matrices that are singular or nearly singular (Press *et al.*, 1992). According to Press *et al.*, "Any *M* x *N* matrix **A** whose number of rows *M* is greater than or equal to its number of columns *N*, can be written as the product of an *M* x *N* column-orthogonal matrix **U**, an *N* x *N* diagonal matrix **W** with positive or zero elements (the *singular values*), and the transpose of an *N* x *N* orthogonal matrix **V**." This technique is extremely useful in the case of an ill-conditioned matrix that includes elements below machine precision. The SVD technique sets elements in a matrix below a certain threshold to zero.

Essential to the FMM are several values within the transition matrices that are extremely small. These values can make important contributions to the solution if they are real parts of the solution, but they are often introduced into the problem by roundoff error. In the case of the Reeder and Stanton (2004) model, SVD removes all elements below a manually set threshold that are zero or near zero within the calculated matrices. "Singular values whose ratio to the largest singular value is less than *N* times the machine precision are set to zero" (Press *et al*., 1992). Problems exist when this threshold is set too high or too low. If it is set too high, then some values will be removed from the matrices that are actually valuable parts of the scattering solution. If the threshold is set too low, many values that are supposed to be zero but are non-zero because of roundoff errors may be left in the solution as erroneous contributions to the scattering solution. The threshold for SVD should be set at-or-below the value of *eps* associated with the increased precision. This will prevent the gains of increased precision from being removed by the SVD commands, which conduct valuable quality control for model calculations.

Singular value decomposition can be a valuable and powerful tool when used properly. Its intention is to remove "erroneous subspaces" in order to produce a more stable result (Reeder and Stanton, 2004). If used incorrectly, SVD can eliminate matrix

values that are positive contributions to the solution. Incorrect implementation could result in reduction of amplitude and damage to the composition of the solution.

G. EXPECTED RESULTS

The planned model improvements should reduce or eliminate model instability that occurs in many of the FMM's higher modal combinations due to compounding roundoff error. The roundoff errors are inherent in nearly all floating-point numbers in the model calculations, and the improvements made in this research are aimed at decreasing the roundoff error to a minimal amount. The improved model should have less instability in the higher modal combinations, in the higher frequency ranges, and when applied to shapes of greater eccentricity and complexity.

Increased accuracy is another anticipated attribute of the improved FMM. If the increased precision can add extra decimal places or binary digits to the values used in calculations, then the expectation is to see greater accuracy in the results of the improved model in comparison to the results of the FMM used in Reeder and Stanton (2004).

Implemented techniques will also aim to improve the performance envelope of the FMM. If the model is more accurate, fewer modes will be required to arrive at a converged solution. This may reduce the overall number of modes that are needed to gain useful results.

THIS PAGE INTENTIONALLY LEFT BLANK

III. IMPLEMENTATION OF THEORETICAL IMPROVEMENTS

The following sections describe the background work that was required to implement various numerical techniques, including information on hardware, software and coding routines that were used to gain numerical results. This chapter covers the specific tools within MATLAB and the numerical techniques that were used to effect changes in mathematical calculations in the FMM, in addition to the methods for evaluating the research results.

A. TERMINOLOGY AND RESEARCH SETUP

In the following discussion of implementation and results, "original" model refers to the model developed by Reeder and Stanton (2004). Additionally, the "improved" model refers to a specific version of the FMM that incorporates the individual changes that are explicated in the following sections. In most cases, the improved model is compared to the original model through direct comparison of output plots, in order to ascertain the added value of the executed technique. The exact formulations to which Reeder and Stanton compared their results, such as the Anderson (1950) solution, are not used in this research, because they too incorporate roundoff errors when calculated at double precision.

B. HARDWARE AND SOFTWARE SPECIFICATIONS

1. Hardware

The machines used for the bulk of this research are Dell Precision 670 n-series machines with Xeon processors. These machines are built to use single precision with 32 bits of storage space for each floating point number. Although these are not the fastest or most robust machines on the market today, they do represent the average technical computing machine that would be used in a non-research, or operational, environment. This provides an element of realism to the research, since the results can be directly applied to the operational environment that uses similar machines. Most machines that are currently being built incorporate 64-bit machine precision, so the research platforms are already surpassed by the storage capabilities of the newest systems.

2. Software

All computer code generated for this research was created or modified in MATLAB Version 7.0.1.24704 (R14) Service Pack 1, which is produced by The Mathworks, Inc. The 3.1.1 (R14) Service Pack 1 version of the Symbolic Math Toolbox was used for all work associated with symbolic numbers and variable-precision arithmetic.

Although the machines described above run at single precision, MATLAB stores variables and performs calculations in double precision, doubling the actual machine precision through the software syntax. This was verified using the *eps* command within MATLAB, which displayed *eps* values that corresponded to double precision on a machine that was known to run at single machine precision.

The Dell machines used in this experiment ran Linux Kernel 2.4.21-32.0.1.ELsmp #1 SMP May 17 17:52:23 EDT 2005 i686 for all applied research. The Linux operating system is very supportive of technical computing and proved to be a reliable research tool.

C. INCREASING PRECISION THROUGH SYMBOLIC MATHEMATICS

The Symbolic Math Toolbox within MATLAB encompasses over 100 functions that allow exploitation of the Maple kernel through the use of specific syntax in the MATLAB language (The Mathworks, Inc., 2002). This enables symbolic calculations to be performed within the floating-point MATLAB arena. This is an extension of the capabilities of MATLAB that enhances the standard computing and plotting tools.

The use of the functions within the Symbolic Math Toolbox creates a variable of a new class, called *symbolic object*, or *sym* for short, which is not a floating-point number. It is a symbolic representation similar to that created when doing calculations with pencil and paper. Internally, the computer stores the *sym* as a character string. *Syms* can be used to represent symbolic variables, expressions, and matrices. To obtain the numerical value of a *sym*, the *double* command must be used, which converts the *sym* into a floating-point number. It is important to note that this conversion requires the number to be rounded to the nearest floating-point value. MATLAB also performs calculations differently with values of class *sym* than it does with floating-point numbers of class

double. Operations performed with double precision will yield answers in *double*. Operations performed with *syms* will yield answers with rational form as *syms*. MATLAB actually searches for the lowest common denominator of the fractions represented by the *sym* numbers and performs calculations in rational arithmetic, with the help of Maple (The Mathworks, Inc., 2002).

Three general representations of *syms* are possible within the Symbolic Math Toolbox utility, including numerical (regular floating-point), variable-precision arithmetic (user-defined), and symbolic representation (rational/exact). These representations are all different, but are all class *sym* variables. Floating-point, or numerical, operations are the least expensive in computational time and memory out of the three forms of *sym* variables, but the results are not exact due to associated roundoff errors. Rational operations use exact rational values without any error in their representation, given that the numbers required for calculation can be represented as ratios or integers. MATLAB stores *sym* numbers in rational form (i.e., 1/5), rather than using floating-point representation. However, symbolic numbers take the most computer time and memory to compute out of any of the available numerical forms in MATLAB (The Mathworks, Inc., 2002). Calculations performed with variable-precision arithmetic (VPA) are between the other two forms with regards to expense, but may afford somewhat of a balance of accuracy and computational time for some calculations. The following paragraphs describe the two specific symbolic forms that were used in this research.

1. Variable-Precision Arithmetic

One way in which machine precision can be extended is to use VPA. With the *vpa* command that is part of the Symbolic Math Toolbox in MATLAB, a user can set precision to a desired value for certain calculations. VPA uses the *digits* command to set the number of decimal digits of precision at which a calculation will be made with symbolic math and variables of class *sym*. Mathworks, Inc. says that the *vpa* command uses "variable precision floating-point arithmetic with D decimal digit accuracy, where D is the current setting of *digits*" (MATLAB Help Document for *vpa.m*). The resulting expression from *vpa* is a symbolic value or array that is similar to a character string.

With *digits*=10, the number 1 is represented by VPA as "1.000000000" throughout all subsequent calculations, with the value 1 counted as one of the ten digits set by *digits*.

The default value of *digits*, unless otherwise specified, is 32 decimal digits. In other words, the number "1" would be expressed as the number 1 followed by 31 zeroes after the decimal point. This increases the accuracy that can be carried through calculations by avoiding roundoff error until the last decimal digit of a long string of decimal digits, whereas double precision introduces roundoff error to store as binary and then rounds again to convert to sixteen decimal digits of output. Thus, roundoff error is still an unavoidable part of VPA calculations, but it is effectively reduced for calculations in which fewer than D digits are required to accurately represent the solution.

Even by using VPA with *digits*=16, which is the same as the MATLAB default output in double precision, the outcome of certain calculations may be improved. This is because VPA values with increased numbers of decimal digits are stored and carried through calculations as character strings, rather than using binary storage and floating-point operations. However, VPA must be used in a way that exploits precision to gain accuracy in smaller decimal places. Otherwise, computational expense is added with no numerical benefit. In the FMM, higher modal combinations use this extended precision afforded by VPA. VPA is the primary tool used in this research.

Implementation of VPA into the Reeder and Stanton (2004) FMM introduced a multitude of programming errors that had to be overcome. Commands such as *max*, *find*, and even *sqrt* can not handle the symbolic variables that are created with the *vpa* utility. Some of these difficulties were overcome via the creation of "dummy" variables within MATLAB. The dummy variables are mirror images of the original variables passed to the *max*, *find*, and *sqrt* commands. Before the original variables are passed to these commands in which errors occur, they are converted from symbolic form back to double precision floating-point format by using the *double* command. The variable can then be reset to its original precision after bypassing the command in which it would have otherwise produced an error. This is accomplished by setting the original variable name equal to the dummy variable. By using this technique, subsequent lines of code do not have to be modified to accommodate new variable names or classes.

The VPA techniques described above were the primary means of improving the FMM. The *vpa* command was used throughout the entire model, where possible, increasing the precision of the majority of calculations performed by the FMM.

2. Symbolic Variables and Rational Arithmetic

The *sym* command within MATLAB creates variables that can be of a few different forms: floating-point, rational, expressions in terms of *eps*, and decimal representations with the number of decimal digits set by *digits*. The rational numbers created by the *sym* command introduce an even higher level of accuracy into computer calculations at the expense of significantly greater computational time.

In this research, the rational representation of *sym*, which is the default used by MATLAB if not otherwise specified, was used to generate exact rational numbers. In this way, calculations performed with *sym* are done with exact numbers and avoid roundoff errors inherent in floating-point and *vpa* calculations. If a number cannot be represented in rational form, then the result will be an irrational number in floating-point format with default precision.

Symbolic math with rational numbers eliminates any and all roundoff errors associated with the symbolic variables, because the rational numbers do not require rounding. The *sym* command would produce exact results for the FMM, if all values could be carried through the model as rational numbers. The *sym* command was implemented into the FMM to evaluate another of the symbolic variable formats, but the computational expense that was inherent in the associated calculations was too great to compute even the first mode in the FMM solution. The model would not run to completion. Errors were encountered before the model could produce results even at mode m=1, n=1. These errors are described at the end of this chapter. The rational numbers produced by *sym* are known to be the most expensive of all variable forms within MATLAB, so the errors experienced here were not surprising.

D. NUMERICAL TECHNIQUES

1. Operator Modifications

With each calculation that is conducted within MATLAB, there is inherent roundoff error. The simplest technique in reducing roundoff error is to reduce the

number of required calculations. In order to do this, all calculations that can be simplified are reduced to their most basic form. Reducing the total number of calculations, even if only by a few, reduces the number of rounding errors incorporated into and amplified within the solution. In certain calculations, the roundoff error could be adding a large enough amount of error that it could affect the solution quite significantly. The FMM code was carefully modified in this research to reduce the total number of calculations being performed by the model.

2. Singular Value Decomposition

Singular value decomposition was not treated in this research because the modal combinations where SVD made significant changes to the answer in the original FMM were outside the capable range of the improved FMM. The range of the improved FMM will be discussed more in Chapter IV. The SVD processes that were incorporated into the model by Reeder and Stanton (2004) remained the same for this research.

E. EVALUATION TECHNIQUES

1. Visual Inspection of Reduced Target Strength (*RTS*) Versus Non-dimensional Frequency (*ka*)

Results from the original Reeder and Stanton (2004) model are plotted simultaneously with the results of the improved model. Graphical comparison of the solution achieved with the improved model to the results of the original model shows improvements in the solution.

2. Measurements of Computational Time

Computational time was recorded for each model run with the use of the *tic* and *toc* commands within MATLAB. The time was started at the beginning of each model run with *tic*, and the runtime was recorded at the conclusion of the calculation of each modal combination with *toc*. In this way, a cost-benefit analysis could be conducted on the results of the model runs, based upon the model's runtime and performance gains.

F. MATLAB/CODE LIMITATIONS

One of the major limitations on the use of VPA was the symbolic form of the output. Calculations with symbolic numbers could be conducted to a certain extent before errors were encountered in MATLAB in association with the variables of class *sym*. The symbolic numbers had to be evaluated with the *eval* or *double* command in

order to convert them back to double precision so that they could be used in further calculations beyond the point of the errors. This is desirable because the higher precision values could be carried through most of the model calculations, and the more accurate results from those calculations could then be used in further computations in double precision.

In addition to the *max*, *find*, and *sqrt* commands mentioned earlier, other operations and commands that posed challenges to the implementation of symbolic variables included indexing operations within matrices and saving variables in symbolic format. The *squeeze* command that is used within the integration process in the FMM would also not work with variables of class *sym*. This meant that the implementation of the VPA or *sym* variables could only be carried through the end of the calculations of the Bessel functions and associated Legendre functions. Once the matrices for these functions were populated by the model, the results had to be converted back into double precision for further computation with the *squeeze* command. The errors in this paragraph were encountered when *sym* variables were introduced into lines of computer code containing the commands listed above. This meant that the variables had to be changed to double precision to carry them through these lines of code.

The most significant error was one that limited the number of modes that could be computed by the improved FMM once VPA was already implemented. An insurmountable error encountered within MATLAB at high modal combinations while running calculations with *vpa* variables was displayed as "Error, integer too large in context" in the MATLAB command window. This error corresponds to Mathworks Technical Solution number 1-1AG3M on the Mathworks.com support website. The solution reads:

> Our development staff has been notified and is currently looking into addressing this problem in a future release of the Symbolic Math Toolbox. This may be a problem with the way memory management is performed by Maple. At present, the only potential workaround is to wrap your call to a symbolic calculation that operates on numbers with large numbers of digits inside a TRY/CATCH block and clear the Maple function… (The Mathworks, Inc., 2005)

This statement at least shows promise for the next-generation Symbolic Math Toolbox. However, the "workaround" presented here does not solve the problem. It merely steps around the error message and proceeds without the numbers that generated the error in the first place. Technical improvements may soon be incorporated into MATLAB that would allow some of the roadblocks associated with the symbolic mathematics in this research to be avoided in future work with the FMM.

IV. NUMERICAL RESULTS

A. GENERAL CHARACTERISTICS OF THE IMPROVED MODEL

There are several specific changes and traits to note in the improved model, but the general changes are noted first for perspective. First, the improved FMM requires a much greater amount of computational time than the original FMM. This was expected because of the use of symbolic variables, in addition to the increased precision. Second, very few modes could be calculated with the improved model because of the computational errors inherent in MATLAB, which were discussed in Chapter III. The model encountered multiple errors, mostly due to several MATLAB functions' incompatibility with the symbolic variables created by the *vpa* and *sym* commands. Third, the research exposed many findings during the implementation of VPA that have been left as items for follow-on research. The following paragraphs expound upon some of these general characteristics and offer insight into the model output that was obtained with the improved model.

B. RESULTS OF EXTENDED PRECISION AND NUMERICAL TECHNIQUES

A variety of techniques were implemented to improve the FMM. The applied operator modifications for addition, subtraction, multiplication and division operations did not produce any noticeable changes in the results of the original FMM. This is most likely because there were no apparent divisions by extremely small numbers or multiplications of extremely large numbers where the last few significant digits of the computed values make a large contribution to the end result. Although no improvements were observed in the results of the improved FMM based on the operator modifications alone, the code was modified for efficiency to ensure that only the smallest number of calculations required would be conducted by the improved FMM when precision was later increased. Increases in precision were the next step in improving the model.

The results of the improved FMM are revealed in the following paragraphs and plots. Model updates used in producing these results include VPA and operator modifications. The effects of SVD are discussed as a point for further research.

Through several iterations of attempted VPA implementation, the *TS* vs. *ka* plots did not change. This information suggested three possible situations: (1) VPA was not being implemented correctly, (2) the improved results from the VPA implementation were being negated by roundoff or SVD, or (3) VPA did not make a difference in model performance. The first two of these situations were investigated in-full to guarantee the validity of the results. This led to several changes in the VPA implementation, which resulted in several discoveries in the way that *vpa* actually works with the computer-stored variables. The stored variables were checked after each line of code to guarantee that the specified level of precision was being carried through calculations. Once VPA was determined to have been implemented correctly, the model was run again, and it produced significant differences from the output of the original model. The VPA technique was the most effective of all of the executed numerical improvements.

Figure 7 shows the first modal combination of the improved FMM with *digits*=16, which produced the same numerical results as the original FMM. The fact that the improved model curve is exactly the same as the original model curve for the initial modal combination of *m*=1, *n*=1 gives confidence in the execution of VPA.

The improved FMM yielded results for only the first two modal combinations (i.e., *m*=1, *n*=1 and *m*=1, *n*=2) with *digits*=16 for a soft sphere. This gave an initial indication that the improved FMM would be extremely limited in the range of modes that it would compute for higher values of *digits*. The improved FMM was run with *digits*=24, *digits*=32, and *digits*=40, corresponding to 1 ½, 2, and 2 ½ times the precision of the original FMM, respectively. These runs yielded results for the first two modal combinations for all values of *digits* and results for the third mode (i.e., *m*=1, *n*=3) for *digits*=32 only. Because results for three modal combinations were attained for *digits*=32 and because 32 digits corresponds to exactly twice the precision of the original FMM, the results for *digits*=32 were used for further analysis.

The improved FMM results for *digits*=32 are shown in Figure 8. For *m*=1 and *n*=2, shown in Figure 8(b), the improved model deviates from the original. The shape of the *RTS* vs. *ka* plot for this modal combination suggests that the addition of one mode at

higher precision improves the result of the original model at the same modal combination by extending the converged, smooth portion of the curve to a higher value of frequency, or *ka*. The same holds true for mode *m*=1, *n*=3, which is displayed in Figure 8(c). The extension of the smooth portion of the curves in these plots indicates that the interference patterns in the representative mathematical functions are less destructive to the solution at lower values of *ka*. With less destructive interference at lower values of *ka*, convergence should occur with fewer modes within that range of *ka*.

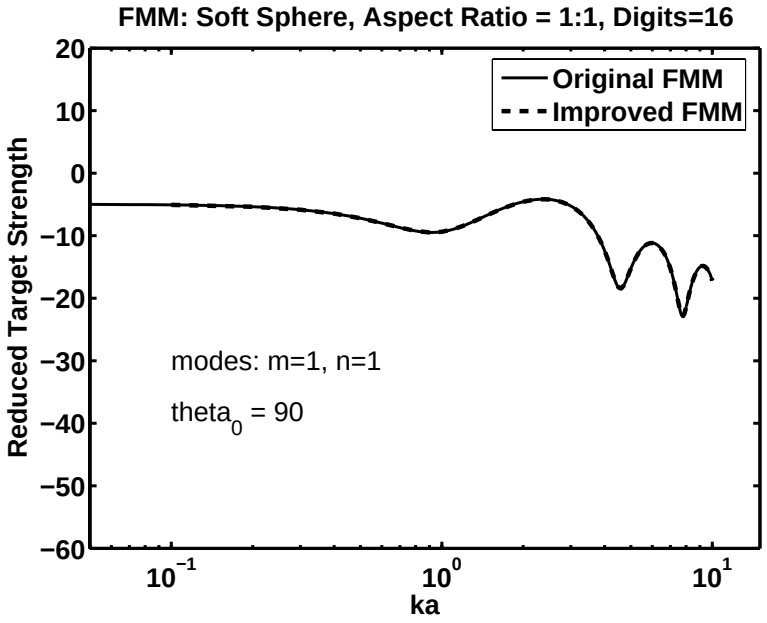
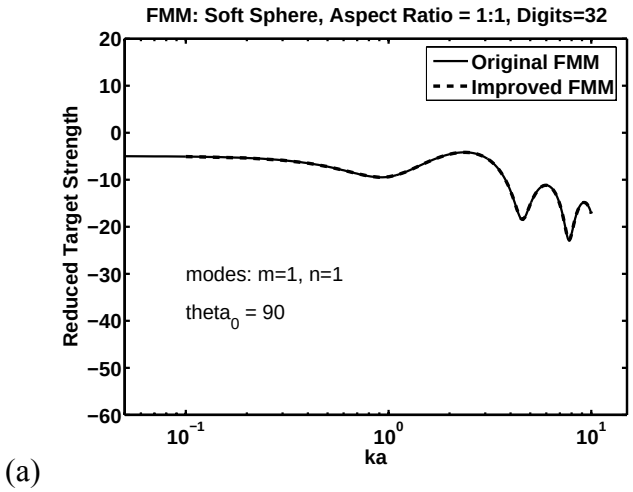


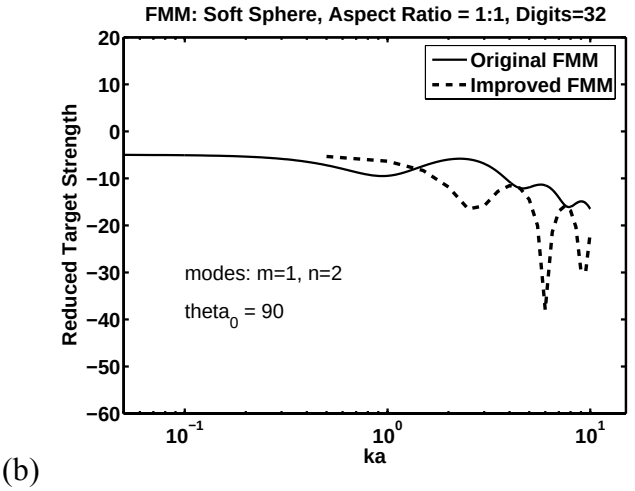
Figure 7. Original FMM and improved FMM for *m*=1, *n*=1. This is the first plot of results achieved with the improved model. With the improved FMM calculated at *digits*=16, which is the same as the original FMM, the identical curves give confidence that VPA was implemented correctly.

The improved FMM was also run for a prolate spheroid with *AR*=5:1. The results for the spheroid are shown in Figure 9. The improved FMM computed modes *m*=1, *n*=1 and *m*=1, *n*=2 for all four values of precision, but no further modes could be computed because of errors. Again, the original FMM results were duplicated by the improved FMM for the modal combination *m*=1, *n*=1.

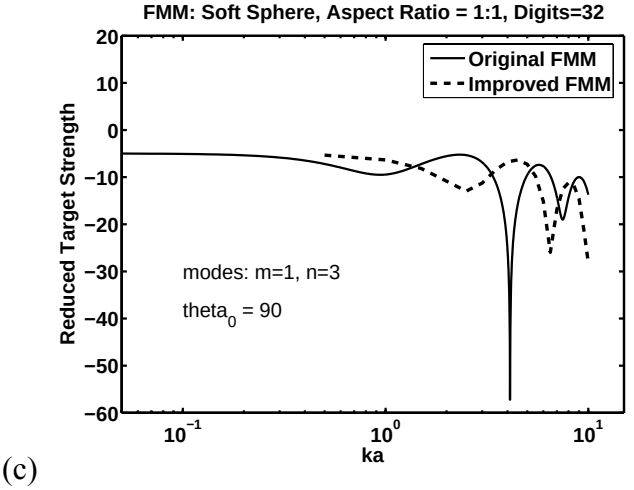
For both *AR*=1:1 and *AR*=5:1, the results of the improved FMM were identical for mode *m*=1, *n*=1 and also for *m*=1, *n*=2 for all four values of *digits* (i.e., 16, 24, 32, and 40). This result, which is shown in Figure 10, suggests that SVD is affecting the solution.



(a)

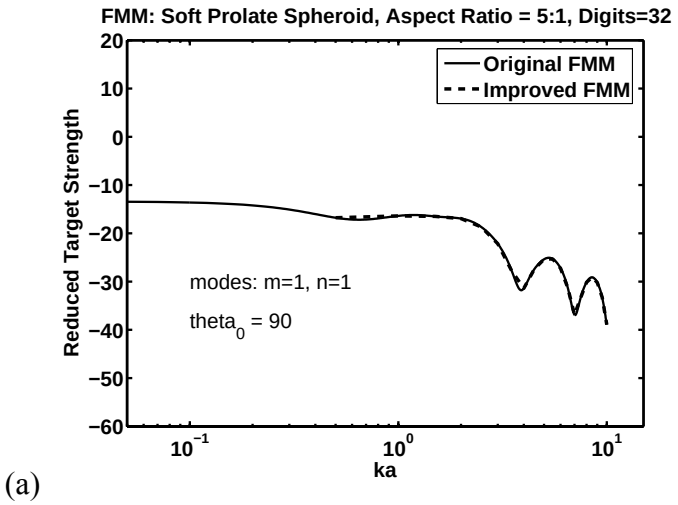


(b)

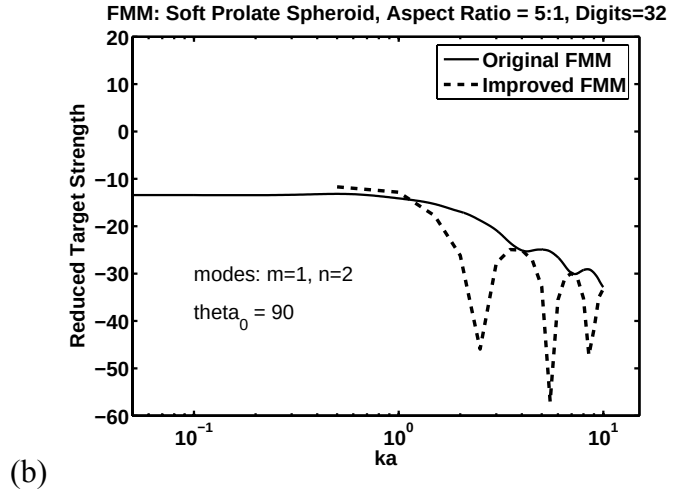


(c)

Figure 8. Modes *m*=1, *n*=1; *m*=1, *n*=2; and *m*=1, *n*=3 for the original FMM and improved FMM with *digits*=32 for a soft sphere. At *digits*=32, the improved FMM runs at twice the precision of the original FMM.



(a)



(b)

Figure 9. Modes *m*=1, *n*=1 and *m*=1, *n*=2 for the original FMM and improved FMM with *digits*=32 for a prolate spheroid with *AR*=5:1. These plots show distinct differences between the two models for these modes. The identical curves in window (a) give confidence that VPA and other techniques were implemented correctly. Precision of *digits*=32 is approximately double the precision of the original FMM.

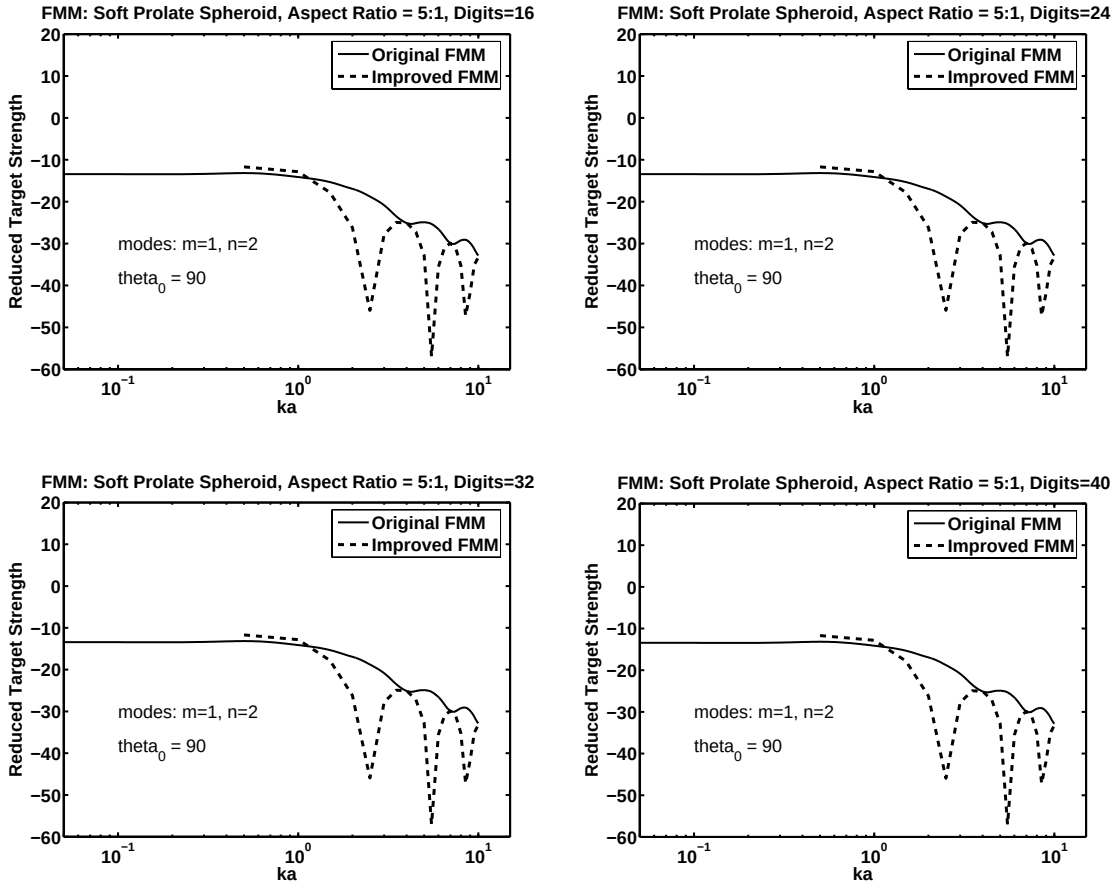


Figure 10. Mode *m*=1, *n*=2 of the improved FMM for a prolate spheroid with *AR*=5:1 at four different levels of precision—*digits*=16, 24, 32 and 40. The plots are identical, suggesting that SVD or another factor is affecting the results.

C. ANALYSIS AND DISCUSSION

1. Gains in Converged Range of *ka*

For the purposes of analysis, the converged *ka* region of the FMM is defined as the value of *ka* under which two consecutive modes overlap. The progression of the convergence of the original FMM is shown in Figure 11, with the three initial modes plotted on top of each other. With the addition of each mode, the curves overlap each other at higher frequency values. Figure 11 illustrates that the original FMM is converged for modes *m*=1, *n*=2 and *m*=1, *n*=3 out to *ka*=1.2. The progression of convergence for the three initial modes is also plotted for the improved FMM in Figure 12. Notice that the two initial modes (i.e., *m*=1, *n*=1 and *m*=1, *n*=2) do not overlap in the improved FMM solution, although they may be converged somewhere in the *ka* range

that is below the plotted range. However, the next two consecutive modes (i.e., m=1, n=2 and m=1, n=3) are converged in the illustrated frequency range.

Figure 11 and Figure 12 can be compared to show that modes m=1, n=2 and m=1, n=3 are converged to ka=1.2. This region is expanded in Figure 13 to show a close-in view of the point of convergence in both the original and improved FMM. For modes m=1, n=2 and m=1, n=3, Figure 13 confirms that the solution converges out to ka=1.2.

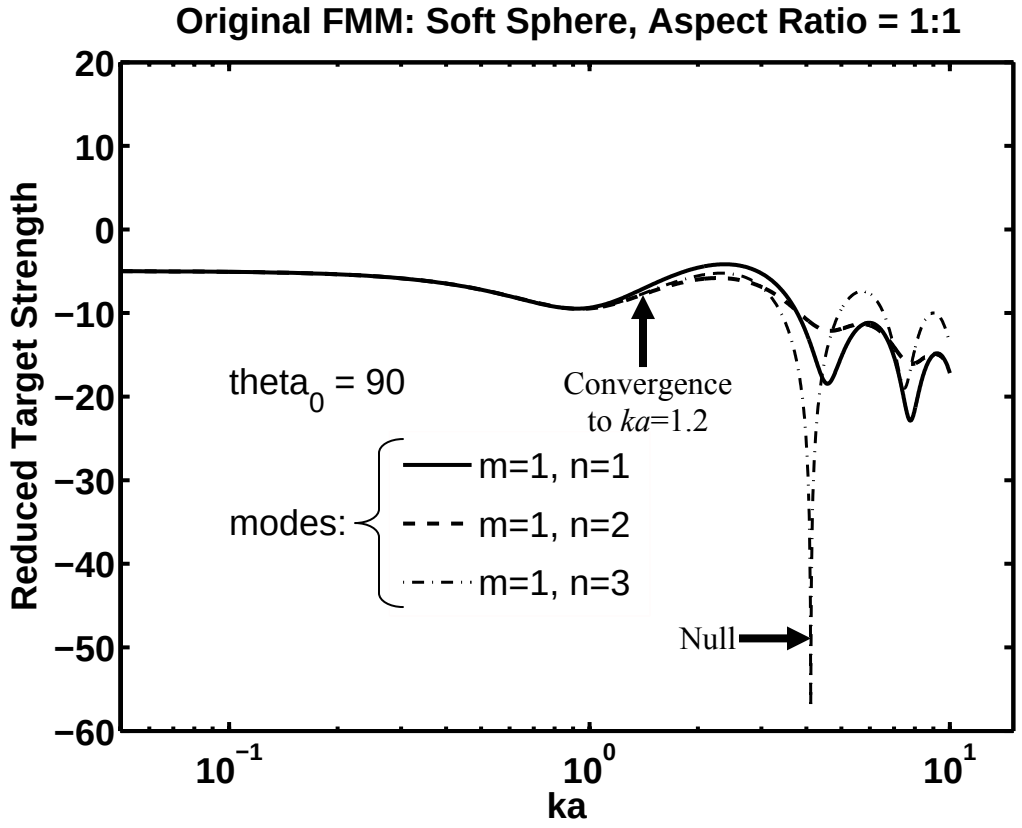


Figure 11. Original FMM progression of convergence with addition of higher modes. The point of convergence is extended to higher values of *ka* as higher modal combinations are added to the solution. The first null, caused by computed scattering interference patterns, is shown by the arrow at *ka*=4 for mode *m*=1, *n*=3.

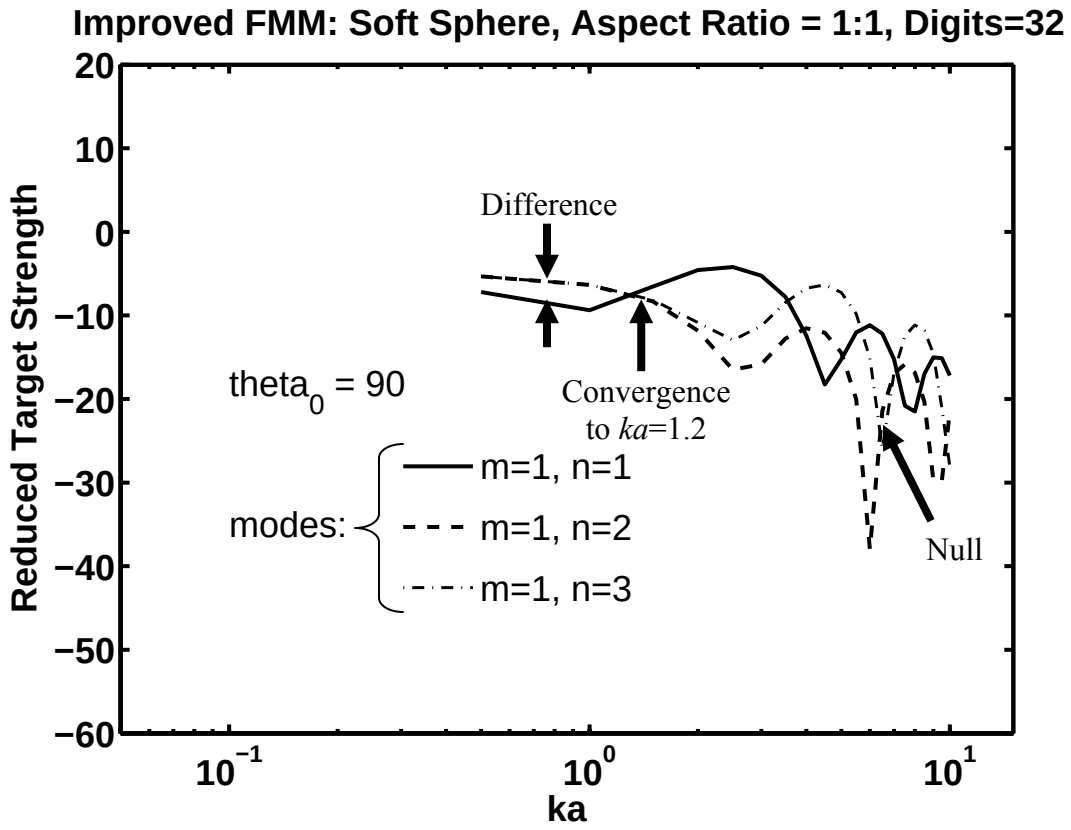


Figure 12. Improved FMM progression of convergence with addition of higher modes. The point of convergence is extended to higher values of *ka* as higher modal combinations are added to the solution. The improved FMM exhibits as much stability as the original FMM out to *ka*=10. The first null is reached in mode *m*=1, *n*=1 at *ka*=4.5. The null is shifted to the right to *ka*=6.5 for *m*=1, *n*=3, which is 2.5 units higher into the frequency range than the original FMM predicted for the same mode.

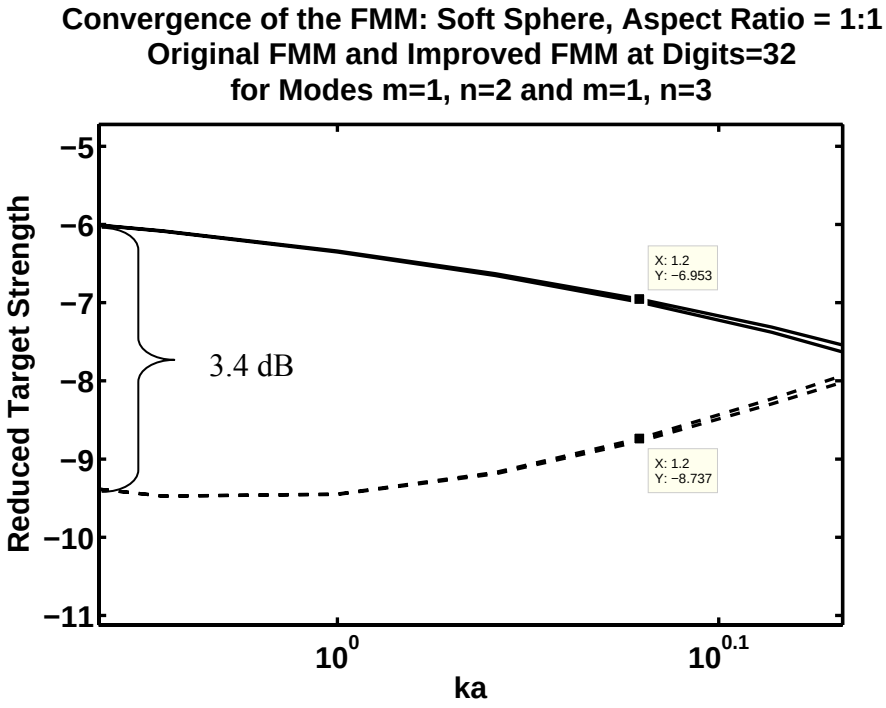


Figure 13. Convergence of the original FMM and improved FMM for modes *m*=1, *n*=2 and *m*=1, *n*=3. The dashed lines indicate the two consecutive modes of the original FMM while the solid lines indicate the two consecutive modes of the improved FMM. The value of *ka* at which the two curves depart from one another is very close for both models at about *ka*=1.2.

While it appears as though the converged, or overlapping, portion of the curves has not changed with the model improvements incorporated into the improved FMM, there is another feature evident in Figure 11 and Figure 12. The null, which results from destructive interference patterns in the various functions of the model calculations, is shown at *ka*=4 in the original FMM for mode *m*=1, *n*=3 in Figure 11. Figure 12 shows that this null is shifted to a higher frequency of *ka*=6.5 for the same mode in the improved FMM. This increase of 2.5 in the range of *ka* marks a significant performance gain for the improved FMM. This suggests that the overall solution, while not overlapping beyond the point at which the original FMM does for these low modes, is a better characterization of the scattering phenomena and that it is closer to the true converged solution. This concept is depicted more intricately in Figure 14, which shows a comparison of the original FMM and improved FMM at mode *m*=1, *n*=3 with the original FMM converged solution at *m*=15, *n*=30. Considering the reduced target strength (*RTS*)

over the entire range of dimensionless frequency (*ka*), rather than just the portion where consecutive modes overlap, the improved FMM appears to be a more stable approximation to *ka*=6.5.

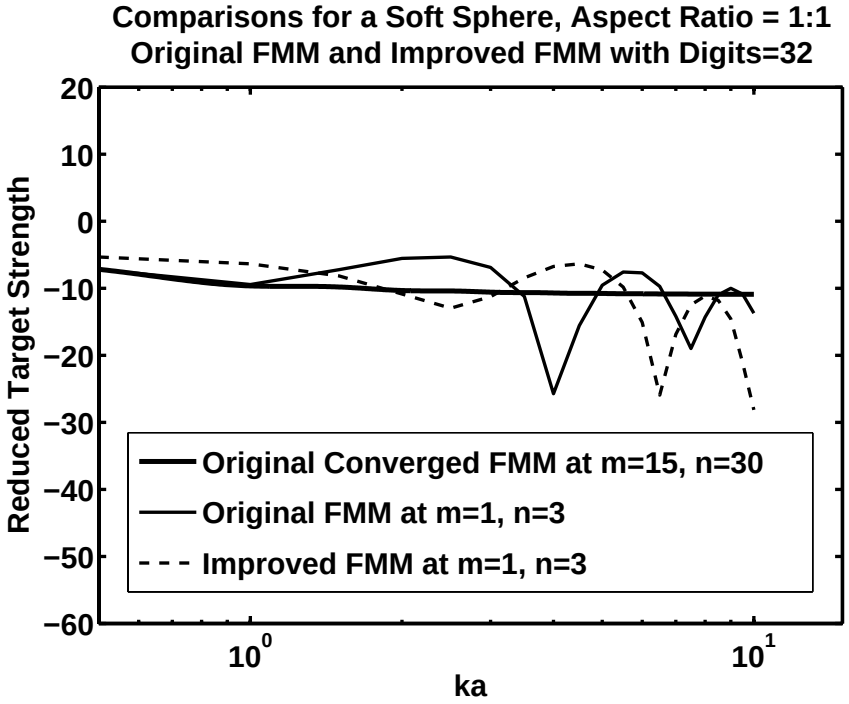


Figure 14. Comparison of mode *m*=1, *n=3* of the original FMM and of the improved FMM at *digits*=32 to the converged solution of the original FMM. The first null space is reached at *ka*=3.8 for the original model, while the improved FMM shifts the first null to *ka*=6.5.

2. Accuracy Gains

An important finding is that the improved model run at *digits*=16 yields a different converged curve than the original model does for modes *m*=1, *n*=2 and higher, although the original model also incorporates double precision, or 16 digits into its answers. Mode *m*=1, *n*=1 of the improved FMM exactly matches the same mode of the original FMM. However, there is a difference in the converged portions of the two models at higher modes, shown by an arrow in Figure 12, that illustrates the variation in model calculations between modes *m*=1, *n*=1 and *m*=1, *n*=2 for the improved FMM, which was not apparent in the original FMM curves displayed in Figure 11. The difference exists because the second term in the improved FMM summation is different

than that incorporated into the original FMM. The explanation is that the improved model performs calculations with symbolic variables, which are not rounded as often as the double precision floating-point values throughout the course of model computations. The values that are set as initial conditions are exact in the improved FMM, whereas they are rounded approximations in the original FMM. Therefore, the improved FMM is more accurate even with *digits*=16—the typical number of digits in the output of double precision calculations in the original FMM.

The difference noted above can be seen more closely in the converged curves of the original and improved FMM in Figure 13, which are separated by 3.4 dB along the *RTS* axis. A 3.4 dB gain in accuracy for that range of *ka* values could be another significant performance gain for the improved FMM. However, this accuracy difference cannot be confirmed until a larger number of modes can be calculated with the improved FMM, after the MATLAB errors are surpassed.

3. Singular Value Decomposition

The VPA caused errors at lower modal combinations in the series solution than expected, and this prevented significant investigation of the effects of SVD on the performance of the improved FMM. The SVD algorithm is more effective at the higher modes, where the numerical contribution to the solution is much smaller and potential error contributions are much higher. Since the improved FMM was significantly limited in the number of modes that it could produce, SVD could not be investigated any further than the useful range of the improved FMM.

When working with increased precision and symbolic numbers, the threshold incorporated into the SVD technique must be sufficiently small to rid the solution of only those values that are beyond the computer's capability to represent them. In this research, symbolic numbers created with *vpa* and *sym* were converted from symbolic form back to double precision after being carried as far into the solution as possible. Since the symbolic variables were carried all the way through the Bessel functions and associated Legendre functions, the only rounding incorporated into the variables before they reached the SVD portion of the model was through the integration process. With

this being the case, there was little instability associated with roundoff error incorporated into the model up until the SVD commands.

Although a larger or smaller number of values may be kept in the solution by the SVD in the improved model, the values in the matrices are more accurate. Fewer rounding operations are incorporated into the symbolic variables of the improved model, so fewer roundoff errors are propagated into the matrices. This means that more of the right values are being kept in the solution by the SVD numerical filtering process while fewer erroneous values are being allowed to stay and affect the results of the scattering simulation.

The SVD algorithm is the most likely cause for the duplication of results for all four values of *digits*, as shown in Figure 10. The SVD may be limiting the number of terms that are included in the solution, even for *digits*=16. Thus, increasing the precision with a higher value of *digits* accomplishes nothing, because the gains afforded by increased precision are negated by the filtering effect of the SVD algorithm.

4. Shortfalls of Extended Precision and Numerical Analysis Techniques

Throughout the course of the research, it was determined that the VPA numbers could not be carried through the entire model from start to finish, because of inherent errors within MATLAB. As mentioned previously, several functions within MATLAB have not been adapted from older source code to handle the capabilities of the Symbolic Math Toolbox. Although the VPA values could not be carried through the entire model, they were carried through the calculation of the Bessel functions and through the associated Legendre functions. If the inputs to these functions from previous sections of the model code were created in double precision (i.e., the precision of ordinary MATLAB numbers), then these inputs would be stored as rounded approximations. Thus, the model would have initiated with erroneous values, and the errors would have only been amplified throughout further calculations. However, the VPA values with extended precision were entered as initial conditions, and the higher precision was carried through the furthest point allowed by the MATLAB software before converting the values back to double precision. The output of the Bessel function, Hankel function, and associated

Legendre function calculations had to be converted to double precision in order to carry out the *max* command—a command that could not be circumvented.

Because of all of the errors encountered with VPA, not enough modal combinations could be calculated to complete a thorough evaluation of the new performance envelope. Even at *digits*=16, impassable errors prevented computations beyond the combination m=1 and n=3 for *AR*=1:1, and results were stopped at m=1 and n=2 for *AR*=5:1. However, the extended precision results that were collected suggest that the performance envelope would be positively affected by the model improvements. The solutions for the second modal combination (i.e., m=1 and n=2) for both *AR*=1:1 and *AR*=5:1 showed that the smooth portions of the *RTS* vs. *ka* curves extended to higher frequencies for the improved model than they did for the original model. This illustrates that destructive interference has less of an effect on the curves of the improved FMM, and fewer modes may be required to achieve a converged solution.

Another limitation of the improved FMM is computational expense. The computational time required by the model was higher than anticipated, and these results are discussed in the following section. This information is presented separately from the aforementioned shortfalls, because computational expense is currently the most significant limitation of the improved FMM.

D. COMPUTATIONAL EXPENSE VERSUS PRECISION ANALYSIS

The FMM requires a great amount of computational expense, especially with increased precision. In order to evaluate the processing time requirements added by increased precision, the model was run several times at different levels of increased precision with all other input variables held constant.

One glaring item of interest in the *RTS* vs. *ka* plots displayed in the previous sections is that the improved FMM was only run for the range $0.5<ka<10$ with a *ka* increment of 0.5, which covers only a portion of the *RTS* vs. *ka* window in each plot (i.e., the improved model curves appear truncated). Fewer values of *ka* could be processed due to the computational time required by the higher values of precision and the symbolic mathematics. The original model was run for the range $0.01<ka<10$ with a *ka* increment of 0.01 providing 50 times the number of data points and while requiring only a small

fraction of the computational time that the improved FMM used. The plots generated with these *ka* ranges were used for visual comparison only, and a separate set of model runs was completed for the computational expense versus precision analysis.

To compare model run times for the original and improved FMM, both models were run for the range $0.5<ka<10$ with a *ka* increment of 0.5. This meant that the same number of data points would be computed by each model. Figure 15 shows the run times for the original FMM, while Figure 16 shows the comparable run times for the improved FMM. Note the lengthy runtimes associated with the improved model that are labeled along the *y*-axis in Figure 16 in comparison to the short runtimes of the original FMM that are shown in Figure 15. The modes that were calculated are annotated on the figures. Mode m=1, n=3 is displayed only for *digits*=32 in Figure 16 because this modal combination encountered errors at the other three precision settings. The times shown in Figure 15 and Figure 16 are the times that were required to calculate all modes up unto and including the indicated mode (i.e., the time shown for m=1, n=2 is the time that was required to calculate both m=1, n=1 and m=1, n=2).

The difference in computational expense between the two models was extremely large. This was mostly due to the differences between the computational expense of floating-point arithmetic and symbolic mathematics. To put this in perspective, mode m=1, n=1 for the original model took only 0.12% of the computational time that the improved model required for the same calculation *with digits*=16. Reciprocally, the improved model took 81,600% of the computational time of the original model at the same mode and precision.

A clear relationship between precision and runtime could not be established for the original versus improved models because of the differences between floating-point and symbolic mathematics. However, it is obvious from the plots that the symbolic mathematics format of the improved FMM with higher precision requires a much greater amount of computational time than standard IEEE double precision format. The relationship between levels of precision for symbolic math calculations alone is linear. This is indicated in Figure 16 by the diagonal line along the bars corresponding to calculations of mode m=1, n=2 at the different values of *digits* in the improved FMM.

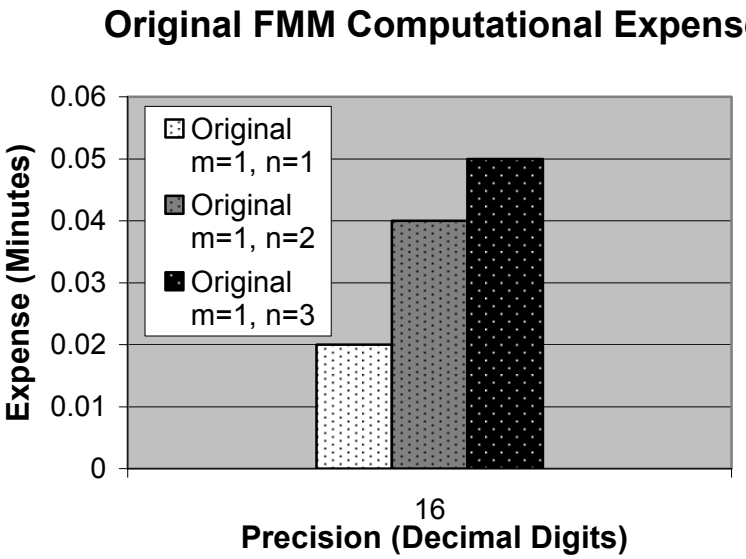


Figure 15. Original FMM computational expense in minutes. The original FMM was run solely at double precision, which corresponds to about 16 decimal digits. Note the relatively short amounts of time required to run the modes, which are annotated along the *y*-axis.

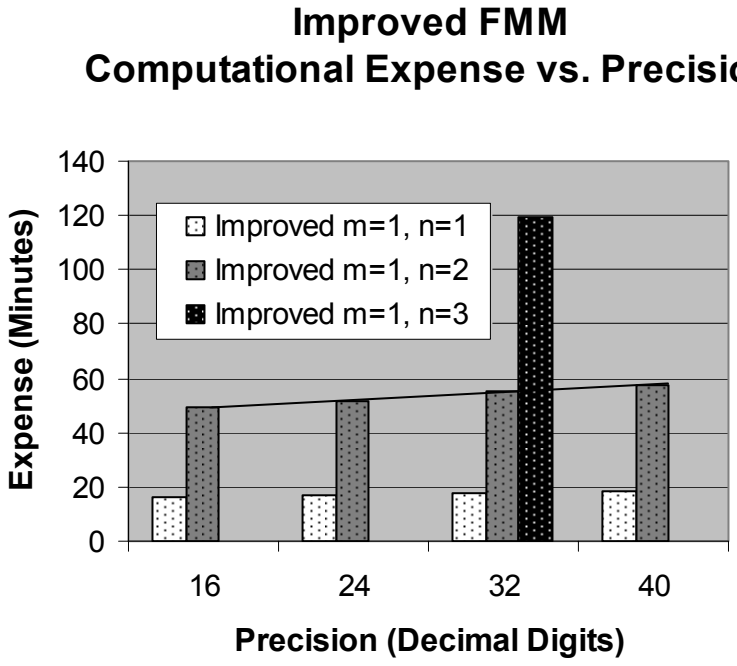


Figure 16. Improved FMM computational expense in minutes versus VPA *digits*. The improved FMM was run at four different levels of precision, which are shown along the *x*-axis here. Note the extremely long computation times.

THIS PAGE INTENTIONALLY LEFT BLANK

V. SUMMARY AND CONCLUSIONS

A. NOTED IMPROVEMENTS IN THE CONVERGED SOLUTION

Variable precision arithmetic afforded a more accurate numerical solution for the improved FMM. A 3.4 dB difference was noted between the converged curves of the original and improved FMM at low frequencies. The converged solution denoted by the positions of the smooth portions of the curves and null spaces in the *RTS* vs. *ka* plots reaches 2.5 units higher into the frequency range for mode m=1, n=3 of the improved FMM because of the executed numerical techniques. Limitations on the improved FMM included unavoidable errors in MATLAB, fewer working modal combinations, and computational expense. While limitations on the number of modes that could be calculated by the improved FMM prevented a complete analysis of the new performance envelope, the expectations of improvements in model performance held true. Another bound has been broken for the FMM, and the improved model offers better scattering predictions.

B. FEASIBILITY FOR IMPLEMENTATION

The FMM requires many improvements before it would be ready for implementation into an operational system. However, the performance envelope discussed by Reeder and Stanton (2004) proves that the model is useful for a large range of modes before increased precision is even considered.

The FMM is intended for use in predicting acoustic scattering by complex scatterer shapes. In light of the computational expense of the improved FMM when describing simple shapes like spheres and prolate spheroids, the computational expense in the case of scatterers with complex or irregular boundaries is currently prohibitive to operational use. This research shows that the Fourier matching method is still far too computationally intensive for current computers to handle as a routine calculation. With the use of an active sonar system, operators need results within seconds to a few short minutes. The FMM developed by Reeder and Stanton (2004) can take up to several hours to run various modes in a single scattering solution, making it impractical for use in today's operational arena. Incorporating increased precision into the formulation and

model source code extends the computational time even further. Higher precision computing only exacerbates the problem of expense, and significant advances in both computer processor speed and storage capacity must be made before higher precision can be utilized in operational systems. However, it is important to maintain the perspective that the FMM is currently a fundamental, physics-based, research level model that is still in developmental stages. The FMM pushes the envelope of technology by demanding faster and more robust computer systems before it can be fully implemented.

C. RECOMMENDATIONS FOR FUTURE WORK

The first four recommendations in this section are provisional solutions that will allow more expedient research of the capabilities of the FMM. The fifth and final recommendation here will bring a major change to the mathematical formulation of the FMM, which may offer significant technical improvements to the future FMM.

1. Implementation of Symbolic Mathematics

The results of symbolic operations are exact, since they use rational arithmetic, so this may be an area to consider further research for the FMM. While the *sym* command introduces even more errors than *vpa* in the current version of the MATLAB software, several of these errors will no longer be encountered once Mathworks resolves compatibility issues with the Symbolic Math Toolbox. The *sym* command works similarly to the *vpa* command, so implementation would be a relatively straightforward extension of this research. Implementation of the *sym* command would be beneficial for the development of the FMM once the issues of MATLAB errors and computational expense are mitigated.

2. Conversion to Fortran

Converting the FMM to run in the Fortran programming language could permit the use of supercomputers for FMM development. Fortran could enable the employment of machines with higher values of machine precision, including computers that are currently capable of running with 128-bit word storage. The results of this research show that increased precision makes a difference in FMM scattering predictions. The full capabilities of the FMM may be realized if it can be run at higher modal combinations with increased precision without running into errors associated with software.

3. Investigation of SVD Thresholding

Future versions of the MATLAB software may allow the errors experienced in this research to be avoided. This would allow the computations of higher modal combinations with the improved FMM. At the higher values of precision afforded by VPA, the threshold for the SVD algorithm should be set to reflect the minimal amount of roundoff error according to the appropriate value of *eps*. This latest version of the FMM incorporates VPA, which allows various settings of precision. Thus, the range of appropriate thresholds for SVD at various levels of computer precision should be investigated to a greater extent. The extent to which SVD is affecting the solution is still unclear. This is the next step in the progression of FMM development, if the improved FMM from this research is intended for future use.

4. Database of FMM Results

An improvised solution to the lengthy runtimes of the FMM that would support implementation may be to tabulate results in a database. A database of FMM results may provide operators with the answers they need in near real-time, rather than having to calculate the lengthy FMM solution from start to finish. If a compilation of results can be generated for use as a look-up table or computer-accessed database, then application of those results to operational problems could possibly be accomplished within a more reasonable amount of time.

5. Incorporation of Spheroidal Wave Functions into the FMM

In 1957, Carson Flammer published a text called *Spheroidal Wave Functions* (Flammer, 1957). This text presents the foundations for the next major revision in the mathematical formulation of the FMM. The purpose of this future improvement is to change the mathematical formulation of the model to match the general scatterer shape more closely. This may yield computational advantages, because the exact solution given by the spheroidal wave functions is closer to the actual scattering phenomena associated with a prolate spheroid than is the approximation from the spherical wave functions. Hence, fewer modal combinations may be required to yield the same result achieved with the spherical wave functions of the current FMM formulation. The resulting model should replicate scattering phenomena for prolate spheroids or elongated irregular bodies

more accurately and possibly with less computational expense. Figure 17 shows a comparison of the Reeder and Stanton (2004) FMM with the exact prolate spheroidal solution.

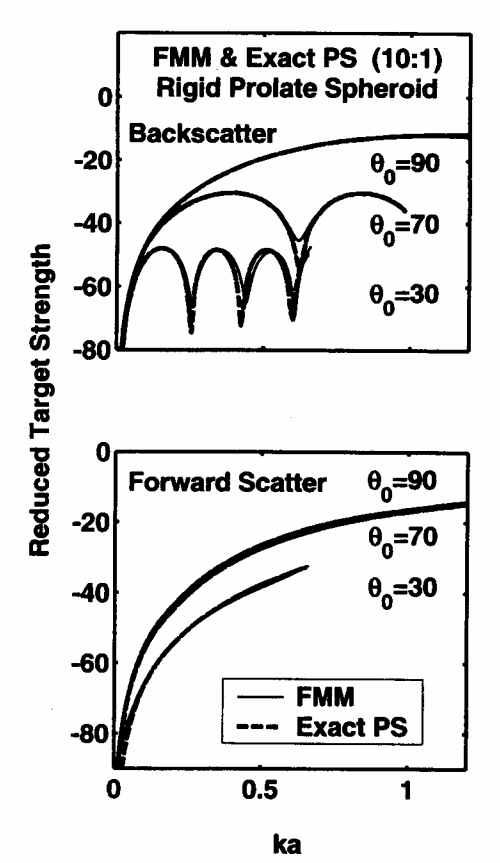


Figure 17. FMM solutions of Reeder and Stanton (2004) compared to exact prolate spheroidal solutions. Many lines coincide because of the similarities in solutions, obscuring the dotted line exact solutions under the solid FMM curves. (From Reeder and Stanton, 2004)

Application of the resulting FMM with incorporated spheroidal wave functions is envisioned for advanced torpedo sonar systems. The new formulation is expected to be more effective in reducing clutter in the sonar picture of a torpedo using active transmissions. Inclusion of spheroidal wave functions would also make the FMM more applicable to detection of elongated scattering bodies, such as submarines, gliders, and autonomous underwater vehicles (AUVs) (Reeder and Stanton, 2005).

D. CONCLUDING REMARKS

The FMM formulation for scattering by irregular, finite-length bodies of revolution developed by Reeder and Stanton (2004) has been adapted for extended precision through the use of the variable-precision arithmetic (VPA) instrument of the MATLAB Symbolic Mathematics Toolbox. The mathematical formulation of the FMM has not changed, but the computer processes by which the solution is calculated have been optimized for increased accuracy and the reduction of roundoff error. The improved FMM results confirm a more accurate converged solution at lower modal combinations, indicating that the performance envelope of the FMM has been improved. Extensions of this research beyond current software limitations may show marked improvements in the model's ability to depict real-world acoustic scattering events.

While implementation of the FMM into operational systems is currently not gainful due to operational time constraints and other limitations, this research shows improvement in the ability of the FMM to represent scattering phenomena. With continued performance gains and functional progress, the FMM will be an integral part of a superior active sonar system.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Abromowitz, M. and Stegun, I.A., *Handbook of Mathematical Functions*, Dover Publications, 1965.

Anderson, V.C. "Sound Scattering from a Fluid Sphere," *Journal of the Acoustic Society of America*, v. 22, pp. 426-431, July 1950.

DiPerna, D.T., and Stanton, T.K., "Sound Scattering by Cylinders of Noncircular Cross Section: A Conformal Mapping Approach," *Journal of the Acoustic Society of America*, v. 96, pp. 3064-3079, August 2004.

Flammer, C., *Spheroidal Wave Functions*, Stanford University Press, 1957.

Francis, D.T., "A Gradient Formulation of the Helmholtz Integral Equation for Acoustic Radiation and Scattering," *Journal of the Acoustic Society of America*, v. 93, pp. 1700-1709, 1993.

Haberman, R., *Elementary Applied Partial Differential Equations*, 3rd ed., Princeton Hall, 1998.

Hackman, R.H., and Todoroff, D.G., "An Application of the Spheroidal-Coordinate-Based Transition Matrix: The Acoustic Scattering from High Aspect Ratio Solids," *Journal of the Acoustic Society of America*, v. 78, pp. 1058-1071, 1985.

Lakhtakia, A., Varadan, V.K., and Varadan, V.V., "Iterative Extended Boundary Condition Method for Scattering by Objects of High Aspect Ratios," *Journal of the Acoustic Society of America*, v. 76, pp. 906-912, 1984.

Mathews, J.H. and Fink, K.D., *Numerical Methods Using MATLAB*, 3rd ed., Princeton Hall, 1999.

The Mathworks, Inc., *Learning Matlab 6.5*, 2002.

The Mathworks, Inc., "Technical Solutions: Solution Number 1-1AG3M," [http://www.mathworks.com/support/solutions/data/1-1AG3M.html?solution=1-1AG3M], August 05.

The Mathworks, Inc., "Help Document for *vpa.m*," MATLAB Version 7.0.1.

Medwin, H. and Clay, C.S., *Fundamentals of Acoustical Oceanography*, Academic Press, 1998.

Ol'shevskii, V.V., *Characteristics of Sea Reverberation*, Consultants Bureau, 1967.

Overton, M.L., *Numerical Computing with IEEE Floating Point Arithmetic*, Society for Industrial and Applied Mathematics, 2001.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P., *Numerical Recipes in Fortran*, 2nd ed., Cambridge University Press, 1992.

Rayleigh, Lord, *The Theory of Sound*, Dover, 1945.

Reeder, D.B. and Stanton, T.K., "Acoustic Scattering by Axisymmetric Finite-length Bodies: An Extension of a Two-Dimensional Conformal Mapping Method," *Journal of the Acoustic Society of America*, v. 116, No. 2, pp. 729-746, August 2004.

Reeder, D.B., Jech, J.M., and Stanton, T.K., "Broadband Acoustic Backscatter and High-resolution Morphology of Fish: Measurement and Modeling," *Journal of the Acoustic Society of America*, v. 116, No. 2, pp. 747-761, August 2004.

Reeder, D.B., and Stanton, T.K., "Physics-Based Detection and Classification Algorithms for Discrete Scatterers and Clutter," Woods Hole Oceanographic Institution Draft White Paper, 2005.

Tobacman, W., "Calculation of Acoustic Wave Scattering by Means of the Helmholtz Integtral Equation," *Journal of the Acoustic Society of America*, v. 76, pp. 599-607, 1984.

Urick, R.J., *Principles of Underwater Sound*, 3rd ed., Peninsula Publishing, 1983.

Varadan, V.K., *et al*., "Computation of Rigid Body Scattering by Prolate Spheroids Using the T-Matrix Approach," *Journal of the Acoustic Society of America*, v. 71, pp. 22-25, 1982.

Waterman, P.C., "New Formulation of Acoustic Scattering," *Journal of the Acoustic Society of America*, v. 45, pp. 1417-1429, 1968.

Yamashita, E., *Analysis Methods for Electromagnetic Wave Problems*, Artech House, 1990.

BIBLIOGRAPHY

Abromowitz, M. and Stegun, I.A., *Handbook of Mathematical Functions*, Dover Publications, 1965.

Anderson, V.C. "Sound Scattering from a Fluid Sphere," *Journal of the Acoustic Society of America*, v. 22, pp. 426-431, July 1950.

Bowman, J.J., Senior, T.B., and Uslenghi, P.L., *Electromagnetic and Acoustic Scattering by Simple Shapes*, Hemisphere Publishing Corporation, 1987.

Boyce, W.E. and DiPrima, R.C., *Elementary Differential Equations and Boundary Value Problems*, 7th ed., John Wiley and Sons, Inc., 2001.

Chapman, S.J., *MATLAB Programming for Engineers*, 2nd ed., Brooks/Cole, 2002.

Crocker, M.J., *Handbook of Acoustics*, John Wiley and Sons, Inc., 1998.

DiPerna, D.T., and Stanton, T.K., "Sound Scattering by Cylinders of Noncircular Cross Section: A Conformal Mapping Approach," *Journal of the Acoustic Society of America*, v. 96, pp. 3064-3079, August 2004.

Flammer, C., *Spheroidal Wave Functions*, Stanford University Press, 1957.

Francis, D.T., "A Gradient Formulation of the Helmholtz Integral Equation for Acoustic Radiation and Scattering," *Journal of the Acoustic Society of America*, v. 93, pp. 1700-1709, 1993.

Haberman, R., *Elementary Applied Partial Differential Equations*, 3rd ed., Princeton Hall, 1998.

Hackman, R.H., and Todoroff, D.G., "An Application of the Spheroidal-Coordinate-Based Transition Matrix: The Acoustic Scattering from High Aspect Ratio Solids," *Journal of the Acoustic Society of America*, v. 78, pp. 1058-1071, 1985.

Jacobs, J.A., *Geomagnetism*, Academic Press, 1987.

Lakhtakia, A., Varadan, V.K., and Varadan, V.V., "Iterative Extended Boundary Condition Method for Scattering by Objects of High Aspect Ratios," *Journal of the Acoustic Society of America*, v. 76, pp. 906-912, 1984.

"The Legendre Polynomials," [http://aemes.mae.ufl.edu/~uhk/Legendre.jpg], August 2005.

Mathews, J.H. and Fink, K.D., *Numerical Methods Using MATLAB*, 3rd ed., Princeton Hall, 1999.

The Mathworks, Inc., *Learning Matlab 6.5*, 2002.

The Mathworks, Inc., "Technical Solutions: Solution Number 1-1AG3M," [http://www.mathworks.com/support/solutions/data/1-1AG3M.html?solution=1-1AG3M], August 2005.

The Mathworks, Inc., "Help Document for *vpa.m*," MATLAB Version 7.0.1.

Medwin, H. and Clay, C.S., *Fundamentals of Acoustical Oceanography*, Academic Press, 1998.

Ol'shevskii, V.V., *Characteristics of Sea Reverberation*, Consultants Bureau, 1967.

Overton, M.L., *Numerical Computing with IEEE Floating Point Arithmetic*, Society for Industrial and Applied Mathematics, 2001.

Pratap, R., *Getting Started with MATLAB*, Oxford University Press, 2002.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P., *Numerical Recipes in Fortran*, 2nd ed., Cambridge University Press, 1992.

Rayleigh, Lord, *The Theory of Sound*, Dover, 1945.

Reeder, D.B. and Stanton, T.K., "Acoustic Scattering by Axisymmetric Finite-length Bodies: An Extension of a Two-Dimensional Conformal Mapping Method," *Journal of the Acoustic Society of America*, v. 116, No. 2, pp. 729-746, August 2004.

Reeder, D.B., Jech, J.M., and Stanton, T.K., "Broadband Acoustic Backscatter and High-resolution Morphology of Fish: Measurement and Modeling," *Journal of the Acoustic Society of America*, v. 116, No. 2, pp. 747-761, August 2004.

Reeder, D.B., and Stanton, T.K., "Physics-Based Detection and Classification Algorithms for Discrete Scatterers and Clutter," Woods Hole Oceanographic Institution Draft White Paper, 2005.

Sommerfield, A., *Partial Differential Equations in Physics*, Academic Press Inc., 1949.

"Spherical Bessel Functions of the First Kind," [http://aemes.mae.ufl.edu/~uhk/sphj.JPG], August 2005.

"Spherical Bessel Functions of the Second Kind," [http://aemes.mae.ufl.edu/~uhk/sphn.JPG], August 2005.

Stanton, T.K. and Chu, D., "On the Acoustic Diffraction by the Edges of Benthic Shells," *Journal of the Acoustic Society of America*, v. 116, No. 1, pp. 239-244, July 2004.

Tobacman, W., "Calculation of Acoustic Wave Scattering by Means of the Helmholtz Integtral Equation," *Journal of the Acoustic Society of America*, v. 76, pp. 599-607, 1984.

Urick, R.J., *Principles of Underwater Sound*, 3rd ed., Peninsula Publishing, 1983.

Varadan, V.K., *et al*., "Computation of Rigid Body Scattering by Prolate Spheroids Using the T-Matrix Approach," *Journal of the Acoustic Society of America*, v. 71, pp. 22-25, 1982.

Waterman, P.C., "New Formulation of Acoustic Scattering," *Journal of the Acoustic Society of America*, v. 45, pp. 1417-1429, 1968.

Wikipedia, "Bessel Function," [http://en.wikipedia.org/wiki/Bessel_function], August 2005.

Wolfram Research(a), "Spherical Bessel Differential Equation," [http://mathworld.wolfram.com/SpericalBesselDifferentialEquation.html], August 2005.

Wolfram Research(b), "Spherical Bessel Function of the First Kind," [http://mathworld.wolfram.com/SphericalBesselFunctionoftheFirstKind.html], August 2005.

Wolfram Research(c), "Spherical Bessel Function of the Second Kind," [http://mathworld.wolfram.com/SphericalBesselFunctionoftheSecondKind.html

Yamashita, E., *Analysis Methods for Electromagnetic Wave Problems*, Artech House, 1990.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Dr. Mary L. Batteen
   Department of Oceanography
   Naval Postgraduate School
   Monterey, California

4. CDR D. Benjamin Reeder, USN
   Department of Oceanography
   Naval Postgraduate School
   Monterey, California

5. Dr. John Colosi
   Department of Oceanography
   Naval Postgraduate School
   Monterey, California

6. LT Matthew K. Henigin, USNR
   Department of Oceanography
   Naval Postgraduate School
   Monterey, California