

**Development and Analysis of Volume Multi-Sphere Method  
Model Generation using Electric Field Fitting**

by

**G. J. Ingram**

B.S., University of Missouri-Columbia 2014

A thesis submitted to the  
Faculty of the Graduate School of the  
University of Colorado in partial fulfillment  
of the requirements for the degree of  
Master of Science  
Department of Aerospace Engineering Sciences  
2017

This thesis entitled:  
Development and Analysis of Volume Multi-Sphere Method Model Generation using Electric  
Field Fitting  
written by G. J. Ingram  
has been approved for the Department of Aerospace Engineering Sciences

---

Prof. Hanspeter Schaub

---

Dr. Daniel Kubitschek

---

Prof. Robert Marshall

Date \_\_\_\_\_

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Ingram, G. J. (M.S., Aerospace Engineering)

Development and Analysis of Volume Multi-Sphere Method Model Generation using Electric Field Fitting

Thesis directed by Prof. Hanspeter Schaub

Electrostatic modeling of spacecraft has wide-reaching applications such as detumbling space debris in the Geosynchronous Earth Orbit regime before docking, servicing and tugging space debris to graveyard orbits, and Lorentz augmented orbits. The viability of electrostatic actuation control applications relies on faster-than-realtime characterization of the electrostatic interaction. The Volume Multi-Sphere Method (VMSM) seeks the optimal placement and radii of a small number of equipotential spheres to accurately model the electrostatic force and torque on a conducting space object. Current VMSM models tuned using force and torque comparisons with commercially available finite element software are subject to the modeled probe size and numerical errors of the software. This work first investigates fitting of VMSM models to Surface-MSM (SMSM) generated electrical field data, removing modeling dependence on probe geometry while significantly increasing performance and speed. A proposed electric field matching cost function is compared to a force and torque cost function, the inclusion of a self-capacitance constraint is explored and 4 degree-of-freedom VMSM models generated using electric field matching are investigated. The resulting *E*-field based VMSM development framework is illustrated on a box-shaped hub with a single solar panel, and convergence properties of select models are qualitatively analyzed. Despite the complex non-symmetric spacecraft geometry, elegantly simple 2-sphere VMSM solutions provide force and torque fits within a few percent.

## Dedication

To my mom for the support. I owe my chance at an education to her, and I hope I can repay her. Without her help, I'd have starved during the time between graduating from Mizzou and coming to CU. She continues to offer help, and I have to refuse it and remind her that I do have an engineering degree, after all, and that I don't need more. To be so willing to sacrifice so much for someone's true benefit isn't such a foreign concept to me, but her willingness to sacrifice for my mere convenience is. But for me to sacrifice as much as possible for her convenience, to try to repay her, seems reasonable. If I happened upon the wealth of the world, I'm not sure handing it to her would be quite enough in thanks.

To my dad for trying to convince me that math was important while I was getting Cs in algebra in high school. At the time I didn't believe him, because the kid getting Cs in algebra *obviously* knows better than the guy that's been around for 50 years. I enjoy that irony, and I'm sure he does too— assuming he's noticed it. Of course he has. And to my dad for looking at me funny when, at 17, I told him I wanted to become a lawyer, and for his consistent attempts to convince me to become an engineer for 20 or so years before they finally stuck. I'm not sure 8 hours a day staring at a computer screen and writing code was what he had in mind, but I'm confident I'll cook something up he'll be able to brag to his friends about.

And to my siblings, my little brother and sister, just for existing. It's nice having you around.

## Acknowledgements

I would like to thank Dr. Hanspeter Schaub for his guidance while the bulk of this technical work was being conducted. I would like to acknowledge Dr. Trevor Bennett and Christine Reilly, who developed a general volume multi-sphere method force and torque matching optimizer that was easily extended for this work in electric field matching. In addition, I would like to thank Philip Chow for his assistance with the finite element analysis software used in this work, and Joseph Hughes for his invaluable help with generating surface multi-sphere models, analytical data and method of images verifications.

## Contents

<b>Chapter</b>	<b></b>
<b>1</b>	<b>Introduction</b> <span style="float: right;"><b>1</b></span>
1.1	Motivation . . . . . 1
1.2	Electrostatic Modeling Techniques . . . . . 4
1.3	Research Scope . . . . . 8
<b>2</b>	<b>Multi-Sphere Method Background</b> <span style="float: right;"><b>10</b></span>
2.1	MSM Concept . . . . . 10
2.2	MSM Formulation . . . . . 10
2.3	MSM Force, Torque and Electric Field Calculations . . . . . 12
<b>3</b>	<b>Electric Field VMSM Model Fitting</b> <span style="float: right;"><b>14</b></span>
3.1	Proposed Workflow . . . . . 14
3.2	SMSM Validation . . . . . 17
3.3	Target Geometry, Data Generation and Optimization Methods . . . . . 18
<b>4</b>	<b>E-Field Fitting Results</b> <span style="float: right;"><b>23</b></span>
4.1	Cost Function Comparison . . . . . 23
4.2	Constrained and Unconstrained Box-and-Panel VMSM Models . . . . . 24
4.3	Reduced-Shell Data Sets with Self-Capacitance Constraint . . . . . 26
4.4	Modeling using General 4 Degree-of-Freedom MSM Spheres . . . . . 29
4.5	Qualitative Analysis of Optimizer Convergence . . . . . 32

<b>5</b>	<b>Concluding Remarks</b>	<b>36</b>
5.1	Future Work . . . . .	36
5.2	Conclusion . . . . .	37
	<b>Bibliography</b>	<b>40</b>
	<b>Appendix</b>	
<b>A</b>	<b>Storm-Time Magnetic Field Modeling</b>	<b>44</b>
A.1	Motivation to Consider Solar Storm Effects . . . . .	44
A.2	Tsyganenko External Magnetic Field Model and Geopack . . . . .	45
A.3	tsygGeoC . . . . .	46
A.4	tsygGeoC Python Module . . . . .	48
A.5	Conclusion . . . . .	51
<b>B</b>	<b>MSMOptimizer3D</b>	<b>52</b>
B.1	Summary of Optimization Program Modifications . . . . .	52
B.2	MSMCall.m . . . . .	53
B.3	Names.input . . . . .	53
B.4	Names.dist . . . . .	54
B.5	Names.const . . . . .	54
B.6	Conclusion . . . . .	54

## Tables

### Table

4.1	Initial and Final States for One-Sphere VMSM Model Optimization, C- Constrained, NC- Non-Constrained . . . . .	24
4.2	Initial and Final States for Two-Sphere VMSM Model Optimization, C- Constrained, NC- Non-Constrained . . . . .	25
4.3	Initial and Final States for Three-Sphere VMSM Model Optimization, C- Con- strained, NC- Non-Constrained . . . . .	25
4.4	Initial and Final States for Three-Sphere 4DOF VMSM Model Optimization . . . . .	30
4.5	Total Function Calls for 4DOF Model Comparison . . . . .	31



## Figures

### Figure

1.1	Illustration of the GLiDeR Concept, Which Uses Inertial Thrusting and the Coulomb Force Arising from Active Charge Control for Active Debris Removal . . . . .	3
1.2	Illustration of Formation Flying using Coulomb Forces Arising from Active Charge Control for Control of Spacecraft Relative Positions . . . . .	3
1.3	Illustration of Electrostatic De-Tumbling Concept, Which Uses Active Charge Control to Reduce Tumbling Angular Velocities to Acceptable Levels for Mechanical Grappling [5] . . . . .	4
1.4	Comparison of Accuracy and Time Expense of Various Electrostatic Modeling Techniques. Of the Selected Methods, the Finite Element Method Provides the Greatest Accuracy with the Largest Time Cost, While Reduced Fidelity Multi-Sphere Method Modeling Provides Reasonable Accuracy and Time Costs . . . . .	7
2.1	Illustration of the MSM Concept– the Red Line Indicates a Vector from the Origin of Some External Control Model Coordinate System to the Origin of the Target Model Coordinate System. Black Lines Indicate Relative Position Vectors of Target Model Spheres with Respect to the Target Model Origin or with Respect to Target Model Spheres. [35] . . . . .	11

3.1	Proposed E-Field Fitting Workflow Comparison with Previous Force and Torque Fitting Method. Box Color Approximately Indicates the Time Cost Required in Each Step . . . . .	16
3.2	Comparison in Force Accuracy Between MOI and SMSM Models of Two Charged Spheres Separated by 5 Meters. The MOI Force Prediction Approaches Truth as the Number of Terms in its Summation Increases. Likewise, the SMSM Model Force Prediction Approaches the MOI Force Prediction as the Number of SMSM Spheres is Increased, Verifying the SMSM Modeling Method. . . . .	18
3.3	SMSM Model of a Cylinder, Color Indicates Charge Distribution on the Cylinder Model in the Absence of an External Electric Field . . . . .	19
3.4	Cylinder Data Distribution and VMSM Model Parameters . . . . .	20
3.5	Box-and-Panel Geometry with a Representative Electric Field Data Point Distribution Generated Using the Golden Spiral Algorithm on Shells Spaced Radially and Centered on the Target Geometry . . . . .	21
4.1	Force and Torque and $E$ -Field Cost Functions for Cylinder Model. The Force and Torque Cost Function is Evaluated Across One Quadrant and Shell of FEM-Generated Force and Torque Predictions. The $E$ -Field Cost Function is Evaluated Across the Same Quadrant and Shell, but Using SMSM-Generated $E$ -Field Predictions	23
4.2	Visualizations of VMSM Box-and-Panel Geometry Models Generated Using the $E$ -Field Fitting Method . . . . .	25
4.3	One, Two and Three Sphere VMSM Model Force and Torque Errors with Respect to SMSM Force and Torque Predictions, (—) Constrained, (- -) Non-Constrained .	26
4.4	Total Matlab Function Calls Required to Generate One, Two and Three-Sphere Box-and-Panel VMSM Models using $E$ -Field Fitting . . . . .	27

4.5	Model Force, Torque and Average (Force and Torque) Errors of Three-Sphere Box-and-Panel VMSM Models Generated Using: (—) 3 Whole Shells, (—) 15 m Shell, (—) 14 m Shell, (—) 12 m Shell . . . . .	28
4.6	<b>fmincon</b> Iterations and Total Model Generation Times Required to Generate Three Sphere VMSM Models Using <i>E</i> -Field Fitting and Shell Reduction for Various SMSM-Generated <i>E</i> -Field Data Sets . . . . .	29
4.7	4DOF Force and Torque Percent Relative Error Comparison to SMSM Force and Torque Predictions for: (—) Three-Sphere Model Three Whole Shells, (—) Three-Sphere Model One Half Shell, (—) Three-Sphere Model One Whole Shell (4DOF) . . . . .	31
4.8	<b>fmincon</b> Optimizer Step Size, Optimality Measure, and Constraint Violation Plots for Various 3 DOF Models. Stopping Criteria for the <b>fmincon</b> Optimization Algorithm Used Depend on Each of the Values Plotted. . . . .	34
4.9	Optimizer Step Size, Optimality Measure and Constraint Violation Plot for a 4 DOF Model . . . . .	35
A.1	Wrapping Procedure used to Make Tsyganenko Models and Geopack Accessible from C and Python. The Underlying Fortran Code is First Wrapped with C Using Mixed Programming. The C Code is then Wrapped Automatically Using the Simple Wrapper Interface Generator to Create an Importable Python Module with the Same Functionality as the Original Fortran Code . . . . .	46
A.2	Colormap of Scalar Difference Between Magnitude of Total Field Vector and Internal Dipole Field Vector in nanoTesla for 10 October 1990 Solar Event. This Figure Clearly Shows the Significant Effects a Solar Storm can have on Earth's Magnetic Field . . . . .	50
A.3	Colormap of Scalar Difference Between Magnitude of Total Field Vector and Internal Field Dipole Field Vector for 10 October 1990 Solar Event as Computed by Tsyganenko. [43] . . . . .	51

## Chapter 1

### Introduction

#### 1.1 Motivation

Electrostatic modeling of spacecraft and space debris is an active area of research with applications in Space Situational Awareness (SSA), Lorentz Augmented Orbits (LAO) and touch-less interaction between spacecraft. Spacecraft and debris in both Low Earth Orbit (LEO) and Geosynchronous Earth Orbit (GEO) can become charged by interactions with ambient plasma to such an extent that they are perturbed by Earth's magnetic field [7, 23]. Although accelerations caused directly by Lorentz forces are small in both LEO and GEO, electromagnetic influence on attitude can cause variation in more prominent attitude-dependent perturbations—such as solar radiation pressure—that lead to significant changes in orbital parameters in a matter of hours [23, 10]. Objects that are particularly affected by this phenomenon are those with high area-to-mass ratios, HAMR objects, which constitute part of the debris population [10, 17]. Improvements in satellite propagation and orbit determination of these debris objects requires electrostatic modeling of some form. Computational efficiency is desirable in both trajectory propagation and orbit determination.

In support of the assertion that HAMR objects are particularly affected by Lorentz forces, natural solar system phenomena have been associated with Lorentz forces induced by planetary magnetic fields. In a 1973 flyby of Jupiter, the Pioneer 10 spacecraft returned the first data suggesting the existence of micron-sized dust particles residing in the Jovian system [13]. During its flyby of Jupiter (within 1 AU,) the Ulysses spacecraft observed sub-micron dust particles streaming from the planet [14]. Further observations by Galileo showed a similar phenomena on its Jupiter ap-

proach within 2 AU of the planet and outside of its magnetosphere [12]. Jupiter's rings and volcanic eruptions on Io have been hypothesized as potential sources of the dust, but the electromagnetic properties of the dust streams and high impact velocities (approximately 200 km/s) suggest that these dust particles are being accelerated and ejected from the Jovian system by Jupiter's magnetic field, which rotates relatively quickly with the planet and applies a small Lorentz force to charged dust particles [13]. Since the charged dust particles have a very small mass, the accelerations experienced by them are significant.

In addition to natural orbital phenomena, when spacecraft or debris charges are controlled artificially using ion or electron beams, both Lorentz and Coulomb forces can be harnessed for useful purposes, such as LAO, active debris removal and touch-less de-tumbling of spacecraft [25, 32, 6]. LAO applications are varied, and include drag compensation in elliptical orbits, inclination control, and rendezvous [25]. In addition, Lorentz forces can be used to augment formation flying reconfiguration control laws to reduce thruster  $\Delta V$  requirements, and for limited propellant-less propulsion [24, 34, 38]. Due to its strong magnetic field and relatively fast rotation, Jupiter has been of particular interest for LAO applications. Trade studies show that both augmented capture and gravity assists are within the realm of possibility— decreasing orbit insertion maneuver costs and opening gravity assist windows, thereby offering greater mission planning flexibility [37, 1].

While LAO exploit the Lorentz force experienced by spacecraft moving quickly through a magnetic field, Coulomb forces between spacecraft and debris can also be utilized. Electrostatic tugging is proposed as a means of active debris removal and for applications in formation flying [32, 16, 15]. The Geosynchronous Large Debris Reorbiter (GLiDeR) is one active debris removal concept that harnesses these electrostatic effects. An illustration of the GLiDeR concept is shown in Figure 1.1. The GLiDeR remotely charges a defunct satellite or piece of space debris with an electron beam, allowing it to interact with the object for the purposes of changing the object's orbit. Orbit raising of 250 km can be accomplished in a little over two months using Coulomb interaction between GLiDeR and a target [30]. It is proposed that Coulomb forces might also be applied for control of relative positions in spacecraft formation flying, illustrated in Figure 1.2 [31, 27].

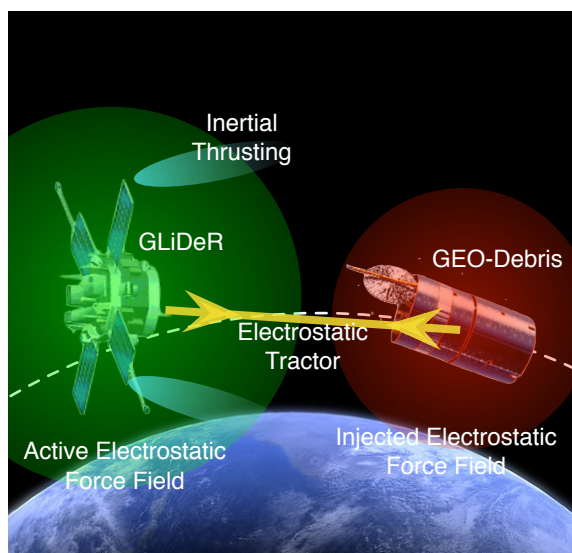


Figure 1.1: Illustration of the GLiDeR Concept, Which Uses Inertial Thrusting and the Coulomb Force Arising from Active Charge Control for Active Debris Removal

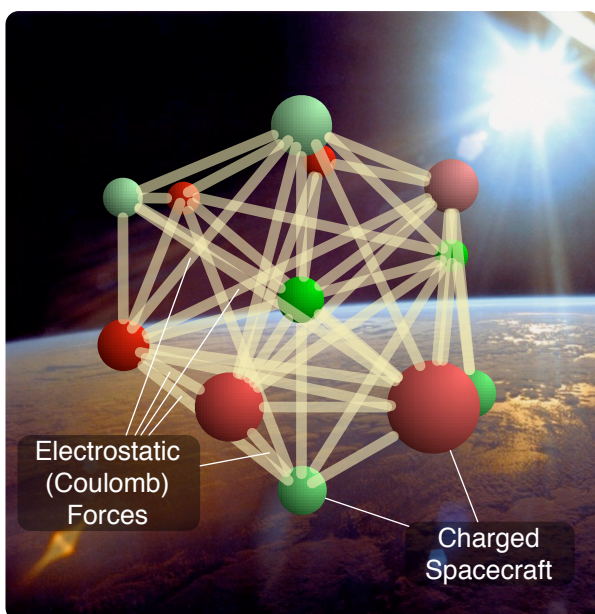


Figure 1.2: Illustration of Formation Flying using Coulomb Forces Arising from Active Charge Control for Control of Spacecraft Relative Positions

Mechanical grappling is sometimes proposed to accomplish similar goals, and also as a method to facilitate on-orbit servicing of spacecraft [45, 46]. When targets are tumbling at even moderate rates mechanical grappling techniques become greatly complicated, and when they are tumbling

rapidly mechanical grappling becomes infeasible [22]. Electrostatic de-tumbling of space objects is currently being researched, and shows promise as a method for reducing debris tumbling rates to within an acceptable magnitude for grappling in servicing missions [6, 4, 3, 29]. An illustration of the electrostatic de-tumble concept is shown in Figure 1.3.

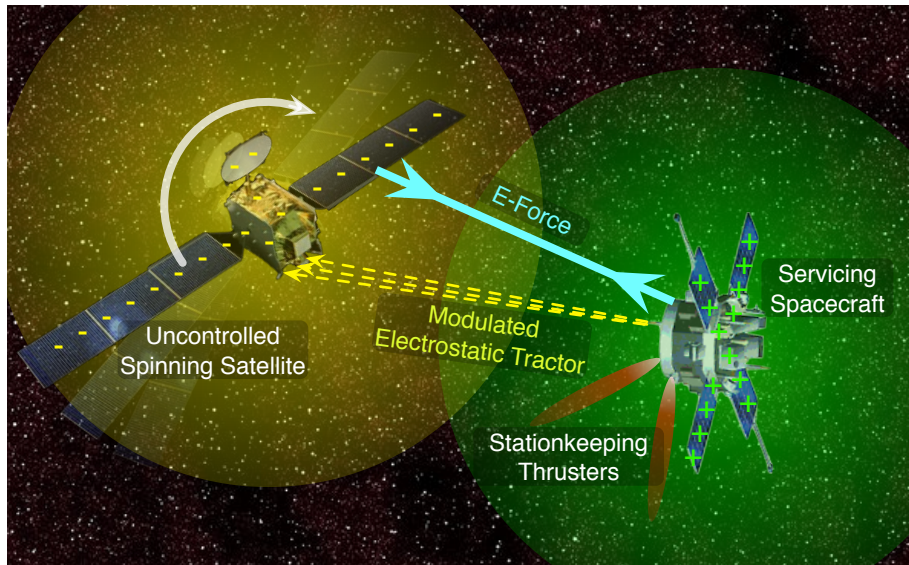


Figure 1.3: Illustration of Electrostatic De-Tumbling Concept, Which Uses Active Charge Control to Reduce Tumbling Angular Velocities to Acceptable Levels for Mechanical Grappling [5]

## 1.2 Electrostatic Modeling Techniques

All of the above applications require knowledge of how charged space objects behave in the presence of external electric and magnetic fields. Various numerical methods have been applied to predict the electrostatic interactions between multiple objects or between an object and a magnetic field. Finite element methods (FEM) provide one means of approximately solving Maxwell's equations over a discretized volume domain. FEM offers the possibility for highly accurate results if the questions it is asked to solve are posed correctly; however, defining a FEM problem requires considerable skill and experience, and at least some a priori knowledge of the behavior of the solution [26]. In addition, the FEM necessarily requires a finite domain to discretize, which is at odds with the fundamental nature of the problem at hand. In order to apply the FEM, the infinite

domain of the problem is reduced to a finite one— at best introducing inaccuracy to the solution and at worst leading to entirely incorrect results [18]. Enormous, although finite, domains must be meshed to analyze spacecraft electrostatic properties, and particular care must be taken to ensure pertinent geometric and solution features are not brushed over when meshing programs are used.

The boundary element method (BEM) is another approximate solution technique. As opposed to FEM’s discretization of a domain, boundary element methods discretize the boundary of the domain. Both the BEM and the FEM are solved as systems of linear equations, and since the BEM generally requires fewer elements than the FEM, the systems associated with the BEM are usually smaller. A particular disadvantage of the BEM in comparison with the FEM is that the coefficient matrices it produces are typically fully populated, whereas coefficient matrices produced by the FEM are banded [18, 26]. The BEM has the advantage that infinite domain problems can be solved.

Many of the aforementioned applications motivate and require faster-than-realtime methods of electrostatic modeling— either for decreased simulation time or for control purposes where long computations can introduce destabilizing phase lag. Although FEM software, such as ANSYS Maxwell 3D©, provide accurate results, calculations of the force and torque between a target and command spacecraft at one relative position and orientation requires time on the order of minutes. A new mesh and solution must be generated for each new relative position and orientation. The duration of this calculation is unacceptable for control purposes, and burdensome even for applications that do not require faster-than-realtime evaluation of forces and torques. The Multi-Sphere Method (MSM) is a lower-fidelity electrostatic modeling technique—bearing some fascinating similarities to the BEM— that can be evaluated quickly enough for faster-than-realtime applications [36, 35]. A comparison of the tradeoff in time consumption and accuracy between various electrostatic approximation techniques is shown in Figure 1.4.

MSM is an elastance-based method for predicting the force and torque on conductors [33]. It is similar to the method of moments in its linear form [11], but differs in that the size and location of the nodes are hand-tuned rather than derived from first principles. Solutions to the MSM problem



are achieved by solving systems of linear equations not unlike those found in FEM and BEM, and, like the BEM, the coefficient matrices encountered in MSM are fully populated. In addition to the computational performance benefits afforded by MSM, its simplicity in comparison to both the BEM and the FEM makes it particularly attractive for electrostatic modeling. Whereas both BEM and FEM require sometimes complicated discretization, and at least qualitative knowledge of the solution's behavior on the problem domain, the MSM requires only knowledge of a minimal number of easily obtainable values, and the discretization method used is remarkably simple.

The number of MSM elements required to obtain an accurate electrostatic model can be very small. Often, only two or three spheres are necessary to characterize a spacecraft. The small size of the resulting matrices affords the possibility of generating force and torque predictions between two spacecraft, or a spacecraft and its magnetic environment, very quickly. In some cases, it is even possible to perform the matrix inversion required by MSM analytically. This property is particularly attractive for applications requiring onboard processing.

Initial development of the MSM technique focuses on the Volume Multi-Sphere Method (VMSM) using a nonlinear solver to automatically size and position perfectly conducting spheres to approximate forces and torques calculated using commercial FEM software [36]. The Surface Multi-Sphere Method (SMSM) addresses shortcomings with the initial VMSM methodology—increasing the accuracy of MSM modeling while simultaneously decreasing the amount of hand-tuning required to generate a good electrostatic model [35]. Recent work in VMSM modeling largely automates the process of generating MSM models using local optimizers to fit force and torque data and eliminates the need for hand-tuning. Because of this, MSM can predict Coulomb forces and torques very quickly at accuracies of 1-2%.

Challenges to current MSM model generation methodologies remain [7]. Current automation of VMSM modeling is accomplished by fitting to FEM generated force and torque data. Not only is the process of generating FEM data time consuming, but it is prone to producing noisy data far from the target model. The effect of noisy data is mitigated by fitting to near-field data and imposing a self-capacitance constraint upon the optimization process. In addition, fitting to force

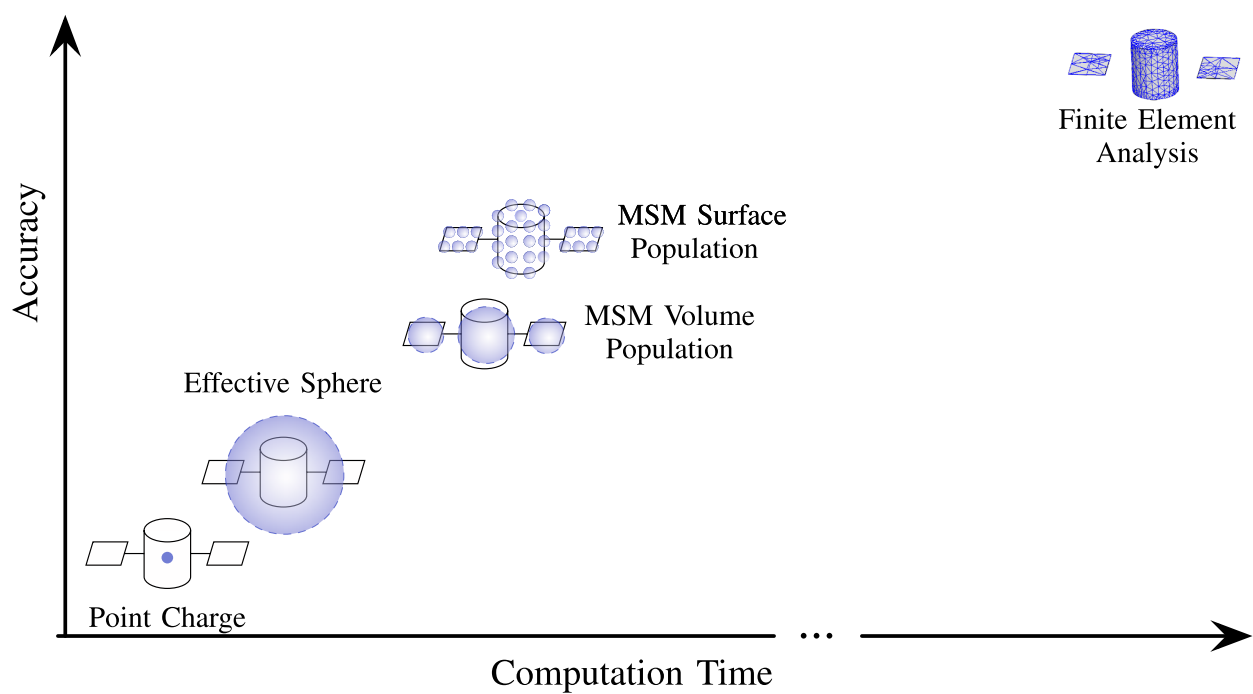


Figure 1.4: Comparison of Accuracy and Time Expense of Various Electrostatic Modeling Techniques. Of the Selected Methods, the Finite Element Method Provides the Greatest Accuracy with the Largest Time Cost, While Reduced Fidelity Multi-Sphere Method Modeling Provides Reasonable Accuracy and Time Costs

and torque data ties VMSM modeling methodologies to a specific probe geometry. Force and torque fitting cost functions must be tailored to deal with data in regions where small absolute numerical error might cause large spikes in relative error.

### 1.3 Research Scope

This work investigates a novel automated method of generating VMSM models using a local optimizer to match the models to the electric field in the vicinity of the target rather than Coulomb forces and torques. Rather than fitting to an electric field generated using the FEM, electric field data is generated using an intermediate SMSM model. Since the electric field data is produced using SMSM, it is much smoother in the far field, which avoids many of the drawbacks of current MSM model generation techniques. A probe geometry is not required when matching to electric field data. *E*-field fitting greatly reduces the time required to generate a VMSM model while maintaining the model accuracy associated with current modeling techniques. The *E*-field matching method can also be demonstrated to allow for creation of 4 degree-of-freedom VMSM models.

A brief review of the Multi-Sphere method is presented. The optimizer cost functions for current VMSM and *E*-field matching model generation methodologies are compared near a solution. Force and torque model accuracies when the *E*-field matching method is applied to a complex asymmetric box-and-panel geometry, with and without a self-capacitance constraint, are investigated. The computational effort of producing one, two and three-sphere VMSM models of the box-and-panel geometry is detailed. The effect of data reduction on model accuracy when a self-capacitance constraint is imposed is shown, and a comparison between the accuracies of reduced-coordinate and 4 degree-of-freedom models is made. Selected *E*-field matched models are used to qualitatively analyze the convergence properties of VMSM modeling using this technique. A summary of the contributions of this work follows:

- (1) A novel *E*-field fitting method of VMSM model generation is presented which avoids pitfalls of conventional force and torque fitting VMSM model generation.

- (2) Cost function properties of  $E$ -field fitting are shown to be favorable for optimization when compared to force and torque cost functions while preserving local minima.
- (3)  $E$ -field fitted VMSM model force and torque predictions when compared to SMSM models of the same geometry are shown to be very accurate, matching within only a few percent at moderate distances from the target geometry when using only two or three model spheres.
- (4) It is shown that imposition of a self-capacitance constraint when using the  $E$ -field fitting method reduces model generation time, ensures good force and torque accuracies at large distances from the target model and can be used to reduce the amount of data required to generate good fits.
- (5)  $E$ -field fitting is shown to be capable of generating full 4 degree-of-freedom VMSM models, providing a stepping stone toward future automation of VMSM model generation.
- (6) Convergence properties of selected models are explored, and it is shown that selecting data at appropriate distances from the target is important for efficient model generation.

## Chapter 2

### Multi-Sphere Method Background

#### 2.1 MSM Concept

MSM provides a reasonably accurate model of a spacecraft's electrostatic properties for applications that require faster-than-realtime results. MSM solves for the force and torque on a conductor by first finding the charge on a number of virtual spheres placed within the body. The charge on the spheres is found by assuming all spheres are equipotential and solving a linear system. MSM divides into VMSM, which uses a small number of spheres placed within the volume of the conductor, and SMSM, which places a large number of spheres on the surface of the conductor. It is more difficult to generate VMSM models because the size and location of the spheres must be found using an optimizer; however, VMSM models are much faster to run once completed. SMSM models are easier to set up because the sphere locations are specified, and the sphere size can be found by matching self capacitance, which is computed from commercial FEM software. Generally, SMSM models show higher accuracy when compared to FEM-generated force and torque data [17]. Although SMSM greatly reduces the effort required to create a MSM model for a given spacecraft geometry and increases the accuracy of the model, it comes at increased computational cost at runtime due to the large number of spheres.

#### 2.2 MSM Formulation

Both the Volume Multi-Sphere Method and Surface Multi-Sphere method represent a conducting object as  $N$  spheres, as shown in Figure 2.1. These methods only differ in the number, size

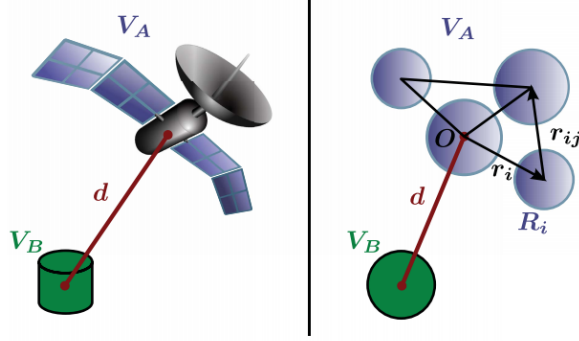


Figure 2.1: Illustration of the MSM Concept– the Red Line Indicates a Vector from the Origin of Some External Control Model Coordinate System to the Origin of the Target Model Coordinate System. Black Lines Indicate Relative Position Vectors of Target Model Spheres with Respect to the Target Model Origin or with Respect to Target Model Spheres [35].

and placement of the spheres used to represent the conductor. The MSM formulation currently assumes that the surface of the target craft or debris is perfectly conducting, which implies that all MSM spheres that constitute a model are equipotential.

Thus, the charge on each sphere in a MSM model, which may include multiple distinct spheres outside of the target model, is only a function of the sizes of the spheres in the complete model and their relative positions. Letting  $N_T$  be the total number of MSM spheres in a MSM model, computed as the sum of the number of spheres in the target model and the number of spheres outside the target model, all MSM spheres can be approximated as point charges. The  $N_T \times 1$  charge matrix,  $\mathbf{q} = [q_1 \ q_2 \ \dots \ q_{N_T}]^T$ , which contains the charge on each sphere of a MSM model, is related to the  $N_T \times 1$  sphere voltage matrix  $\mathbf{V} = [V_1 \ V_2 \ \dots \ V_{N_T}]^T$  through Eq. (2.1) [36]

$$\mathbf{q} = \frac{1}{k_c} [C_M] \mathbf{V} \quad (2.1)$$

where  $k_c$  is the Coulomb constant, and the  $N_T \times N_T$  matrix  $\frac{1}{k_c} [C_M]$  is the Position Dependent Capacitance (PDC) matrix. For a general  $N_T$  sphere MSM model, the PDC matrix is difficult to produce; however, its inverse, the elastance matrix  $[S]$ , contains only functions of the sphere radii

$R_i$  and relative distances between spheres  $r_{i,j}$ .  $[S]$  is formulated as shown in Eq. (2.2) [33]

$$[S] = k_c \begin{bmatrix} \frac{1}{R_1} & \frac{1}{r_{1,2}} & \cdots & \frac{1}{r_{1,N_T}} \\ \frac{1}{r_{2,1}} & \frac{1}{R_2} & \cdots & \frac{1}{r_{2,N_T}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{r_{N_T,1}} & \frac{1}{r_{N_T,2}} & \cdots & \frac{1}{R_{N_T}} \end{bmatrix} \quad (2.2)$$

The charges, which lead to the force and torque, are found by solving the linear system. For an isolated conductor with a small number of spheres this can be done analytically, but in the majority of cases it is done numerically. This computation is expensive, and is the cause of decreased run-time performance when using the SMSM method.

### 2.3 MSM Force, Torque and Electric Field Calculations

The charge set  $\mathbf{q}$  is calculated using Eq. (2.1). The charges,  $q_i$ , are used to calculate the electric field produced by the model, and forces and torques that an object model experiences subject to one or more external point charges. Denoting the position vector of an external point charge as  $\mathbf{r}_k$  and the relative position vector of each MSM model sphere with respect to its center of mass as  $\mathbf{r}_i$ , the force,  $\mathbf{F}$ , and torque,  $\mathbf{T}_O$ , applied to the target model about its center of mass by all external spheres with charges  $q_k$  are given in Eqs. (2.3) and (2.4) [35]

$$\mathbf{F} = -k_c \sum_{k=1}^M q_k \sum_{i=1}^N \frac{q_i}{\|\mathbf{r}_{i,k}\|^3} \mathbf{r}_{i,k} \quad (2.3)$$

$$\mathbf{T}_O = -k_c \sum_{k=1}^M q_k \sum_{i=1}^N \frac{q_i}{\|\mathbf{r}_{i,k}\|^3} \mathbf{r}_i \times \mathbf{r}_{i,k} \quad (2.4)$$

where  $\mathbf{r}_{i,k}$  is the relative position of the  $k^{\text{th}}$  external sphere with respect to the  $i^{\text{th}}$  internal MSM sphere.  $N$  is the total number of spheres in the target model, and  $M$  is the total number of spheres external to the target. Adding arbitrary numbers of external spheres does not add to the complexity of MSM modeling because they can be appended to the charge and elastance matrices; however, doing so will increase computational burden—primarily due growing size of the elastance matrix, which always needs to be inverted. Whole MSM models can be appended when the calculation of

force and torque between two complex geometries is desired. The Lorentz force and torque applied to a MSM model can be calculated using Eqs. (2.5) and (2.6)

$$\mathbf{F}_L = \sum_{i=1}^N q_i (\mathbf{v} + \boldsymbol{\omega} \times \mathbf{r}_i) \times \mathbf{B} \quad (2.5)$$

$$\mathbf{T}_L = \sum_{i=1}^N q_i \mathbf{r}_i \times [(\mathbf{v} + \boldsymbol{\omega} \times \mathbf{r}_i) \times \mathbf{B}] \quad (2.6)$$

Note that the portion of velocity due to the rotational motion of the MSM model will generally be much smaller than the portion due to translational motion and can typically be ignored.

The electric field at any point exterior to the spheres of a MSM model is given by the superposition of the electric fields of each sphere, as shown in Eq. (2.7)

$$\mathbf{E} = k_c \sum_{i=1}^{N_T} \frac{q_i}{\|\mathbf{r}_{i,l}\|^3} \mathbf{r}_{i,l} \quad (2.7)$$

where  $l$  is a point of interest in the space outside of the MSM model spheres, and  $\mathbf{r}_{i,l}$  is the relative position between the  $i$ -th MSM model sphere and  $l$ .

Current VMSM model fitting methods minimize a cost function based upon the difference between the force and torque vectors predicted by a VMSM model and the force and torque vectors generated by commercial FEM software. One such cost function is given by Eq. (2.8) [7]

$$J = \frac{100}{L} \left( \sum_{l=1}^L \frac{\|\mathbf{F}_{\text{VMSM},l} - \mathbf{F}_{\text{FEM},l}\|}{\|\mathbf{F}_{\text{FEM},l}\|} + \frac{\|\mathbf{T}_{\text{VMSM},l} - \mathbf{T}_{\text{FEM},l}\|}{\|\mathbf{T}_{\text{FEM},l}\|} \right) \quad (2.8)$$

To alleviate far-field data noise effects in the optimization process, current methods utilize a self-capacitance constraint. The self-capacitance  $C$  of a MSM model is given by Eq. (2.9) [7]

$$C = \frac{Q}{V} = \frac{\sum_{i=1}^N q_i}{V} = \frac{1}{k_c} \sum_{i=1}^N \sum_{j=1}^N (C_M)_{ij} \quad (2.9)$$

To enforce a self-capacitance constraint, the self-capacitance of the target geometry is calculated using FEM and the MSM model self-capacitance is matched to the FEM result—reducing the optimizer search space by one dimension. The capacitance matching can be accomplished analytically for simple models, or numerically. Matlab's **fmincon** optimizer allows the self-capacitance constraint to be defined in a constraint function and enforced numerically.



## Chapter 3

### Electric Field VMSM Model Fitting

#### 3.1 Proposed Workflow

Using the current VMSM methodology, the sphere positions and radii are varied by an optimizer to best match force and torque data produced by a commercial FEM program [7, 35, 36]. This methodology requires significant hand-tuning of model parameters for a model that fits force and torque data accurately due to noisy FEM data in the far field, dependence on probe size, and convergence properties of the force and torque cost function. Recent work largely streamlines the process of generating VMSM models using ANSYS Maxwell 3D to generate numerical data [7]. The workflow for current VMSM model fitting starts with generating solid models for a target and probe geometry in FEM software and calculating accurate force and torque values between them at many relative positions. Self-capacitance of the target spacecraft is also calculated.

A complete sweep of locations for force and torque matching and validation may take hours and variations in accuracy depend heavily upon how many data points are used. In addition, complicated workarounds that automatically change probe radius are required to get accurate results. Current techniques use a spherical probe which has a radius that is a function of its distance from a target. The dependence on external probe geometry for force and torque calculations is clear in the above force and torque equations. This dependence is included into VMSM model generation when a cost function based on force and torque is used.

The workflow proposed in this paper seeks to address both of these issues. The initial step in modeling is the same — a solid model of the target spacecraft is created. However, a solid model

of external probe geometry is optional, and only used to gather limited force and torque data for verification of a model. A target voltage is prescribed and FEM software is used to calculate the self-capacitance of the target.

A SMSM model of the target geometry is then generated using any appropriate method. One method, appropriate for spherical and cylindrical geometries, is to use a golden spiral algorithm to distribute SMSM spheres on the surface of the geometry [35]. A method appropriate for flat geometries is to use a function like Matlab's **meshgrid** to generate a distribution of sphere positions on the surface of a plane. This distribution can then be translated, rotated and combined with other planar distributions of spheres to form complex SMSM geometries [28]. When combining distributions, care should be taken to eliminate redundant spheres so that none overlap. Root finders like Matlab's **fsolve** can then be used with Eq. (2.9) to determine appropriate sphere radii to match the SMSM model self-capacitance to the self-capacitance of the solid model. Using the prescribed target voltage and Eq. (2.1), the charge on each SMSM sphere is calculated. Using the same equation applied to an  $N$ -sphere VMSM model optimization problem, an optimizer is used to select the optimal positions and radii of the model spheres to minimize the following cost function at sample points  $l$

$$J(R_1, R_2, \dots, R_N, \mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_N) = \sum_{l=1}^L \frac{\|\mathbf{E}_{VMSM,l} - \mathbf{E}_{SMSM,l}\|}{\|\mathbf{E}_{SMSM,l}\|} \quad (3.1)$$

where  $L$  is the total number of electric field sample points.

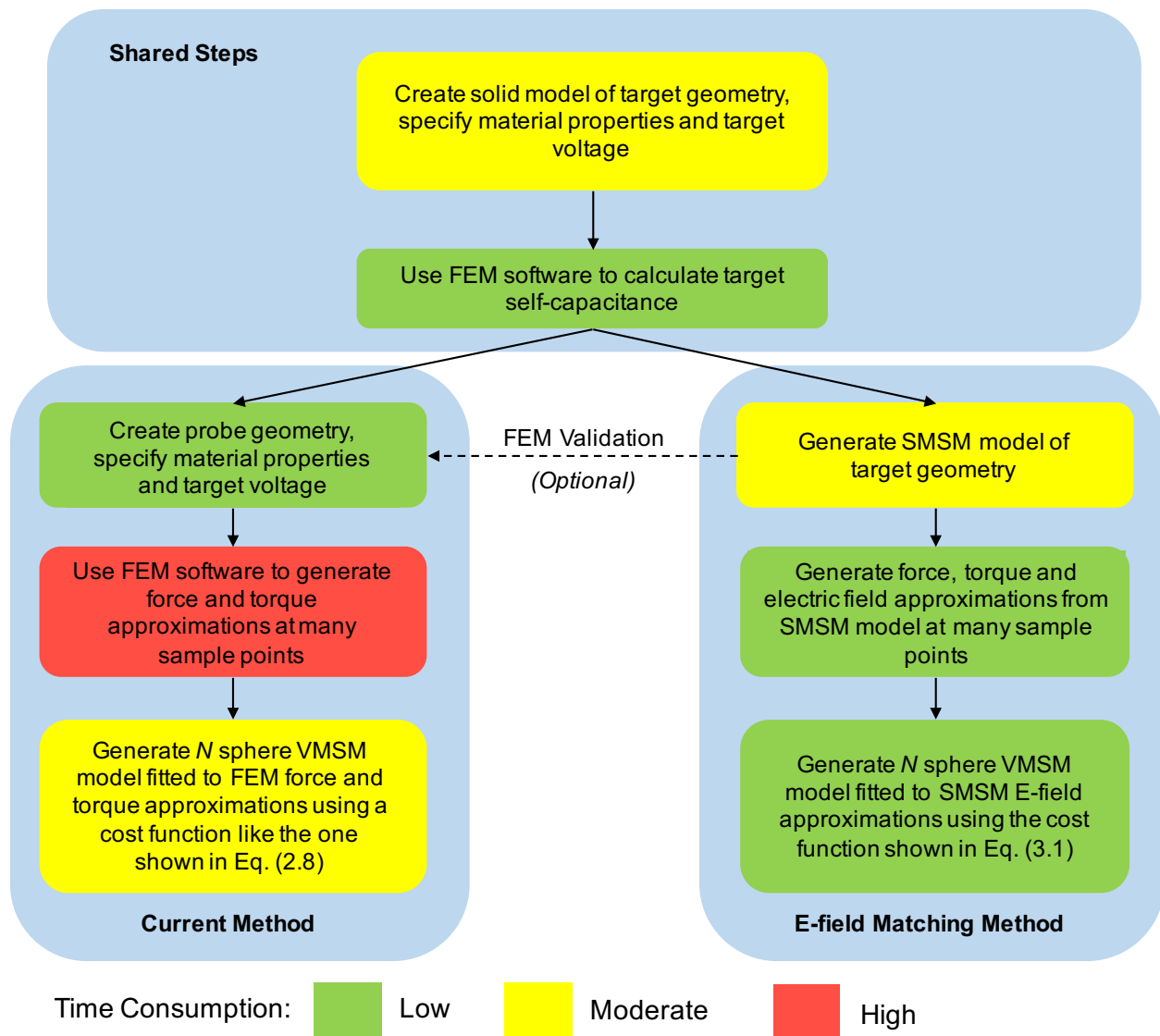


Figure 3.1: Proposed E-Field Fitting Workflow Comparison with Previous Force and Torque Fitting Method. Box Color Approximately Indicates the Time Cost Required in Each Step

Using this cost function results in two major improvements to the prior methodology. The first improvement is the separation of VMSM model generations from any external probe geometry. The second improvement is in the time required to complete all the steps in generating a VMSM model. Previous studies of total VMSM model generation time show that generating moderately accurate FEM force and torque predictions at just 82 sample points requires computation time measured in hours [36]. Moreover, the geometry modeled in Stevenson and Schaub[36] has significant symmetry that can be exploited to reduce the amount of FEM force and torque samples required for a VMSM model fit; however, for a general target geometry such symmetries might not exist and significantly more FEM calculation time and data can be required. Although the proposed  $E$ -field fitting methodology adds the step of creating a SMSM model for a given target geometry and voltage, this process requires significantly less computational effort than computing forces and torques at many relative positions with FEM. Chow[7] discusses fitting to much smaller numbers of force and torque data points; however, using the proposed method, hundreds of data points are calculated and used to create a VMSM model in less than the time it requires to calculate one force and torque calculation using commercial FEM software. A flowchart comparing the previous force and torque optimization procedure and the proposed  $E$ -field matching method is included in Figure 3.1.

### 3.2 SMSM Validation

Before optimizing on a cost function built from a SMSM model, it is prudent to compare the accuracy of SMSM to other methods. The Method Of Images (MOI) provides an analytic infinite series solution to the problem of predicting the force between two spherical conductors [21, 20]. The method consists of placing smaller and smaller image charges within the body of each sphere to balance out the potential on the surfaces. As the number of image charges increases, the solution converges.

SMSM models of spheres are made with a variable number of spheres, and the force between the two bodies is compared to that predicted by MOI. For this case, two spheres with 1 meter radius are separated by 5 meters, and charged to  $\pm 10$  kV. The results are plotted in Fig. 3.2

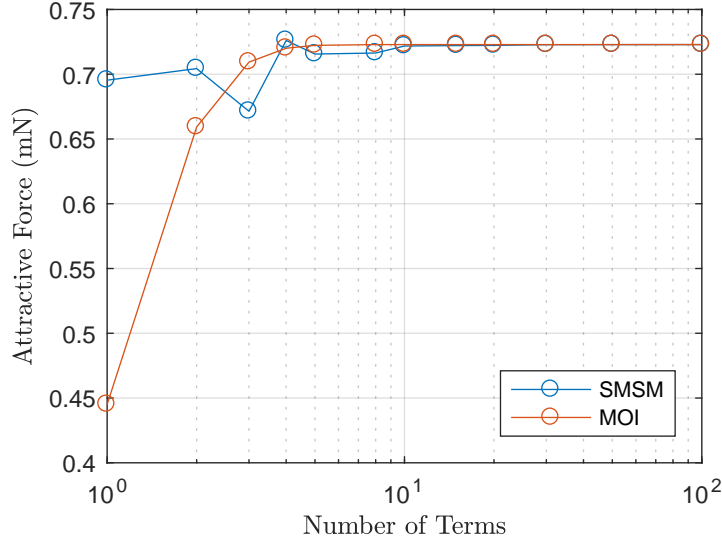


Figure 3.2: Comparison in Force Accuracy Between MOI and SSM Models of Two Charged Spheres Separated by 5 Meters. The MOI Force Prediction Approaches Truth as the Number of Terms in its Summation Increases. Likewise, the SSM Model Force Prediction Approaches the MOI Force Prediction as the Number of SSM Spheres is Increased, Verifying the SSM Modeling Method.

The  $x$ -axis indicates either the number of image charges used in each sphere for the MOI, or the number of individual spheres in each SSM sphere. The MOI converges with 40 image charges, and SSM converges with around 100 spheres. Since both methods converge to the same value, SSM is validated as a method for predicting forces and torques on perfect conductors.

### 3.3 Target Geometry, Data Generation and Optimization Methods

Two target geometries are used to compare the proposed  $E$ -field matching method to the previous force and torque method and to establish the accuracy of models generated using  $E$ -field matching. A two-sphere VMSM model is created for a charged cylinder, with a radius of 0.5 m and a height of 3 m, centered at the origin and oriented along the  $y$ -axis using the  $E$ -field matching method. This geometry is selected for easy comparison to previous work using force and torque matching [7, 36, 35].

Data for the study involving the cylinder geometry are generated at evenly spaced intervals

on 90 degree arcs of circles centered at the origin, lying in the  $x$ - $y$  plane and in the fourth quadrant. The symmetry of the cylinder model requires data in only one quadrant for the optimization process. Force and torque data is generated using Maxwell 3D at the selected sample points. A SMSM model of the cylinder, shown in Figure 3.3, is created to generate  $E$ -field data at the same sample points. Only sample points lying on the arc of a 5 m radius circle are used for optimization, as shown in Figure 3.4(a).

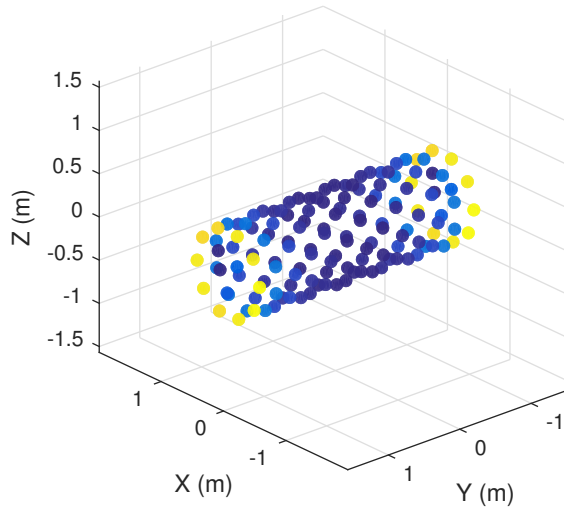
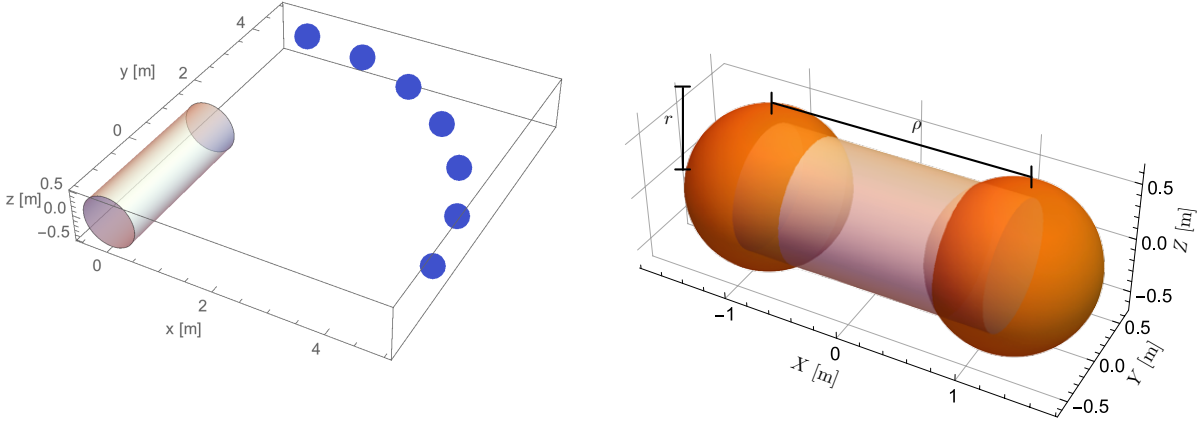


Figure 3.3: SMSM Model of a Cylinder, Color Indicates Charge Distribution on the Cylinder Model in the Absence of an External Electric Field

Two optimization methods are used to generate cylinder VMSM models: a global optimization method using Mathematica’s Differential Evolution algorithm is used to optimize the force and torque cost function of Eq. (2.8), and MATLAB’s **fmincon** local optimizer is used to minimize the  $E$ -field cost function of Eq. (3.1). Self-capacitance for the cylinder is calculated using Maxwell 3D.

Since a two-sphere VMSM model is used to represent the cylinder and the cylinder has symmetry about the  $x$ -axis, the VMSM model can be parameterized by one radius value  $r$ , which is shared by both spheres, and the separation distance,  $\rho$ , between the two spheres. This model is shown in Figure 3.4(b).

Using this model, an exact formula for the VMSM model self-capacitance can be derived, as



(a) Distribution of Electric Field and Force/Torque Sample Points Around a Cylinder Model. This Distribution of Data is Used to Generate VMSM Models of the Cylinder (b) Illustration of a Two-Sphere VMSM Cylinder Model with Model Parameter Annotations [7]

Figure 3.4: Cylinder Data Distribution and VMSM Model Parameters

shown in Eq. (3.2) [7]

$$C = \frac{1}{k_c} \frac{2r\rho}{r + \rho} \quad (3.2)$$

Imposing a self-capacitance constraint, matching the VMSM model self-capacitance to the value calculated by FEM, ensures that Lorentz forces and far-field Coulomb forces are automatically matched. In addition, it reduces the optimization problem to one dimension, because for any  $r$  value there is only one corresponding  $\rho$  value for which the constraint is met. Solving Eq. (3.2) for  $\rho$  yields Eq. (3.3) [7]

$$\rho = \frac{rC_{\text{mod}}}{2r - C_{\text{mod}}} \quad (3.3)$$

where

$$C_{\text{mod}} = k_c C \quad (3.4)$$

Since  $C_{\text{mod}}$  is a scaling of the self-capacitance, it is used to impose the self-capacitance constraint in the following studies because its value is on the order of model parameters. Both the optimal  $r$  and  $\rho$  values are solved for using **fmincon** and the self-capacitance constraint is imposed numerically.

The second target spacecraft geometry that is selected for this study is a box-and-panel satellite modeled as a cube shaped bus with a long, slender panel attached. The bus width, height and depth are 3 m. The panel width is 3 m, depth is 0.2 m and height is 8.5 m. The panel is located on the top of the bus extending in the  $z$ -axis direction with one of its large faces coplanar with the positive  $y$  face of the bus. This model geometry is shown in Figure 3.5. This geometry is selected because it has significant variation from simple geometric shapes like cylinders and represents a realistic target spacecraft. It also shows symmetry that can be used to impose additional constraints on the positions of the VMSM model spheres, and this symmetry is exploited to reduce the amount of data that is required for the optimization procedure.

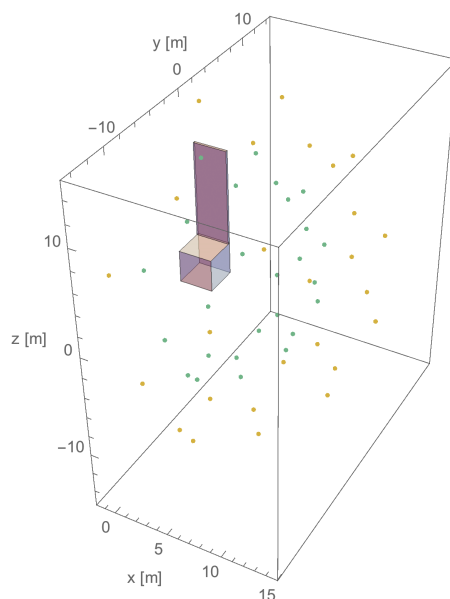


Figure 3.5: Box-and-Panel Geometry with a Representative Electric Field Data Point Distribution Generated Using the Golden Spiral Algorithm on Shells Spaced Radially and Centered on the Target Geometry

A SMSM model of the box-and-panel is generated using MATLAB's **meshgrid** feature, and electric field data is generated using Eq. (2.7), at positions spread radially about the SMSM model. Sample points are generated using the golden spiral algorithm, which places equally spaced points on the surface of a sphere [2]. This radial spread of electric field data produces distributions of shells of data, which are easily selected or neglected, or reduced to hemispheres or octants for



fitting, such as the hemisphere shown in Figure 3.5. Whole spherical shells of data are produced for the box-and-panel geometry; however, due to the symmetry of the model, the complete behavior of the electric field produced by the SMSM model can be captured in one hemisphere of data.

One, two and three-sphere VMSM reduced-coordinate models are created with this data using  $E$ -field matching and MATLAB's **fmincon** optimization algorithm. Various sets of data, including whole spheres, hemispheres and multiple shell sets, are used in the optimization processes. In addition, a three-sphere 4 degree-of-freedom VMSM model is generated using  $E$ -field matching and **fmincon**. Only shells with radii larger than 12 m are used for optimization to avoid overlap between sample points and SMSM spheres, which would invalidate Eq. (2.7). Like the cylinder model optimization process, the box-and-panel self-capacitance is calculated using Maxwell 3D and the self-capacitance constraints on the optimization processes are imposed numerically. Additional inequality constraints are imposed for the two and three-sphere VMSM models to ensure aesthetically pleasing models in which spheres do not overlap. These additional inequality constraints are not required to generate accurate VMSM models.

## Chapter 4

### E-Field Fitting Results

#### 4.1 Cost Function Comparison

The cost function of Eq. (2.8) is compared with the proposed electric field cost function of Eq. (3.1) to compare their characteristics. Both cost functions are used to generate two-sphere VMSM models for a perfectly conducting cylinder charged to 30 kV. Maxwell 3D is used to determine a cylinder self-capacitance value of 106.8345 pF. Noting a potential singularity in Eq. (3.3) when  $r$  is one half  $C_{\text{mod}}$ , the cost function values, shown in Figures 4.1(a) and 4.1(b), along the intersection with the self-capacitance constraint surface are plotted against  $r$  for which  $\rho$  is positive. The  $r$  values minimizing both cost functions are nearly identical, differing by only 0.2%.

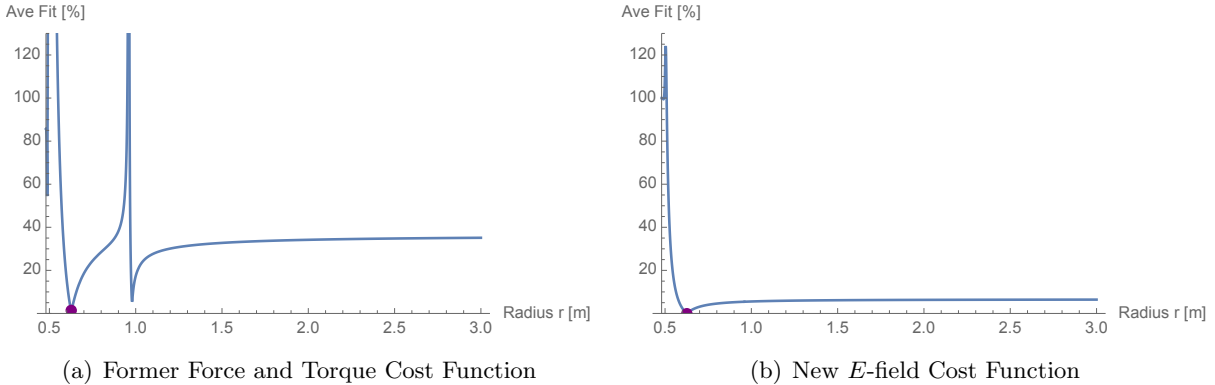


Figure 4.1: Force and Torque and  $E$ -Field Cost Functions for Cylinder Model. The Force and Torque Cost Function is Evaluated Across One Quadrant and Shell of FEM-Generated Force and Torque Predictions. The  $E$ -Field Cost Function is Evaluated Across the Same Quadrant and Shell, but Using SMSM-Generated  $E$ -Field Predictions

Investigating the cost function sensitivities near both minima suggests that force and torque errors

are more sensitive to model parameters than  $E$ -field errors are. The proposed  $E$ -field cost function shows desirable characteristics, particularly for a local optimization algorithm like **fmincon**. It is generally smooth when compared with the force and torque cost function, and apart from a small region near the singularity in  $\rho$ , approaches its minimum unimpeded.

## 4.2 Constrained and Unconstrained Box-and-Panel VMSM Models

One, two and three-sphere box-and-panel VMSM models are generated with and without a self-capacitance constraint in order to determine the usefulness of the constraint when applied to an optimization problem without far-field data noise. Models are fit to data in three whole spherical shells with radii of 15, 20 and 25 m. Symmetry of the box-and-panel geometry about the  $y - z$  plane is used to reduce the  $4N$  degree-of-freedom optimization problem to a  $3N$  degree-of-freedom problem. Imposing a self-capacitance constraint reduces the degrees-of-freedom to  $3N - 1$ . The model is charged to 30 kV. Maxwell 3D is once again used to calculate self-capacitance, and returns a value of 336.14 pF. Optimizer initial conditions are chosen loosely based on the box-and-panel geometry. A visualization the VMSM models of the box-and-panel geometry is included in Figure 4.2.

Tables 4.1-4.3 show the initial and final conditions for the one, two and three-sphere box-and-panel models, with and without the self-capacitance constraint.

Table 4.1: Initial and Final States for One-Sphere VMSM Model Optimization, C- Constrained, NC- Non-Constrained

	$R_1$ [m]	$y_1$ [m]	$z_1$ [m]
Initial	1.000	0.000	0.000
Final C	3.021	0.626	2.914
Final NC	2.951	0.608	2.785

Shell-averaged force and torque errors of the six VMSM models with respect to the SMSM generated data are shown in Figures 4.3(a) and 4.3(b), respectively. As can be seen, the self-capacitance constraint has utility even when far-field noise is not a concern. In particular, when generating a one-sphere model the self-capacitance constraint ensures that force errors far from

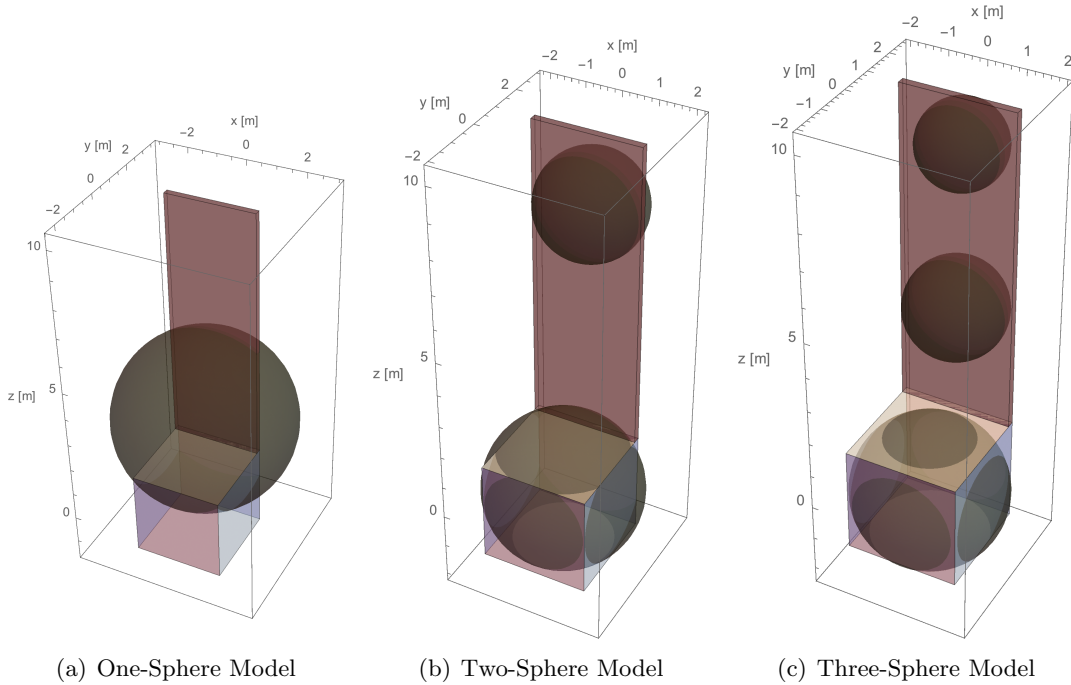


Figure 4.2: Visualizations of VMSM Box-and-Panel Geometry Models Generated Using the  $E$ -Field Fitting Method

Table 4.2: Initial and Final States for Two-Sphere VMSM Model Optimization, C- Constrained, NC- Non-Constrained

	$R_1$ [m]	$R_2$ [m]	$y_1$ [m]	$z_1$ [m]	$y_2$ [m]	$z_2$ [m]
Initial	1.000	1.000	0.000	0.000	1.400	6.000
Final C	2.202	1.458	0.135	0.210	1.596	8.183
Final NC	2.201	1.471	0.134	0.207	1.600	8.177

Table 4.3: Initial and Final States for Three-Sphere VMSM Model Optimization, C- Constrained, NC- Non-Constrained

	$R_1$ [m]	$R_2$ [m]	$R_3$ [m]	$y_1$ [m]	$z_1$ [m]	$y_2$ [m]	$z_2$ [m]	$y_3$ [m]	$z_3$ [m]
Initial	1.000	1.000	1.000	0.000	0.000	1.400	6.000	1.400	9.000
Final C	2.039	1.323	1.120	-0.008	-0.166	1.319	4.584	1.555	8.972
Final NC	2.041	1.322	1.119	-0.007	-0.163	1.321	4.612	1.556	8.974

the target geometry continue to decay as sample point radius is increased. The effect of the self-capacitance constraint is also dramatic for the two-sphere model. The force errors of the two and three-sphere models are nearly identical at 30 m from the target geometry when the constraint is imposed, whereas without it the two-sphere model model force errors remain significantly larger

than the three-sphere model's. It is interesting to note that, as more spheres are added to the VMSM model, self-capacitance is matched automatically as displayed by the three-sphere force and torque errors. Both the constrained and non-constrained model errors overlap, and the optimal model parameters of each are nearly identical.

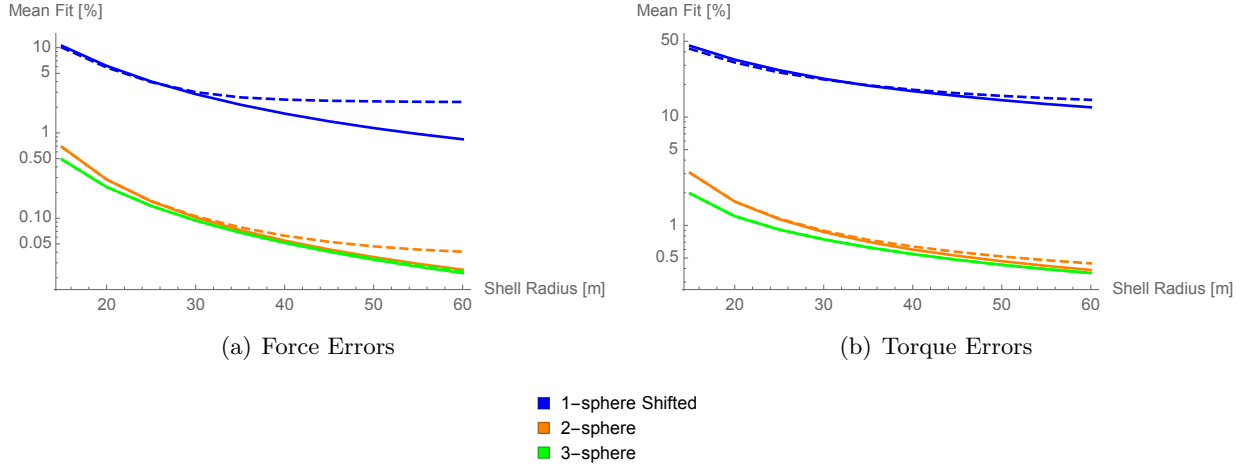


Figure 4.3: One, Two and Three Sphere VMSM Model Force and Torque Errors with Respect to SMSM Force and Torque Predictions, (—) Constrained, (- -) Non-Constrained

Another benefit of imposing the self-capacitance constraint is that it can significantly reduce computational effort. The number of function calls required to generate each of the six models is recorded. The results are shown in Figure 4.4. The results show that imposing the self-capacitance constraint consistently results in increased performance regardless of how many sphere are used to create the VMSM model. This performance increase is due to the reduced dimension of the search space when a self-capacitance constraint is enforced. Given the increased force and torque accuracy and decreased computation effort, the self-capacitance constraint should be enforced even when far-field noise is not a concern.

### 4.3 Reduced-Shell Data Sets with Self-Capacitance Constraint

Since imposing a self-capacitance constraint forces force vectors in the far-field to automatically match, it is desired to investigate whether or not fewer numbers of shells with smaller radii

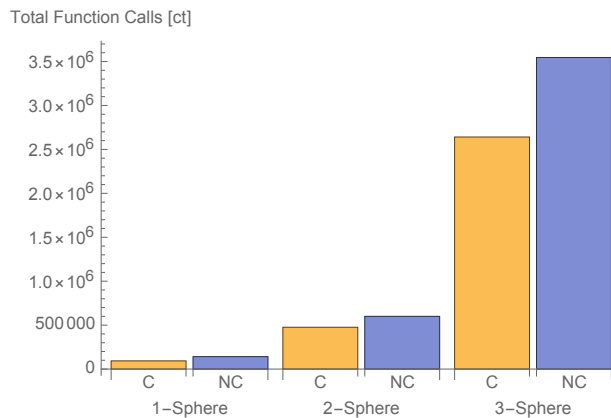


Figure 4.4: Total Matlab Function Calls Required to Generate One, Two and Three-Sphere Box-and-Panel VMSM Models using  $E$ -Field Fitting

can be used to effectively generate VMSM models. This question is interesting for two reasons: smaller data sets will reduce the amount of time required for optimization, and fitting to data closer to the target might increase near-field accuracy without significantly affecting accuracy at larger distances.

A three-sphere VMSM model is used to model the box-and-panel geometry. The VMSM model is fit to single shells at radii of 12, 14 and 15 m. The model voltage is set to 30 kV, and a self-capacitance constraint of 336.14 pF is imposed. The accuracy of these models are compared with each other and to the three-sphere constrained model generated using three whole shells. The resulting shell-averaged force, torque, and average error plots when compared with the SMSM model are shown in Figures 4.5(a)-4.5(c). These results show negligible differences in force and torque accuracy near the target, while the model fitted to the 12 m radius shell shows degraded accuracy far from the target. Accuracies of models fit to three whole shells and to 14 m and 15 m shell radii are nearly identical. The practical impact of the degraded accuracy for the model fitted to the 12 m shell is very small; however, for high accuracy at large distances from a target, a model should be fitted to at least one shell with a larger radius.

The performance impact of reducing the number of shells is analyzed by recording the number of **fmincon** iterations and total time required to generate each model. The results are shown in

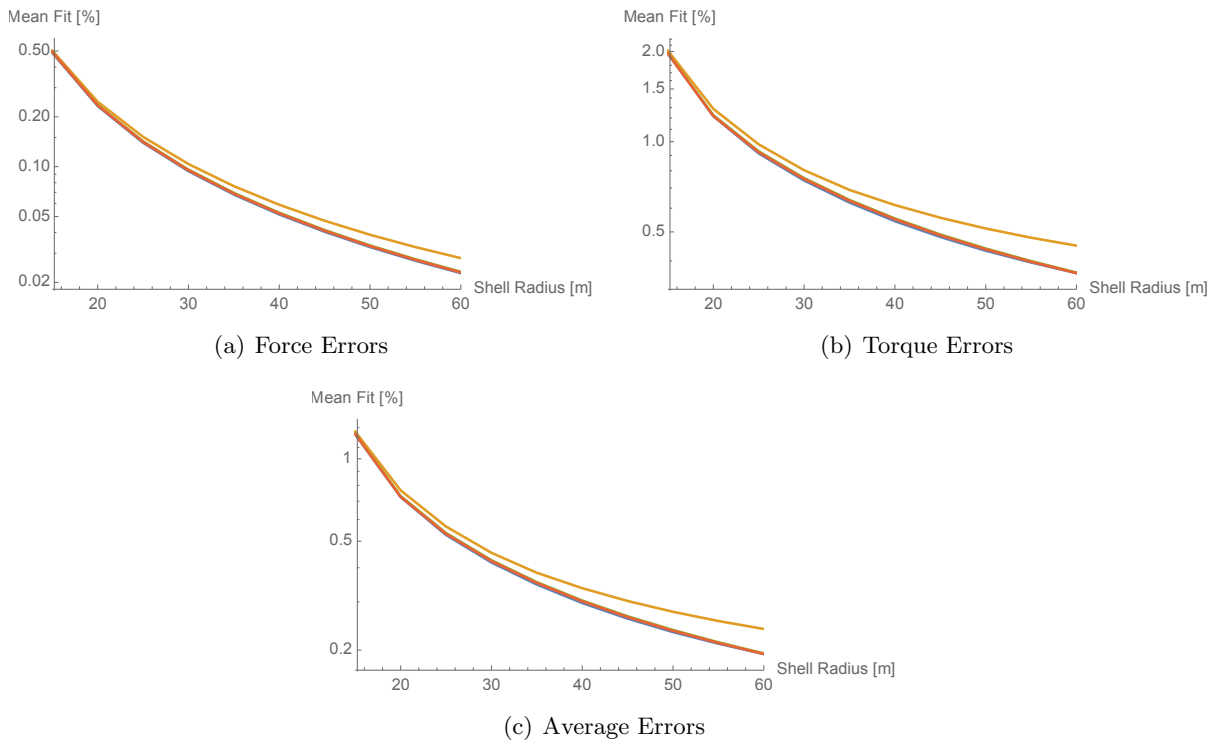


Figure 4.5: Model Force, Torque and Average (Force and Torque) Errors of Three-Sphere Box-and-Panel VMSM Models Generated Using: (—) 3 Whole Shells, (—) 15 m Shell, (—) 14 m Shell, (—) 12 m Shell

Figures 4.6(a) and 4.6(b).

These results show that there is not necessarily a positive correlation between decreased numbers of data points and increased performance. Fitting to data points very close to the target geometry increases the number of iterations required to generate a model. In addition, the time required to generate a model is not solely dependent upon the number of data points. While it takes more time to generate a model using three whole shells of data when compared to one whole shell at a radius of 15 m, it takes approximately the same amount of time to generate a model using one whole shell with a radius of 12 m even though the number of **fmincon** iterations is 12% less. The competing interests of maintaining a self-capacitance constraint while fitting to electric field data near the target— which is far more variable than the same data at larger radii— is likely to blame for this phenomenon. Due to the increased error and decreased performance associated with fitting to shells very close to the target, it is recommended to fit models using shells at intermediate ranges.

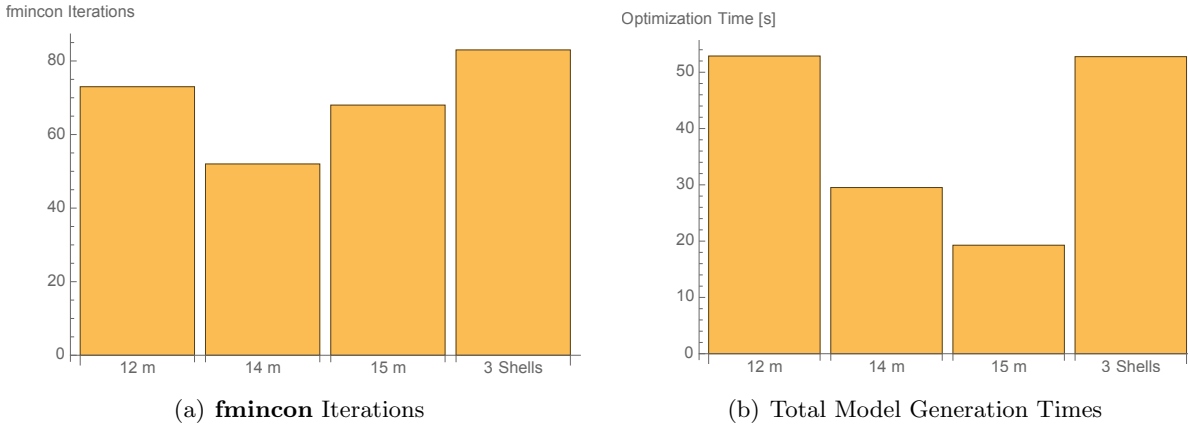


Figure 4.6: **fmincon** Iterations and Total Model Generation Times Required to Generate Three Sphere VMSM Models Using  $E$ -Field Fitting and Shell Reduction for Various SMSM-Generated  $E$ -Field Data Sets

Doing so aids convergence of the optimizer, and the total time required to generate a model is decreased either because the number of iterations to convergence is decreased or because the total time required for one iteration of the optimizer is decreased, or both.

#### 4.4 Modeling using General 4 Degree-of-Freedom MSM Spheres

Previous work focuses on simple geometries that have high levels of symmetry, like the cylinder model. For geometries like these, the number of MSM optimization parameters can be reduced significantly by specifying, for example, that the cylinder MSM spheres must lie on the symmetry axis. As geometric complexity increases, so does the effort required to find symmetries to exploit and analytically parameterize them. Using general MSM 4 degree-of-freedom (4DOF) modeling allows each MSM sphere to have 3 location and one radius degree of freedom. This avoids the need to find symmetry and allows for optimization in a  $4N$  dimension search space. Each parameter associated with a VMSM model sphere is found through optimization, and the up-front human-involvement cost of generating a model is reduced.

A three-sphere VMSM model of the box-and-panel geometry is generated using data in three whole shells with radii of 15, 20 and 25 m. A self-capacitance constraint is enforced using a value of 336.14 pF. The model voltage is set to 30 kV. The initial and final conditions for the 4DOF fitting



Table 4.4: Initial and Final States for Three-Sphere 4DOF VMSM Model Optimization

	Initial	Final
$R_1$ [m]	1.000	2.051
$R_2$ [m]	1.000	1.299
$R_3$ [m]	1.000	1.106
$x_1$ [m]	0.000	-0.004
$y_1$ [m]	0.000	-0.004
$z_1$ [m]	0.000	-0.136
$x_2$ [m]	0.000	0.038
$y_2$ [m]	1.400	1.385
$z_2$ [m]	6.000	4.788
$x_3$ [m]	0.000	-0.028
$y_3$ [m]	1.400	1.534
$z_3$ [m]	9.000	8.993

problem are shown in Table 4.4.

The force and torque accuracy of the 4DOF model with respect to the box-and-panel SMSM model is compared to three-sphere 3DOF models using three whole shells and one half shell at 15 m in Figures 4.7(a) and 4.7(b). The model with the least accuracy is the 3DOF model fitted to one half shell of data. The 4DOF model's accuracy resembles the 3DOF model fitted to three whole shells. Since the optimizer knows nothing about the symmetry of the box-and-panel model, a whole shell of data must be used to generate an accurate model when 4DOF is used. Total **fmincon** iterations for each of the models shown in Figure 4.7 are shown in Table 4.5.

Surprisingly, the 4DOF modeling method yields a converged result using only 58% of the iterations required for a 3DOF model using a half shell of data with the same radius. These results reiterate that the number of iterations required for convergence upon a useful model is far more dependent on using data at intermediate radii than on the number of data points used. However, when considering time required to generate a model, the 3DOF 1 half shell model is by far the most efficient to generate, requiring only about a third as much time as the three whole shell models. An interesting result is that the 4DOF model is generated in approximately the same amount of time as the 3DOF 3 whole shell model. These results show that 4DOF modeling using  $E$ -field fitting is promising from both an accuracy and performance perspective, and a 4DOF can be generated in

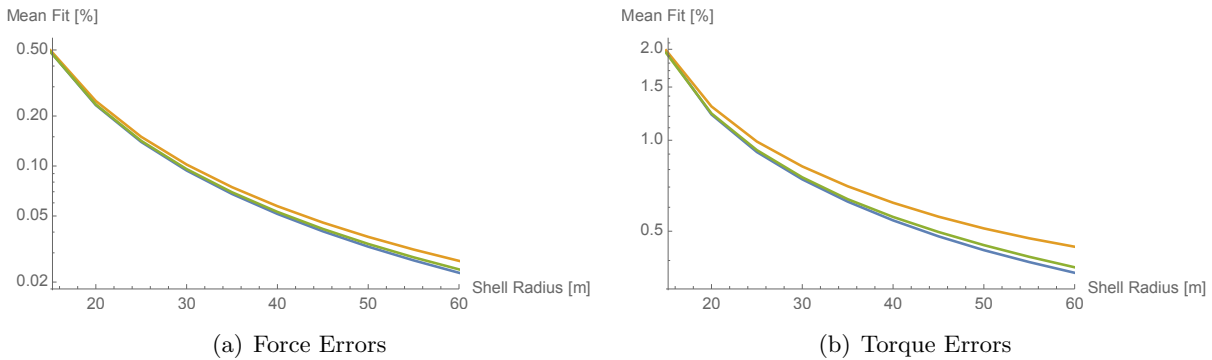


Figure 4.7: 4DOF Force and Torque Percent Relative Error Comparison to SMSM Force and Torque Predictions for: (—) Three-Sphere Model Three Whole Shells, (—) Three-Sphere Model One Half Shell, (—) Three-Sphere Model One Whole Shell (4DOF)

Table 4.5: Total Function Calls for 4DOF Model Comparison

Model	Total <b>fmincon</b> Iterations
4DOF 3 Whole Shells	77
3DOF 3 Whole Shells	83
3DOF 1 Half Shell	132
3DOF 1 Whole Shell	68

approximately the same amount of time as a 3DOF model using the same number of data points at the same shell radii.

#### 4.5 Qualitative Analysis of Optimizer Convergence

Previous **fmincon** iteration results warrant further investigation. **fmincon** attempts to minimize a cost function subject to nonlinear equality and inequality constraints. A solution is found using the **fmincon** algorithm when the step size is sufficiently small, constraints are met to within a sufficient tolerance, and a measure of optimality is met to within a sufficient tolerance. The measure of optimality used by **fmincon** for constrained problems is a measure analogous to the common unconstrained optimality condition that the gradient of the cost function must be sufficiently close to zero for a solution to be considered optimal. The constrained optimality measure varies in that constraints are taken into account explicitly. That the optimality measure be near zero is a necessary condition for a solution to be optimal, but not sufficient to guarantee optimality.

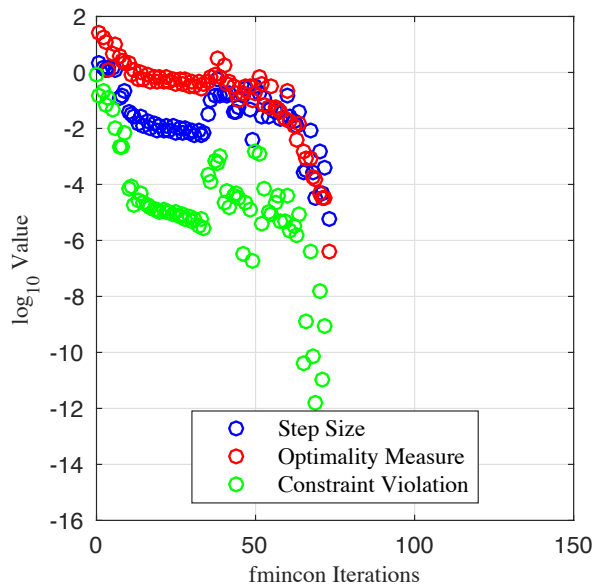
Each of the measures are plotted against **fmincon** iterations for 3DOF models generated using 1 whole shell at 12 m, 14 m and 15 m, and a half shell at 15 m. The same plot is made for a 4DOF model using one whole shell of data at 15 m. Figures 4.8(a)-4.8(d) show the results for various 3 DOF models, and Figure 4.9 show the same results for a 4 DOF model.

Of the three performance measures, the optimality measure is by far the most interesting. If the dimension of the optimizer search space is  $N$  and a self-capacitance constraint is imposed, the optimality measure reveals information about the shape of the  $N - 1$  dimensional surface on which a solution must lie. This information far more important for qualitative analysis of convergence than either the constraint violation or step size.

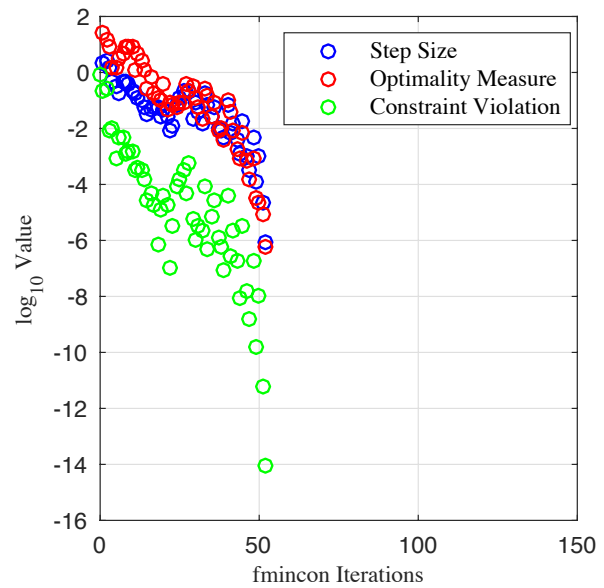
Figure 4.8(a) reveals why convergence is hindered by using a shell small shell radius. The optimization algorithm encounters as 8 dimensional valley that slowly flattens and follows it for approximately 25% of the total optimizer iterations before passing into a region in which the slope of the cost function increases or decreases sharply. It encounters another region where the slope

changes sharply before settling into an optimum. Figures 4.8(b) and 4.8(c) show that the optimizer encounters no such valley region, and instead follows a relatively well-defined path toward the the optimum– explaining their quicker model generation times. Similar behavior is reflected in Figure 4.9.

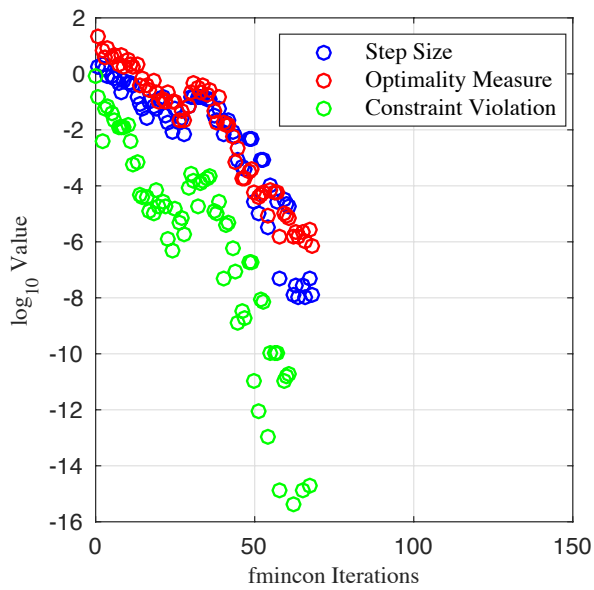
Figure 4.8(d), which displays the optimizer outputs for the 3 DOF half-shell case, shows starkly different behavior. While the constraint violation rapidly approaches machine precision, the optimality measure never sinks beneath  $1E-2$ . It reaches a nearly constant value, suggesting that the optimizer has approached a reasonably flat region with a nearly constant directional derivative in every direction the optimizer steps. Since only the magnitude of the step size is reported, it is unclear whether or not the optimizer is moving in one direction, although this possibility seems likely. What is most concerning is that the final optimality measure in this case is not particularly small– which indicates that the final solution’s optimality is questionable. While the final measure of performance for any MSM model is how well it fit forces and torques, and the 3 DOF half-shell model has been shown to perform well in that sense, it is far less optimal than the models generated using whole shells. Due to decreased performance and optimality when using a half shell of data, it is suggested that only whole shells of data be used.



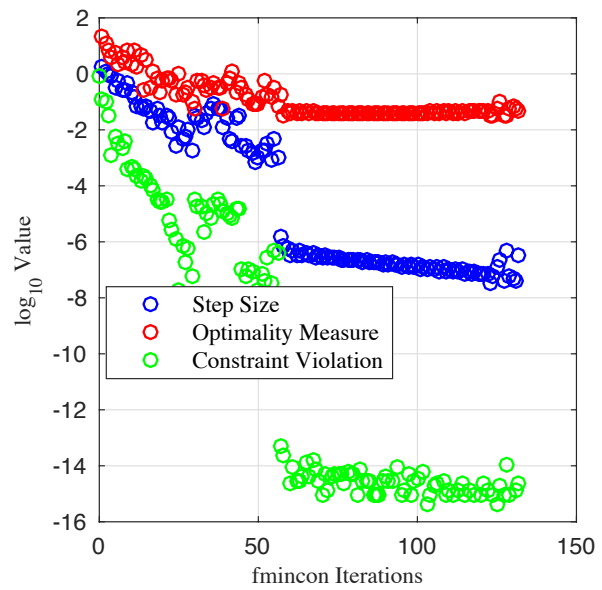
(a) 3 DOF 12 m Shell



(b) 3 DOF 14 m Shell



(c) 3 DOF 15 m Shell



(d) 3 DOF Half Shell (15 m)

Figure 4.8: **fmincon** Optimizer Step Size, Optimality Measure, and Constraint Violation Plots for Various 3 DOF Models. Stopping Criteria for the **fmincon** Optimization Algorithm Used Depend on Each of the Values Plotted.

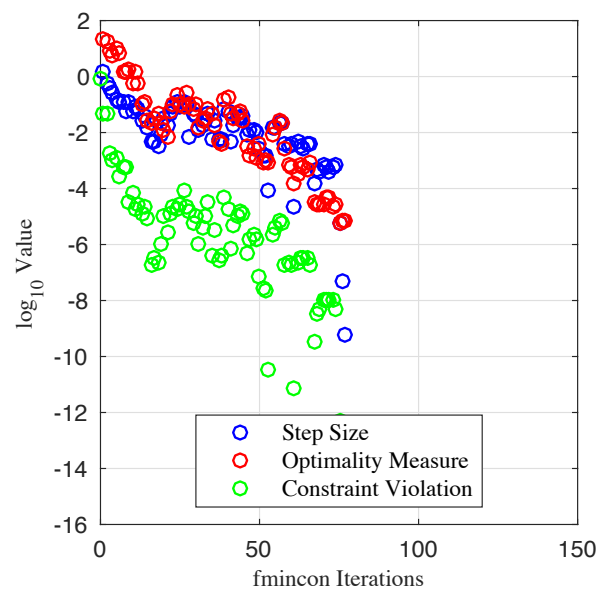


Figure 4.9: Optimizer Step Size, Optimality Measure and Constraint Violation Plot for a 4 DOF Model

## Chapter 5

### Concluding Remarks

#### 5.1 Future Work

The ultimate test of the  $E$ -field fitting method is a comparison between the generated VMSM model forces and torques and numerically generated force and torque data from FEM software. SMSM models produce very accurate force and torque data when compared to Maxwell 3D, and it is promising that the  $E$ -field fitting technique produces VMSM models that not only match a SMSM model's electric field to high accuracy, but also match SMSM predicted forces and torques. Nevertheless, future research must be conducted to verify that VMSM models generated using electric field fitting do match FEM data to high accuracy.

The performance of the  $E$ -field matching method suggests that the results of the cost function analysis can be extrapolated to higher dimension search spaces; however, quantitative research into the convergence properties of the  $E$ -field fitting technique should be carried out. Although a self-capacitance constraint is imposed in this work, other appropriate constraints might be found to further reduce the dimension of the search space and aid performance. Research into the convergence properties of the  $E$ -field matching method might lead the way to fully autonomous model generation using 4DOF VMSM models.

The selected target geometry represents a particular target of interest that might be found at GEO. The box-and-panel target geometry is not representative of all objects of interest, and VMSM modeling of more complicated geometries should be investigated. These geometries should be selected to represent a healthy cross section of potential targets at GEO. An assumption is made

in this work that the target is a perfect conductor, but many objects of interest have dielectric properties. Extension of this work to model dielectrics is a worthwhile pursuit.

## 5.2 Conclusion

This work is focused on developing an alternative VMSM model fitting technique that addresses shortcoming with fitting models to forces and torques directly.  $E$ -field fitting requires no probe geometry, which is a concern when using force and torque fitting techniques. It significantly reduces the amount of time necessary to generate a model. Furthermore, it requires minimal FEM to generate a model—greatly reducing the possibility of user error and the amount of FEM expertise required.

It is shown that using  $E$ -field fitting reduces cost function complexity, and enhances cost function properties. The smooth quality of the  $E$ -field fitting cost function is a great aid when using local optimizers. In addition, solutions generated using  $E$ -field fitting are shown to agree closely with those generated using force and torque fitting. Accurate one, two and three sphere VMSM models of a complex geometry are shown to be easily created using the  $E$ -field fitting technique. Analysis shows that, although the self-capacitance constraint was originally imposed to counter noisy far-field FEM force and torque data, it has practical performance benefits when imposed using smooth  $E$ -field data. It is also shown that as the number of VMSM model spheres is increased, self-capacitance is automatically matched— an intuitive result supporting the validity of fitting to  $E$ -fields.

The self-capacitance constraint can be exploited to reduce the number of data points while maintaining accuracy; however, performance benefits are lost when the self-capacitance constraint and variable fit data are at odds with each other. In addition, model accuracy at large distances is degraded when fitting to shells very close to the target geometry, while model accuracy near the target is not improved significantly. For this reason, it is suggested that models be fit to data at intermediate ranges, which aids in optimizer convergence and improves model accuracy at large distances. It is shown that an “intermediate” range for the complex box-and-panel geometry is a



shell with a radius of 14 or 15 m.

The  $E$ -field fitting method can be used to generate full 4 DOF models. This is beneficial because it reduces the amount of up-front human cost in determining and hard-coding model symmetries, and 4 DOF fitting offers a path toward full automation of VMSM modeling. 4 DOF models are shown to be nearly as accurate as 3 DOF constrained models, and incur no performance penalty in the optimization process because optimization performance is more strongly influenced by the number of data points and agreement between constraints and  $E$ -field data than by the size of the search space.

Qualitative analysis of the convergence properties of this method reiterate that optimizer performance is heavily influenced by the complexity of the cost function and number of data points used. The number of data points is correlated with the time required by each optimizer iteration, and the shape of the cost function is correlated with the total number of iterations required to convergence. Along a constrained path, it is shown that the cost function associated with shell radii near the box-and-panel target has features detrimental to quick convergence. The opposite is shown for shells with intermediate radii. The properties of cost functions and convergence of 3 DOF models are shown to hold true for a 4 DOF model. This analysis largely explains why each model considered requires a particular number of iterations.

Electrostatic modeling has many practical applications, particularly for SSA where debris objects can have large area-to-mass ratios. Electrostatics can be used for tugging and de-tumbling of spacecraft, and formation flying. Lorentz Augmented Orbits can be devised for practical purposes. Although accurate electrostatic modeling techniques, such as the FEM and the BEM, exist, they are too computationally intensive to use for applications requiring real-time evaluation of forces and torques. MSM provides a reasonably accurate, lightweight alternative for these applications. VMSM is a simple and elegant approach to electrostatic modeling that requires far less expertise than more accurate methods to implement, and evaluates forces and torques rapidly. It is shown that VMSM models agree extraordinarily well at intermediate ranges with their more accurate SMSM counterparts. Although challenges to VMSM modeling still exist,  $E$ -field matching promises

accurate VMSM modeling with sharply reduced model generation times, decreased human cost, and decreased complexity when compared to force and torque fitting methods.

## Bibliography

- [1] Justin A. Atchison and Mason A. Peck. Lorentz-augmented jovian orbit insertion. Journal of Guidance, Control, and Dynamics, 32(2):418–423, 2009.
- [2] Robert Bauer. Distribution of points on a sphere with application to star catalogs. Journal of Guidance, Control, and Dynamics, 23(1):130–137, 2000.
- [3] Trevor Bennett and Hanspeter Schaub. Touchless electrostatic three-dimensional detumbling of large axi-symmetric debris. Journal of Astronautical Sciences.
- [4] Trevor Bennett and Hanspeter Schaub. Touchless electrostatic detumbling while tugging large axi-symmetric geo debris. In AAS/AIAA Space Flight Mechanics Meeting, Williamsburg, VA, Jan. 11–15 2015. Paper AAS 15-383.
- [5] Trevor Bennett and Hanspeter Schaub. Capitalizing on relative motion in electrostatic detumbling of axi-symmetric geo objects. In 6th International Conference on Astrodynamics Tools and Techniques (ICATT), Darmstadt, Germany, March 14–17 2016.
- [6] Trevor Bennett, Daan Stevenson, Erik Hogan, Lauren McManus, and Hanspeter Schaub. Prospects and challenges of touchless debris despinning using electrostatics. Advances in Space Research, 56(3):557–568, Aug. 2015.
- [7] Philip Chow, Joseph Hughes, Trevor Bennett, and Hanspeter Schaub. Automated sphere geometry optimization for the volume multi-sphere method. In AAS/AIAA Space Flight Mechanics Meeting, number AAS-16-672, Napa Valley, California, Feb. 14–18 2016.
- [8] V. A. Davis and L. W. Duncan. Spacecraft Surface Charging Handbook. Phillips Laboratory, Directorate of Geophysics, 1992.
- [9] S. E. DeForest. Spacecraft charging at synchronous orbit. Journal of Geophysical Research, 77(4):651–659, Feb. 1972.
- [10] Carolin Früh, Dale Ferguson, Chin Lin, and Moriba Jah. The effect of passive electrostatic charging on near-geosynchronous high area-to-mass ratio objects. International Astronautical Congress, 64, 2013.
- [11] Walton C. Gibson. The Method of Moments in Electromagnetics. Chapman & Hall, November 28 2007.
- [12] E. Grün, D. P. Hamilton, R. Riemann, and S. Dermott et. al. Dust measurements during galileo’s approach to jupiter and io encounter. Science, 274(5286):399, 1996.

- [13] E. Grün, H. Krüger, A. L. Graps, D. P. Hamilton, A. Heck, G. Linkert, H. A. Zook, S. Dermott, H. Fechtig, B. A. Gustafson, M. S. Hanner, M. Horányi, J. Kissel, B. A. Lindblad, D. Linkert, I. Mann, J. A. M. McDonnell, G. E. Morfill, C. Polanskey, G. Schwehm, and R. Srama. Galileo observes electromagnetically coupled dust in the jovian magnetosphere. Journal of Geophysical Research, 103(E9):20011–20022, 1998.
- [14] E. Grün, H. A. Zook, M. Baguhl, A. Balogh, S. J. Bame, H. Fechtig, R. Forsyth, M. S. Hanner, M. Horányi, J. Kissel, B. A. Lindblad, D. Linkert, G. Linkert, I. Mann, J. A. M. McDonnell, G. E. Morfill, J. L. Phillips, C. Polanskey, G. Schwehm, N. Siddique, P. Staubach, J. Svestka, and A. Taylor. Discovery of jovian duststreams and interstellar grains by the ulysses spacecraft. Nature, 362:428–430, 1993.
- [15] Erik Hogan and Hanspeter Schaub. Relative motion control for two-spacecraft electrostatic orbit corrections. AIAA Journal of Guidance, Control, and Dynamics, 36(1):240–249, Jan. – Feb. 2013.
- [16] Erik A. Hogan and Hanspeter Schaub. General high-altitude orbit corrections using electrostatic tugging with charge control. Journal of Guidance, Control, and Dynamics, 28(4):699–705, April 2015.
- [17] Joseph Hughes and Hanspeter Schaub. Charged geosynchronous debris perturbation using rapid electromagnetic force and torque evaluation. In Advanced Maui Optical and Space Surveillance Technologies Conference, Maui, Hawaii, Sep. 20–23 2016.
- [18] John T. Katsikadelis. The Boundary Element Method for Engineers and Scientists. Academic Press, 2 edition, 2016.
- [19] Lawrence Livermore National Laboratory. Using c/c++ and fortran together. <https://computing.llnl.gov/tutorials/bgq/mixedProgramming2.pdf>. Accessed: 2017-02-01.
- [20] John Lekner. Electrostatics of two charged conducting spheres. Proceedings of the Royal Society, 468:2829–2848, 2012.
- [21] J.C. Maxwell. A Treatise on Electricity and Magnetism. Oxford University Press, 1893.
- [22] Shin-Ichiro Nishida and Satomi Kawamoto. Strategy for capturing of a tumbling space debris. Acta Astronautica, 68(1-2):113–120, 2011.
- [23] Smriti Nandan Paul and Carolin Frueh. Space Debris Charging and its Effect on Orbit Evolution. American Institute of Aeronautics and Astronautics, 2016.
- [24] M. A. Peck, B. Streetman, C. M. Saaj, and V. Lappas. Spacecraft formation flying using lorentz forces. Journal of British Interplanetary Society, 60:263–267, July 2007.
- [25] Mason A. Peck. Prospects and challenges for lorentz-augmented orbits. In AIAA Guidance, Navigation and Control Conference, San Francisco, CA, August 15–18 2005. Paper No. AIAA 2005-5995.
- [26] J. N. Reddy. An Introduction to the Finite Element Method. McGraw-Hill Education, 3 edition, 2005.

- [27] Hanspeter Schaub. Stabilization of satellite motion relative to a coulomb spacecraft formation. AIAA Journal of Guidance, Control, and Dynamics, 28(6):1231–1239, Nov.–Dec. 2005.
- [28] Hanspeter Schaub. Faster-than-realtime electrostatic force and torque modeling for ssa applications. AFOSR Annual Report, University of Colorado, Aerospace Engineering Sciences Department, Boulder, CO, September 14 2016.
- [29] Hanspeter Schaub, Trevor Bennett, and Joseph Hughes. Current developments in three-dimensional electrostatic detumble of axi-symmetric geo debris. In 4th International Workshop on Space Debris Modeling and Remediation, CNES, Paris, June 6–8 2016. Paper No. 6.4.
- [30] Hanspeter Schaub and Daniel F. Moorer. Geosynchronous large debris reorbiter: Challenges and prospects. In AAS Kyle T. Alfriend Astrodynamics Symposium, Monterey, CA, May 17–19 2010. Paper No. AAS 10-311.
- [31] Hanspeter Schaub, Gordon G. Parker, and Lyon B. King. Challenges and prospects of coulomb spacecraft formation control. Journal of the Astronautical Sciences, 52(1-2):169–193, 2004.
- [32] Hanspeter Schaub and Zoltán Sternovsky. Active space debris charging for contactless electrostatic disposal maneuvers. In 6th European Conference on Space Debris, Darmstadt, Germany, April 22–25 2013. ESOC. Paper No. 6b.O-5.
- [33] W. R. Smythe. Static and Dynamic Electricity. McGraw–Hill, 3rd edition, 1968.
- [34] Ludwick A. Sobiesiak and Chris J. Damaren. Lorentz-augmented spacecraft formation reconfiguration. In AIAA/AAS Astrodynamics Specialist Conference, San Diego, California, Aug. 4–7 2014.
- [35] Daan Stevenson. Optimization of sphere population for electrostatic multi sphere model. In 12th Spacecraft Charging Technology Conference, Kitakyushu, Japan, May 14–18 2012.
- [36] Daan Stevenson and Hanspeter Schaub. Multi-sphere method for modeling electrostatic forces and torques. Advances in Space Research, 51(1):10–20, Jan. 2013.
- [37] Brett Streetman and Mason Peck. Gravity-assist maneuvers augmented by the lorentz force. Journal of Guidance, Control, and Dynamics, 32(5):1639–1647, 2009.
- [38] Brett Streetman and Mason A. Peck. General bang-bang control method for lorentz augmented orbits. Journal of Spacecraft and Rockets, 47(3):484–492, May–June 2010.
- [39] N. A. Tsyganenko. Modeling the earth’s magnetosphere using spacecraft magnetometer data. <http://geo.phys.spbu.ru/~tsyganenko/modeling.html>. Accessed: 2017-02-01.
- [40] N. A. Tsyganenko. Modeling the earth’s magnetospheric magnetic field confined within a realistic magnetopause. Journal of Geophysical Research, 100(A4):5599–5612, 1995.
- [41] N. A. Tsyganenko. Effects of the solar wind conditions on the global magnetospheric configuration as deduced from data-based field models. In International Conference on Substorms, Versailles, France, May 12–17 1996.
- [42] N. A. Tsyganenko. Solar wind control of the tail lobe magnetic field as deduced from geotail, ampte/irm, and isee 2 data. Journal of Geophysical Research, 105(A3):5517–5528, 2000.

- [43] N. A. Tsyganenko. A model of the near-magnetosphere with dawn-dusk asymmetry. Journal of Geophysical Research, 107(A8):10-2-10-17, 2002.
- [44] N. A. Tsyganenko. Modeling the dynamics of the inner magnetosphere during strong geomagnetic storms. Journal of Geophysical Research, 110(A03208):1-16, 2005.
- [45] Wenfu Xu, Bin Liang, Dai Gao, and Yangsheng Xu. A space robotic system used for on-orbit servicing in the geostationary orbit. In IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, Oct. 18-22 2010.
- [46] Wenfu Xu, Bin Liang, Bing Li, and Yangsheng Xu. A universal on-orbit servicing system used in the geostationary orbit. Advances in Space Research, 48(1):95-119, 2011.

## Appendix A

### Storm-Time Magnetic Field Modeling

#### A.1 Motivation to Consider Solar Storm Effects

Solar storms are capable of greatly altering nominal planetary magnetic field conditions. This effect is troublesome, in particular because spacecraft and debris passing through magnetospheric substorm regions, which trap high energy charged particles, can become charged to tens of kilovolts [9, 8]. The surface charging during such events can be so dramatic that arcing can occur between points on the spacecraft surface with large potential differences [8]. From a dynamic modeling perspective, the combination of greatly altered magnetic field conditions and high surface charging can lead to significant errors in the Lorentz forces and torques a spacecraft or debris object is predicted to experience. If a standard planetary magnetic field model like the International Geomagnetic Reference Field (IGRF) or a dipole model is used without compensating for solar storm effects, it is not reasonable to assume Lorentz force and torque predictions using any electrostatic modeling technique will be accurate.

The goal of this appendix is to alert the reader to a new development in magnetic field modeling software that greatly reduces the amount of effort and programming expertise required to compensate for strong solar effects on Earth's magnetic field. This chapter makes a brief mention of a family of models that can be utilized to compensate for solar effects on Earth's magnetic field, and an associated set of subroutines, called Geopack, that can be used to perform calculations for the IGRF and dipole internal magnetic field models, calculate coordinate transformations between various useful geocentric reference frames, trace magnetic field lines, and more. All magnetic

field model and Geopack subroutines, originally implemented in Fortran, are wrapped in C code to make the subroutines available and easy to use in standalone magnetic field and large-scale astrodynamics simulations built using C/C++. The resulting C code is wrapped once more using the Simple Wrapper Interface Generator (SWIG) software, compiled as a shared object and made available to Python as a module. Figure A.1 clarifies the procedure used to make Tsyganenko models and Geopack available in C and Fortran. A figure is generated using the resulting Python module and compared to a figure generated using the original Fortran code.

## A.2 Tsyganenko External Magnetic Field Model and Geopack

One method to compensate for unusually strong solar effects is to superimpose an external magnetic field model on top of a standard internal magnetic field model. Tsyganenko models are a family of semi-empirical external magnetic field models, fit to many spacecraft observations, that have been refined and updated many times since the early 1980s [42, 43]. These models take interplanetary magnetic field measurements, geomagnetic indices, derived quantities and a position vector as inputs, calculate an external magnetic field and superimpose the calculated field on top of either a dipole or IGRF magnetic field model. Their return value is the complete  $B$ -field vector at a point in space corresponding to the input position vector. A detailed discussion of the features and use of Tsyganenko models is not intended here, and the interested reader is referred to the many papers on them [42, 43, 40, 41, 44].

The Geopack subroutines, also created by Tsyganenko, are a collection of 20 subroutines that can be used for a variety of purposes relating to geomagnetic field modeling. Subroutines exist for the purposes of transforming between a multitude of geocentric reference frames, calculating IGRF and dipole fields, determining sun location given a time in UTC, calculating the magnetopause boundary, tracing field lines and more. These subroutines are an invaluable resource for geomagnetic field modeling and the coordinate transformation and sun-vector capabilities contained within Geopack have many applications elsewhere. Tsyganenko models, Geopack and referenced Tsyganenko papers are freely available for download from Tsyganenko’s personal academic website [39].



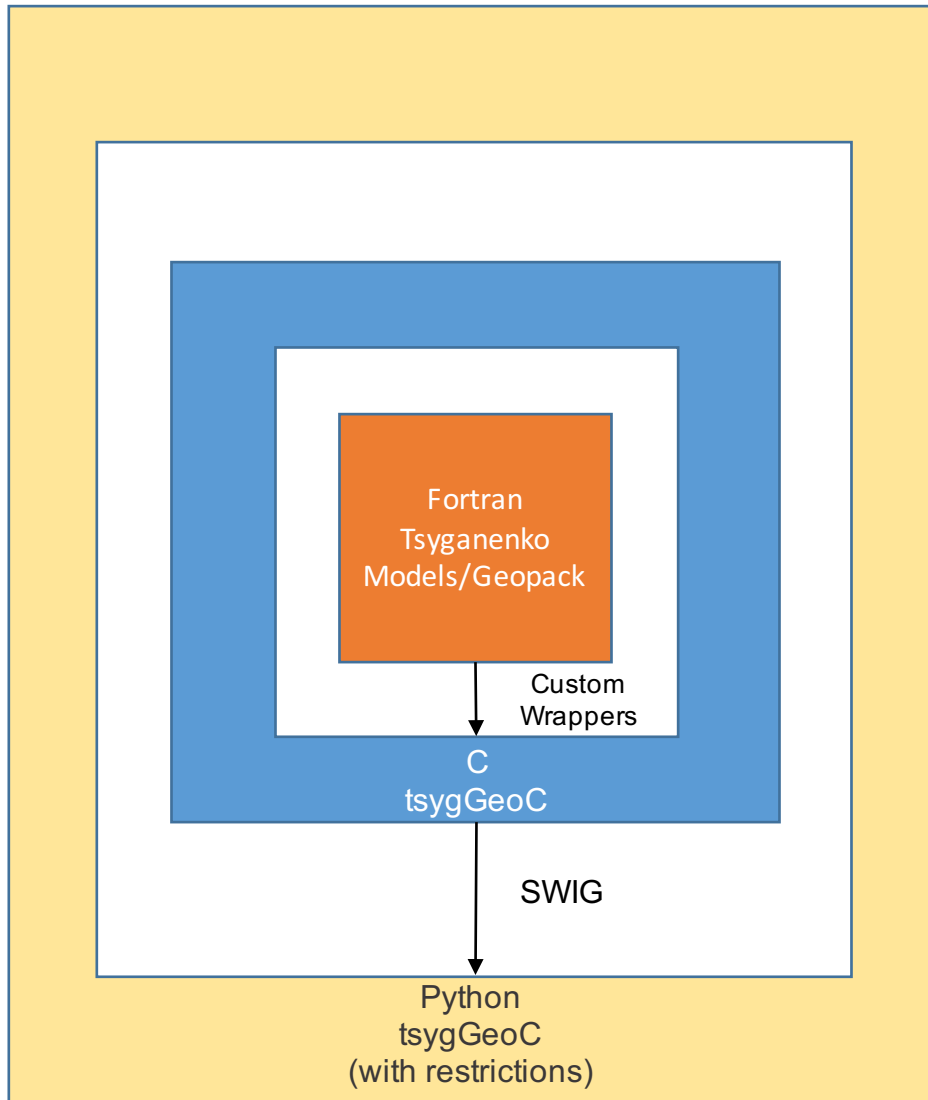


Figure A.1: Wrapping Procedure used to Make Tsyganenko Models and Geopack Accessible from C and Python. The Underlying Fortran Code is First Wrapped with C Using Mixed Programming. The C Code is then Wrapped Automatically Using the Simple Wrapper Interface Generator to Create an Importable Python Module with the Same Functionality as the Original Fortran Code

### A.3 `tsygGeoC`

The `tsygGeoC` C header and source files contain detailed documentation and wrappers for Fortran Tsyganenko model and Geopack subroutines. The primary motivation for developing C wrappers is the potential future inclusion of magnetic field modeling and calculations in the Autonomous Vehicle Systems laboratory Basilisk astrodynamics simulation software, which has

been developed primarily in C++. A secondary consideration for development of C wrappers for the Fortran subroutines is ease of use—most engineering students have not learned Fortran. In addition, Fortran is dated, arcane, and unfamiliar even for programmers versed in other compiled languages. Unless coding standards are followed rigorously, Fortran allows (almost encourages) bad coding practices, and contains 60 years worth of backwards compatibility which can make understanding source code a hassle.

Fortran code is made available to a C compiler by first compiling the Fortran subroutines contained in the Tsyganenko models and Geopack. C-style functions are declared as type “extern void” in **tsygGeoC.h**, and named appropriately given the Fortran compiler naming conventions. C-style functions representing Fortran subroutines must be named to match exactly with the symbols in the Fortran object file. If the symbols representing Fortran subroutines in the Fortran compiler object code, they can be found using a utility such as **nm**; however, Fortran compiler flags exist which will enforce particular naming conventions. These flags allow for standardization of the format of the C functions names associated with Fortran subroutines. During C compilation, the Fortran object code and the Fortran compiler library used to make it are linked.

When declaring the C functions associated with Fortran subroutines, the structure of Fortran must be taken into account. Fortran subroutines are always pass-by-reference. As such, inputs to associated C functions must be memory addresses. As a result, particular care must be taken to ensure that the data type and size of C function inputs is matched with what the Fortran subroutine expects. If the inputs are not matched exactly, but the names and number of inputs to a C function associated with a Fortran subroutine are, compilation and execution will in all likelihood proceed, but the C function output will be junk. Care must be taken to ensure that C output matches Fortran subroutine output. Tables showing correspondence between C and Fortran data types are easily found online.

Fortran common blocks are structured equivalently to C structs. To make common block data available in C, a C struct is declared matching the name of the Fortran common block precisely. The size and type of C struct fields must match exactly with those defined in the Fortran common block.

The resulting C struct is a much safer representation of the contiguous block of data defined in the Fortran common block declaration. For further discussion of mixed-programming using C and Fortran, the interested reader is referred to Lawrence Livermore National Laboratory's notes [19].

If naming conventions are followed, Fortran structure is considered and input data types are declared correctly, mixed-programming using C and Fortran is a relatively smooth process. The C wrappers developed for **tsygGeoC** are designed primarily to hide the pass-by-reference nature of Fortran from the C programmer. The Tsyganenko model C function wrapper is designed to take a function pointer to the underlying Tsyganenko model Fortran subroutine to use as a callback function, making adding and implementing additional Tsyganenko models in C trivial. The resulting C code allows access to Fortran subroutines using familiar C-style functions, and hides Fortran code as much as possible while giving the C programmer the same access to subroutines and variables as the Fortran programmer. To verify the C code, unit tests in Fortran and C are written that call subroutines and their associated C wrapper functions with identical inputs, and the outputs are compared. The outputs of both are identical.

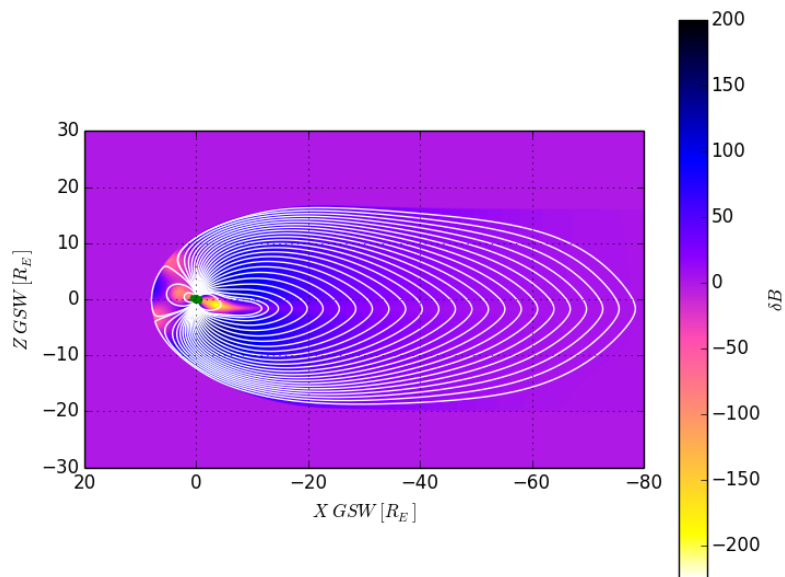
#### A.4 **tsygGeoC Python Module**

SWIG affords simple wrapping of C/C++ code for use in many easier-to-use scripting languages, including Python. **tsygGeoC** is wrapped using SWIG to create a module that can be imported by Python. This affords another level of abstraction and reduction in complexity when compared with the C code. SWIG typemaps, which defined a correspondence between Python and C inputs and outputs, are exploited to create natural Python function calls and return values. Inputs to Python functions associated with the underlying C code are nearly identical, except for functions associated with underlying C functions that take output arrays as inputs. The **numpy.i** interface file is used to define SWIG typemaps that hide output array inputs from the Python programmer, and instead output arrays are created automatically at the C level and returned as numpy arrays. The goal of the Python implementation of **tsygGeoC** is to make access to the Tsyganenko and Geopack Fortran subroutines as familiar as possible for a programmer fluent in

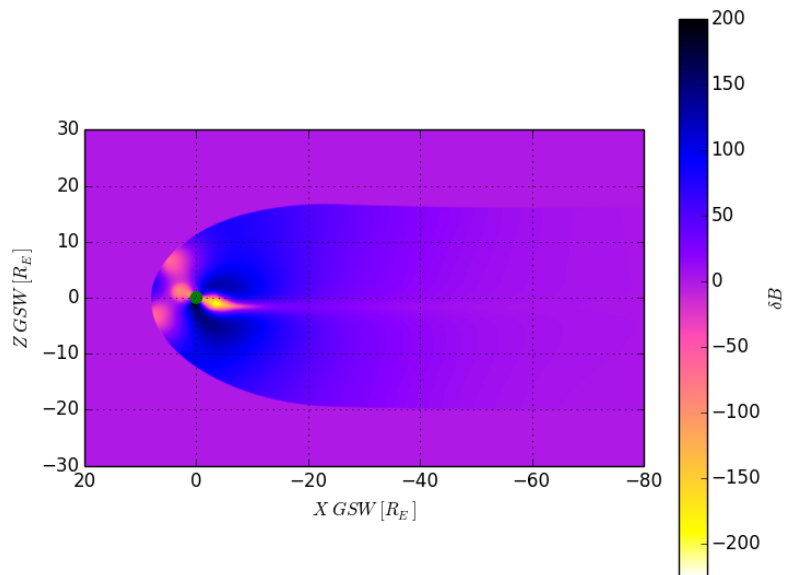
Python.

One sacrifice is made in this implementation— Fortran common blocks can be read from but, in general, not written to. Two Fortran common blocks, “GEOPACK1” and “GEOPACK2” are available for manipulation at the C and Fortran levels. “GEOPACK2” is used to store IGRF coefficients and should never be manipulated manually. “GEOPACK1” affords the option of manually setting the dipole tilt angle. In Python, “GEOPACK2” data can be extracted, but no manipulation of the data is allowed. A helper function is written in C to allow for manually defining the dipole tilt angle at the Python level, and all other data in “GEOPACK1” is read-only. Compilation of the SWIG **tsygGeoC** library requires linking to a Python interpreter and a Fortran compiler library.

Regardless of the difficulty associated with creating it, using the Python implementation of **tsygGeoC** allows for simple and fast exploitation of Geopack’s field line plotting capabilities. Sort plotting scripts can be written and all data processed automatically, whereas the same procedure in Fortran requires writing a field line data file and plotting externally. Figures A.2(a) and A.2(b) are colormap plots displaying the scalar difference in the magnitudes of the total field and an internal dipole field in nT for a solar event that occurred on 10 October 1990, the first with field lines and the second without field lines for clarity. Light colors represent regions of internal magnetic field compression. Compare these figures with Figure A.3, which is shown in one of the papers on the Tsyganenko 2001 model [43]. Figures A.2(a) and A.2(b) agree with Figure A.3, but are created quickly conveniently using Python plotting libraries.



(a) Colormap with Field Lines Superimposed



(b) Colormap without Field Lines

Figure A.2: Colormap of Scalar Difference Between Magnitude of Total Field Vector and Internal Dipole Field Vector in nanoTesla for 10 October 1990 Solar Event. This Figure Clearly Shows the Significant Effects a Solar Storm can have on Earth's Magnetic Field

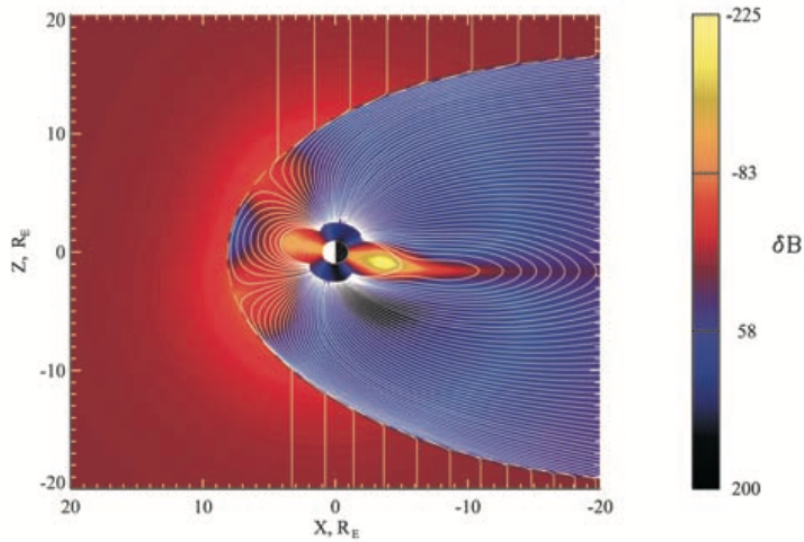


Figure A.3: Colormap of Scalar Difference Between Magnitude of Total Field Vector and Internal Field Dipole Field Vector for 10 October 1990 Solar Event as Computed by Tsyganenko [43].

## A.5 Conclusion

Source code and header files for C wrappers of Tsyganenko models and Geopack subroutines have been developed for ease of use and future implementation in projects built on C/C++. The resulting software hides underlying Fortran code from the C/C++ programmer without incurring significant overhead. In addition, a Python module has been created from the C wrappers, which allows for access of underlying Fortran subroutines with the scripting convenience of Python. The Python implementation is designed such that inputs and outputs from the **tsygGeoC** Python module functions are easily recognizable to a programmer fluent in Python. It is shown that results at the Python level match results in Tsyganenko's own work.

## Appendix B

### MSMOptimizer3D

#### B.1 Summary of Optimization Program Modifications

The **MSMOptimizer3D** program, as written by Christine Reilly and Trevor Bennett, functions to fit VMSM models to FEM generated force and torque data. Modifications are made in this work to extend the program to fit to analytically generated  $E$ -field data. A list of modifications appears below:

- A new variable called **fitType** is introduced to allow selection between fitting to force and torque data or fitting to  $E$ -field data.
- Logic is introduced to parse  $E$ -field data, filter it by shell radius and store it in a new **Efield** data structure with fields corresponding to  $E$ -field vector components.
- A new function, **MSMEfieldCalc.m**, is included to sum the  $E$ -field vector components contributed by each MSM model sphere at some relative position to a MSM model.
- Many low-level functions used for calculating cost are modified with logic to use the new  $E$ -field fitting cost function.
- A self-capacitance constraint is developed and introduced to the constraint function for the box-and-panel geometry.

## B.2 MSMCall.m

**MSMCall.m** is the main script used to begin an optimization run. The user establishes inputs to the optimization run in this script. Two definitions must be made: the input data file describing electric field or force and torque data must be defined, and the **fitType** variable, a string that can take on values of either “E-field” or “Force/Torque,” must be defined in order to use the correct cost function and *E*-field calculations in lower-level function calls. A display of the final state is currently hardcoded into this script, and must be modified to display the correct state based on the states that are being used for optimization.

Assignment to the **Names** data structure also takes place within this script. The **Names** data structure stores function handles for the cost function and three geometry-dependent functions that must be written by the user: the **input** function, which describes the state to be optimized and loads data, the **const** function, which describes nonlinear equality and inequality constraints the optimization process will be subjected to, and the **dist** function, which describes how the optimizer states relate to model geometry. The **Optimizer** variable stores a function handle to whatever optimizer will be used, and options for optimization are set in it. No significant alterations to the **Optimizer** function from the original are made in this work.

## B.3 Names.input

The **input** function is used to specify initial optimizer state, bounds on the state for local optimization, and to load the input data. Filtering of input data by shell radius is done most appropriately in this function. The number of spheres in the model is defined using the **commandSpheres** and **targetSpheres** variables. For *E*-field fitting, the number of command spheres is 0. The variable **maxwellCapac** is used to store the FEM computed self-capacitance of the target model for use in enforcing a self-capacitance constraint.



## B.4 Names.dist

The **dist** function is used to relate the optimization state to MSM model geometry. For instance, if symmetry is to be used to decrease the search space of the optimizer, it is enforced in this function. Optimizer position states are extracted and stored in the array **initialSphereCoord**, which describes the positions of each MSM model sphere. Likewise, the sphere radius states are extracted from the optimizer state array and stored in the **sphereRadius** array.

## B.5 Names.const

The **const** function is used to define nonlinear equality and inequality constraints the optimization is subject to. For instance, if self-capacitance matching is enforced, it is done here as an equality constraint. Relevant inequality constraints might be that model spheres must not overlap.

## B.6 Conclusion

A summary of modifications extending the original **MSMOptimizer3D** program to handle  $E$ -field fitting is shown. Brief summaries of scripts and functions that must be modified to accommodate specific geometry and data input are included. Relatively minor alterations of existing Matlab software are required to extend MSM optimization to  $E$ -field fitting from force and torque fitting.