Article

Lunar Finite Element Gravity Model

Giaky Nguyen , Ahmad Bani Younes * , Ahmed Atallah

Posted Date: 2 September 2024

doi: 10.20944/preprints202409.0081.v1

Keywords: Spherical harmonics model; Finite element model; Lunar gravity model

Article

Lunar Finite Element Gravity Model

Giaky Nguyen [1], Ahmad Bani Younes [2,*] and Ahmed Atallah [1]

[1] Graduate Student, Aerospace Engineering, San Diego State University, 5500 Campanile Drive, San Diego, CA. 92182-1308
[2] Associate Professor, Aerospace Engineering, San Diego State University, 5500 Campanile Drive, San Diego, CA. 92182-1308, abaniyounes@sdsu.edu
* Correspondence: abaniyounes@sdsu.edu; Tel.: +1 (619) 594-2931

Abstract: In this paper, we integrate the Finite Element Method (FEM) with orthogonal polynomial approximation in high-dimensional spaces to innovatively model the Moon's surface gravity anomaly. Our aim is to approximate solutions to Laplace's classical differential equations of gravity, employing classical Chebyshev polynomials as basis functions. Using classical Chebyshev polynomials as basis functions, the least-squares approximation was used to approximate discrete samples of the approximation function. These test functions provide an understanding of errors in approximation and corresponding errors due to differentiation and integration. These test functions provide an understanding of errors in approximation and corresponding errors due to differentiation and integration. The first application of this project is to substitute the globally-valid classical spherical harmonic series of approximations with locally-valid series of orthogonal polynomial approximations (i.e. using the FEM approach). With an error tolerance set at 10^{-9} m s^{-2}, we use this method to adapt the gravity model radially upwards from the lunar surface. The results showcase a need for a higher degree of approximation on and near the lunar surface, with the necessity decreasing as the radius increases. Notably, this method achieves a computational speedup of five orders of magnitude when applying the method to radial adaptation. More intrinsically, our second application involves using the methodology as an effective tool in solving boundary value problems. Specifically, we implement this approach to solve classical differential equations involved with high precision, long-term orbit propagation. This application provides a four-order of magnitude speedup in computational time while maintaining an error within the 10^{-10} m s^{-2} error range for various orbit propagation tests. Alongside the advancements in orthogonal approximation theory, the FEM enables revolutionary speedups in orbit propagation without compromising accuracy.

Keywords: spherical harmonics model; finite element model; lunar gravity model

1. Introduction

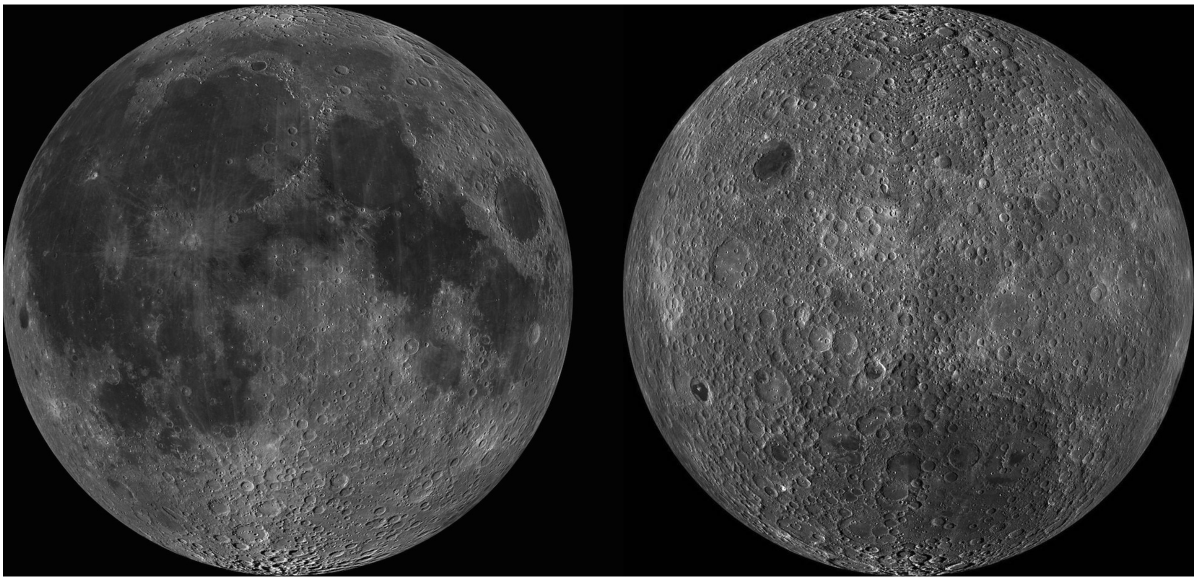


Figure 1. Left: The Moon's near side. Right: The Moon's far side. Image credit: [1].

On March 4, 2022, at approximately 4:25 a.m. PST, a rocket component was reported to have crashed into the far side of the Moon [2]. Although the specific rocket component is still not verified, losing control over spacecraft during lunar orbit missions has been an ongoing issue since the launch of the Luna 1 rover in 1959. Moreover, scientists have noticed a peculiar phenomenon about the lunar surface gravity: as satellites and probes orbited the Moon over certain craters and impact basins, they periodically veered off course and sometimes even plummeted toward the lunar surface. This is a particular problem when observing these effects on the far side of the Moon (See Fig. 1). It wasn't until the Apollo 15 and 16 missions that scientists described the Moon's surface gravity as "lumpy" due to the differing mass concentrations caused by the geology. These fluctuations in the lunar surface gravity have always puzzled scientists, which led to a high demand for precise lunar gravity data in the hopes of avoiding these crashes.

From the Lunar Reconnaissance Orbiter (LRO) [3] to the recent Artemis program [4], the field of lunar-based orbit propagation research has been increasingly growing over the decades. As a result of this, the importance of modeling the lunar surface gravity anomaly to aid in future Moon missions has become crucial in predicting the trajectories of spacecraft orbiting around the lunar surface. Using high-resolution gravity data, many lunar surface gravity models, such as NASA's Gravity Recovery and Interior Laboratory's (GRAIL) GRGM1200B model [5] and the GSFC LP-100k lunar potential Model [6], have been created as an attempt to constrain lunar surface gravity anomalies in response to this demand. Although sufficient enough to model the lunar surface gravity, these existing models rely on the global spherical harmonic (SH) series to solve Laplace's equation for gravity. The SH method for solving the equation of gravity was initially implemented as a way to model the geopotential through the use of Earth-based satellites recognized in the late 1950s [7]. Recent projects that implement gravitational acceleration modeling about the Moon can be viewed in the following literature: [8], [9], and [10]. However, the SH approach proves to be computationally demanding, with three main issues arising: [11–13]:

1. At polynomial orders $n > 2$, convergence proves to be inefficient and slow, which requires tens of thousands of terms to obtain accuracy for higher-order gravity representation.
2. In any astronomical surface gravity modeling, the north and south poles represent singularities in SH which result in poor approximations at the poles. For further examinations on obstacles in SH gravity approximations [14–16]
3. SH is only accurate with spherical-shaped astronomical bodies. This means that the gravity of objects such as comets and oblate bodies cannot be modeled using this method.

We hope to introduce a FEM local gravity representation of the higher order perturbation to offset the issue of slow convergence of the gravity models at polynomial orders greater than 2. Specifically, we hope to use lower-order functions to sufficiently model the gravity perturbations about the lunar surface. This is highly motivated by literature where the projects propose similar advancements to the classical SH approach, such as in Junkins [17] and, more importantly, in recent investigations by the following: [18–20].

This paper details a new approach to modeling surface gravity anomaly, increasing computational speed while maintaining a conservative error. Specifically, we propose replacing the current LP-100k lunar SH gravity model with a series of local orthogonal polynomial approximations implemented by the Finite Element Method (FEM) of modeling lunar surface gravity. In Bani Younes's dissertation [21], a FEM surface gravity model was implemented for the geopotential, which holds significant relevance to our applications on the lunar FEM model. With this paper, steps have been taken to produce a cost-effective and accurate FEM lunar surface gravity model that will mirror that of the dissertation. Furthermore, parallelization of the lunar FEM surface model will be implemented building upon the research of [22] where a parallel evaluation of Chebyshev approximation was studied. Parallelization of the original FEM lunar model and astrodynamics applications using the FEM model will further encourage computational speedup in both regards. With this, creating a FEM lunar surface gravity model will elevate the surface gravity modeling field with these main contributions:

1. Computational efficiency
2. Memory allocation
3. Precision
4. Parallelization

1.1. Finite Element Method

There have been a variety of approximation techniques and basis functions implemented for gravity representation which includes weighted functions [12,23,24], wavelets [25], splines [26], octrees [27], psuedocenters [28] and 3D digital modeling [29]. FEM can be used in a wide variety of applications that are both interesting and important in the fields of mathematical modeling, engineering, and computational analysis. Some examples include using FEM to analyze electric sail static structures [30], determination of the gravity potential and tidal stresses of asteroids [31], and using FEM to analyze a lunar lander during a soft landing [32]. Similar to our approach, each method balances accuracy with efforts to encourage computational speed. FEM was first proposed by Junkins [17] as a way to interpolate the geopotential of Earth-based gravitational modeling. Bani Younes [21] provided the computational framework to implement FEM to model the geopotential, acceleration due to gravity, and radial adaption about the Earth's surface. With this descent, the purpose of this paper is to interpolate the lunar gravitational acceleration about the surface and above the surface to help aid future lunar-based spacecraft missions that require precise orbital propagation.

Using the lunar FEM model, truncating the classical expansion at $n = 2$ will allow for local gravity representation at the higher-order perturbations. The FEM modeling approach is applicable to any spherical model and even to irregularly shaped astronomical bodies, while SH is purely for spherical bodies. Furthermore, this method increases computation by trading computer memory for run-time speed.

For our purposes, FEM will be used on the lunar surface, where the method will subdivide the entire lunar surface as a large system and subdivide it into smaller finite elements. The gravitational differential equations will then be solved in each shell and will then be connected by nodes. However, determining the correct shell iteration for these higher-order approximations is key with this method.

1.2. Chebyshev Polynomial Interpolation

Developing a more efficient theory to solve higher-order Laplace's equation for gravity requires a structure that is both radially adaptive and efficient. If implemented correctly, interpolation at any radial or point on the surface can be made. One approach involves using the Chebyshev polynomial method. Created by mathematician Pafnuty Chebyshev in the nineteenth century, Chebyshev polynomials are two sequences of polynomials related to the cosine and sine functions denoted as $T_n(x)$ and $U_n(x)$, respectively. The polynomials can be defined starting with trigonometric functions where:

The Chebyshev polynomials of the first kind (T_n) are given by:

$$T_n(cos\,\theta) = cos(n\,\theta) \tag{1}$$

and the polynomials of the second kind (U_n) are:

$$U_n(cos\,\theta)\,sin(\theta) = sin((n+1)\,\theta) \tag{2}$$

in which n is any integer number that can be used to convert these definitions into a polynomial form. Chebyshev polynomials are orthogonal polynomials that are solutions of a special kind of Sturm-Liouville differential equation called a Chebyshev differential equation. Most importantly, the roots of $T_n(x)$ i.e. the Chebyshev nodes, are used as estimation points. Approximating at this point produces efficient polynomial interpolation, which is useful for precise radial adaptation and other applications which require higher-order approximation.

Although typically used as a numerical method for solving initial value problems (IVPs) and certain classes of boundary value problems (BVPs), this method can also be used to approximate solutions of linear or non-linear ordinary differential equations (ODEs) like that of Laplace's equations for gravity. Using these Chebyshev polynomials, an orthogonal approximation can be placed for our FEM modeling process in which the path approximations would encourage computational speedup.

1.3. Orbit Propagation

The last application in this project includes using the FEM model to approximate a satellite's orbital trajectory over a provided time span. Similar works which inspired this method of orbital propagation can be found in the following literature: [33–35]. This application will be achieved using the resulting Chebyshev polynomial coefficients from the original FEM routine at specific radii above the lunar surface to approximate the acceleration a satellite with negligible mass would experience during the course of a short and long-term orbit around the Moon. This propagation considers the two-body problem, which consists of a satellite with negligible mass orbiting a larger astronomical body. This problem allows us to use Newton's second law force equations to then come up with a finalized two-body equation of motion for the theoretical satellite orbiting the Moon. In this paper, we will first cover in detail the orthogonal approximation techniques used alongside Chebyshev polynomials in Section II. Then in Section III, the process of utilizing FEM on the LP-100k model will be discussed and compared to the original SH model. In Section IV, we will discuss how we implemented the coefficients produced by the FEM model on a two-body orbit propagation problem about the lunar surface. Lastly, in Section V, we will discuss a summary of the finding and concluding remarks.

2. Orthogonal Approximation

The following are some examples of treatments of discrete approximation using Chebyshev polynomials: [24,36–43]. Although [39,40] hold more significance in the field of orthogonal approximation, in [42], the orthogonal approximation is used to show that multi-resolution approximation of input and output maps can be linear and nonlinear. Typically, a linear least-squares approximation is used for an orthogonal approximation when dealing with experimental data or an arbitrary function generating polynomials. In this paper, a least-squares approximation with classical Chebyshev polynomials is used to analyze discrete samples of the "to-be" approximated function. Furthermore, arranging the regression matrix to be "Kronecker factorable" (a technique used in [21]) allows the use of array algebra identities to generate the multidimensional orthogonal least-squares operators. This approach can be seamlessly adapted to higher-dimensional approximations. This section summarizes how using the Kronecker product technique extends a single classical Chebyshev polynomial approximation result to higher dimensions. This method avoids dimensional challenges related to how many attributes a dataset has and paves the way for efficient computation. The importance of this method allows considering the advantages of orthogonal approximation in gravity modeling. Most mathematical derivations are based on the work presented in [21].

First, let's briefly consider the following approximation of a single-valued function, $f(x)$, with x as one independent variable:

$$g(x), \{x_{min} \leq x \leq x_{max}\} \tag{3}$$

For simplification in multi-dimensional analysis, we introduce a new independent variable ξ where $\{-1 \leq \xi \leq 1\}$. Thus, through the process of normalization, the forward and inverse transformation for ξ can be produced:

$$\begin{aligned} Forward &\left\{ \xi(x) = 2(x - x_{min})/(x_{max} - x_{min}) - 1 \right. \\ Inverse &\left\{ x(\xi) = x_{min} + \ (\xi + 1)\ (x_{max} - x_{min})/2 \right. \end{aligned} \tag{4}$$

The final function to be approximated can be created by substituting the forward and inverse functions into equation 3:

$$f(\xi) = g(x_{min} + (\xi + 1)\,(x_{max} - x_{min})/2\,) \tag{5}$$

In a general case, for basis functions, $\phi_n(\xi)$, we seek to approximate $f(\xi)$ as a linear combination of a set of $N+1$ linearly independent basis functions $\{\phi_0(\xi), \phi_1(\xi), ..., \phi_N(\xi\}$ as the following:

$$f(\xi) \approx \sum_{n=0}^{N} a_n \phi_n(\xi) \tag{6}$$

For discrete cases, we introduce a set of sample points in which we call **nodes** as $\{\xi_0, \xi_i, ..., \xi_M; M \geq N\}$. With this, we can consider the residual approximation error at each node to be:

$$r_j = f(\xi_j) - \sum_{n=0}^{N} a_n \phi_n(\xi);\; j = 0, 1, ..., M \tag{7}$$

In general, the decision to use a least-squares approximation routine is motivated by the advantages of applying the one-dimensional analysis to higher dimensions. This motivation is discussed later in Section 2.4. Using least-squares approximation, we want to find the coefficient vector a that minimizes the weighted sum square of the residuals:

$$J = \frac{1}{2}(f - \Phi a)^T W (f - \Phi a); W = W^T \tag{8}$$

where $\Phi = \sum_{n=0}^{N} a_n \phi_n(\xi)$, and W is a positive definite weighted matrix determined by classical orthogonal conditions for Chebyshev polynomials (see Section 2.2).

Using a weighted least-squares solution and orthogonal conditions, this result can be displayed as normal equations:

$$a = (\Phi^T W \Phi)^{-1} \Phi^T W \mathbf{f} = C_x \mathbf{f} \tag{9}$$

This finalized relationship can be displayed in matrix form as the following:

$$f = \begin{bmatrix} f(\xi_0) \\ f(\xi_i) \\ \vdots \\ f(\xi_N) \end{bmatrix}, C_x = \frac{1}{M_x} \begin{bmatrix} \frac{T_0(\xi_0)}{2} & T_0(\xi_1) & \cdots & T_0(\xi_{M_{x-1}}) & \frac{T_0(\xi_{M_x})}{2} \\ T_1(\xi_0) & 2T_1(\xi_1) & \cdots & 2T_1(\xi_{M_{x-1}}) & T_1(\xi_{M_x}) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ T_{N_x-1}(\xi_0) & 2T_{N_x-1}(\xi_1) & \cdots & 2T_{N_x-1}(\xi_{M_{x-1}}) & T_{N_x-1}(\xi_{M_x}) \\ T_{N_x}(\xi_0) & 2T_{N_x}(\xi_1) & \cdots & 2T_{N_x}(\xi_{M_{x-1}}) & T_{N_x}(\xi_{M_x}) \end{bmatrix}, a = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix}$$

Similarly, for more than one independent variable, we can consider the function $f(x, y)$, with the two independent variables x and y:

$$g(x, y), \{x_{min} < x < x_{max}\}, \{y_{min} < y < y_{max}\} \tag{10}$$

as in our one variable approximation example, we will use a set of new independent variables, ξ, and η, to reduce the computational burden of dimensionality which leads to the normalized forward and inverse functions:

$$
\begin{aligned}
&Forward\begin{cases} \xi(x) = -1 + 2(x - x_{min})/(x_{max} - x_{min}) \\ \\ \eta(x) = -1 + 2(y - y_{min})/(y_{max} - y_{min}) \end{cases} \\
&Inverse\begin{cases} x(\xi) = x_{min} + \ (\xi + 1)\ (x_{max} - x_{min})/2 \\ \\ y(\eta) = y_{min} + \ (\xi + 1)\ (y_{max} - y_{min})/2 \end{cases}
\end{aligned}
\tag{11}
$$

Substituting these equations into equation 11 provides us with the function we wish to approximate:

$$
f(\xi,\eta) = g(x(\xi), y(\eta)) = g\left(x_{\min} + (\xi+1)\frac{(x_{\max} - x_{\min})}{2}, y_{\min} + (\eta+1)\frac{(y_{\max} - y_{\min})}{2}\right) \tag{12}
$$

and in the discrete measurement case tested at sample nodes, $\{\xi_0, \xi_1, ..., \xi_{M_x}; M_x \geq N_x\}$, $\{\eta_0, \eta_1, ..., \eta_{M_y}; M_y \geq N_y\}$, the residual approximation error at each node is:

$$
r_{ij} = f(\xi_i, \eta_j) - \sum_{\alpha=0}^{N_x} \sum_{\beta=0}^{N_y} a_{\alpha\beta}\phi_{\alpha\beta}(\xi_i, \eta_j);\ i = 0, 1, ..., M_x,\ j = 0, 1, ..., M_y \tag{13}
$$

Which can be displayed in vector-matrix notation: $r = f - \Phi a$ where $\Phi = \sum a_{ij}$. When in matrix form, a common issue that needs to be considered is dimensionality. To avoid this problem, efficient choices of basis functions and associated sample points must be made. This technique is examplified in [44]. Referring to sections 1.6.2, Φ is formed as the Kronecker product of two basic matrices:

$$
\Phi = \begin{bmatrix} 1 & \xi_0 & \xi_0^2 \\ 1 & \xi_1 & \xi_1^2 \\ 1 & \xi_2 & \xi_2^2 \\ 1 & \xi_3 & \xi_3^2 \end{bmatrix} \otimes \begin{bmatrix} 1 & \eta_0 \\ 1 & \eta_1 \\ 1 & \eta_2 \end{bmatrix} = \Phi_x \otimes \Phi_y \tag{14}
$$

For our purposes, the Kronecker product, $C = A \otimes B$ will be implemented in MATLAB [45] via the "kron" command. Subsequently, our weighted least-square solution matrix becomes:

$$
a = (\Phi^T W \Phi)^{-1}\Phi^T f = \left\{\left[(\Phi_x^T \Phi_x)^{-1}\Phi_x^T\right] \otimes \left[(\Phi_y^T \Phi_y)^{-1}\Phi_y^T\right]\right\} f \tag{15}
$$

As equation 2.19 is "Kronecker favorable", it can be further represented as $\Phi = \Phi_x \otimes \Phi_y$. Furthermore, this result can be generalized to higher dimensions as the following: $\Phi = \Phi_x \otimes \Phi_y \otimes \Phi_z$ which allows for us to generalize Eq. 2.19 into a three-dimensional approximation space (x, y, z) or (ξ, η, ζ):

$$
a = (\Phi^T W \Phi)^{-1}\Phi^T f = \left\{\left[(\Phi_x^T \Phi_x)^{-1}\Phi_x^T\right] \otimes \left[(\Phi_y^T \Phi_y)^{-1}\Phi_y^T\right] \otimes \left[(\Phi_z^T \Phi_z)^{-1}\Phi_z^T\right]\right\} f \tag{16}
$$

Using the Kronecker product of three small least-squares operators to produce a larger operator allows us to approximate in higher dimensions by taking the cube root of the size of the three inverted matrices. Going up dimensions would require us only to introduce a fourth variable, which is the beauty of this Kronecker product trick. See the Appendix for more details about the properties of the Kronecker product and how lower dimensional analysis can be easily "upgraded" to higher dimensions. With this, we can generalize the above equation to n-dimensions by taking the n^{th} root of the inverted matrix dimension:

$$a = (\Phi^T W \Phi)^{-1}\Phi^T W f \equiv Cf \equiv (\bar{\Phi}^T\bar{\Phi})^{-1}\bar{\Phi}^T\bar{f} \equiv \bar{C}\bar{f} \tag{17}$$

where C is the Cholesky square root of $W^{1/2}$ of the weighted matrix $W = W^{1/2}W^{1/2}$. and $\bar{\Phi} \equiv W^{1/2}\Phi$, $\bar{f} = W^{1/2}f$ which makes Φ Kronecker factorable. This final result allows the same advantages to be used in the identity-weighted matrix case. With this, equation 2.20 can be expressed as:

$$a = (\bar{\Phi}^T\bar{\Phi})^{-1}\bar{\Phi}^T\bar{f} = \left\{\left[(\bar{\Phi}_x^T\bar{\Phi}_x)^{-1}\bar{\Phi}_x^T\right] \otimes \left[(\bar{\Phi}_y^T\bar{\Phi}_y)^{-1}\bar{\Phi}_y^T\right] \otimes \left[(\bar{\Phi}_z^T\bar{\Phi}_z)^{-1}\bar{\Phi}_z^T\right]\right\}\bar{f} \tag{18}$$

where $\bar{\Phi} \equiv W^{1/2}\Phi$, $\bar{\Phi}_x \equiv W^{1/2}\Phi_x$, $\bar{\Phi}_y \equiv W^{1/2}\Phi_y$, $\bar{\Phi}_z \equiv W^{1/2}\Phi_z$, and $\bar{f} \equiv W^{1/2}f$.

It is useful to note that W, W_x, W_y, and W_z have a Kronecker relationship: $W = W_x \otimes W_y \otimes W_z$ which allows for us to sample locations corresponding to the Chebyshev polynomials. The Kronecker product trick for constructing the two-dimensional Chebyshev least-squares matrix can be provided by:

$$\bar{C} \equiv (\bar{\Phi}^T\bar{\Phi})^{-1}\bar{\Phi}^T = (\Phi^T W \Phi)^{-1}\Phi^T W^{1/2} = CW^{-1/2} \equiv \bar{C}_x \otimes \bar{C}_y \tag{19}$$

where $\bar{C_x} = (\bar{\Phi}_x{}^T\bar{\Phi}_x)$. The most important takeaway from this step is that C is also Kronecker factorable, just as our weighted matrices W is. From the above conclusions, we can finally apply this formulation to n-dimensions. Take the following example for approximating a function in three dimensions:

$$\bar{C} = (\Phi^T W \Phi)^{-1}\Phi^T W = \bar{C_x} \otimes \bar{C_y} \otimes \bar{C_z}, W \equiv W_x \otimes W_y \otimes W_z, \bar{f} = W^{1/2}f \tag{20}$$

For our final generalized function, consider a function of n independent variables:

$$g(x_1, x_2, ..., x_n), \{x_{i,min} \leq x_i \leq x_{i,max}\}, i = 1, 2, ..., n \tag{21}$$

with the following forward and inverse functions:

$$\begin{aligned} &Forward\left\{\ \xi_i(x_i) = -1 + 2(x_i - x_{i,min})/(x_{i,max} - x_{i,min})\right. \\ &Inverse\left\{\ x_i(\xi_i) = x_{i,min} + \ (\xi_i + 1)\ (x_{i,max} - x_{i,min})/2\right. \end{aligned} \tag{22}$$

We can then write the function we wish to approximate by substituting Eqs. 2.26 into 2.25 using our independent variable (as we've done before):

$$f(\xi_1, \xi_2, ..., \xi_n) = f(\xi_i) = g(x_{i,min} + (\xi_i + 1)(x_{i,max} - x_{i,min})/2) \tag{23}$$

Similarly, for discrete measurements, we introduce a set of sample points (nodes): $\{\xi_{i,0}, \xi_{i,i}, ..., \xi_{i,Mx,i}; M_{xi} \geq N_{xi}\}$. This produces the following residual approximation error at each measurement node:

$$r_{i_k} = f(\xi_{1_k}, \xi_{2_k}, ..., \xi_{n_k}) - \sum_{\alpha_1=0}^{N_{x_1}}\sum_{\alpha_2=0}^{N_{x_2}}\cdots\sum_{\alpha_n=0}^{N_{x_n}} a_{\alpha_1\alpha_1...\alpha_n}\phi_{\alpha_1\alpha_1...\alpha_n}(\xi_{1_k}, \xi_{2_k}, ..., \xi_{n_k}) \tag{24}$$

where the k indices represent the measurement for the i^{th} variable. Equation 2.28 can written in vector-matrix notation, $r = f - \Phi a$. Finally, as mentioned in our two and three-dimensional examples, the weighted least-squares solution can be written as the Kronecker product of n smaller least-squares matrices:

$$a = (\Phi^T W \Phi)^{-1}\Phi^T W f \equiv \bar{C_{x1}} \otimes \bar{C_{x2}} \otimes ... \otimes \bar{C_{xn}} W^{1/2} f \tag{25}$$

where $\bar{C_{xi}} = C_{xi}W_{xi}^{-1/2}$, $W = W_{x1} \otimes W_{x2} \otimes ... \otimes W_{xn}$, and $a = vec\{a_{1k,2k,...nk}\}$, and $f = vec\{f_{1k,2k,...nk}\}$.

To conclude this section, it is good to note that the previously discussed methods of orthogonal approximation, i.e. the use of least-squares approximation with Chebyshev polynomials alongside the Kronecker product trick, will be implemented in our FEM approximation routine, specifically for the three-dimensional case in approximating Laplace's gravitational differential equations. Figures 2 and 3 summarize the steps taken in our approximation routine for the one, two, and n-dimension approximation routine.

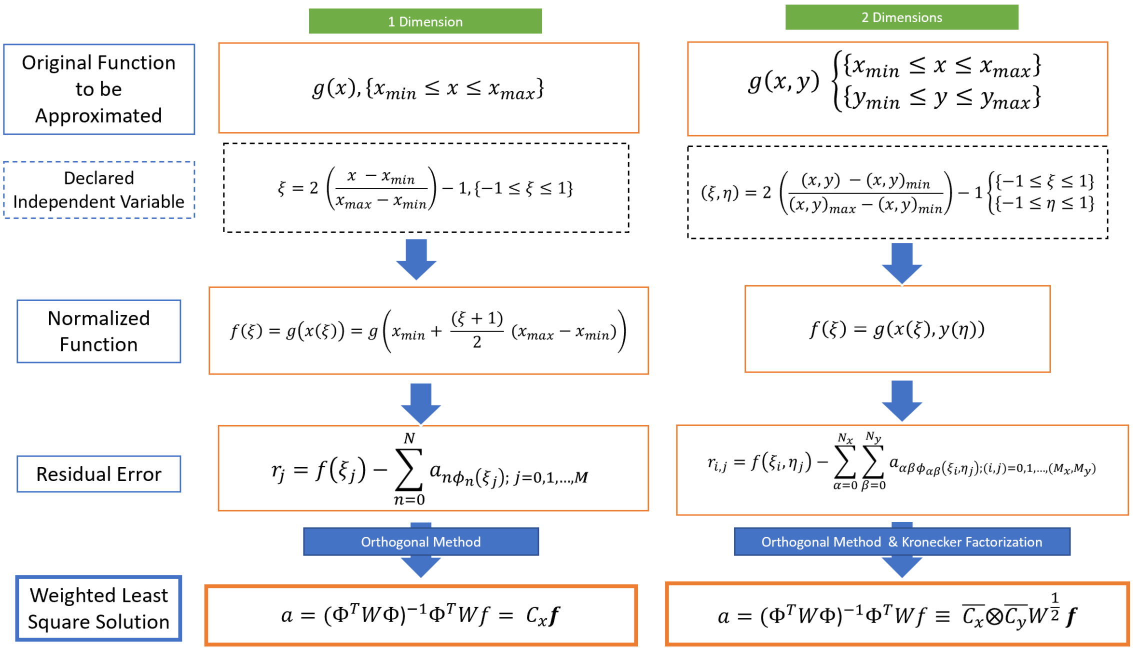

1 Dimension
2 Dimensions
Original Function to be Approximated
$g(x), \{x_{min} \leq x \leq x_{max}\}$
$g(x,y) \begin{cases} \{x_{min} \leq x \leq x_{max}\} \\ \{y_{min} \leq y \leq y_{max}\} \end{cases}$
Declared Independent Variable
$\xi = 2\left(\frac{x - x_{min}}{x_{max} - x_{min}}\right) - 1, \{-1 \leq \xi \leq 1\}$
$(\xi,\eta) = 2\left(\frac{(x,y) - (x,y)_{min}}{(x,y)_{max} - (x,y)_{min}}\right) - 1 \begin{cases} \{-1 \leq \xi \leq 1\} \\ \{-1 \leq \eta \leq 1\} \end{cases}$
Normalized Function
$f(\xi) = g(x(\xi)) = g\left(x_{min} + \frac{(\xi+1)}{2}(x_{max} - x_{min})\right)$
$f(\xi) = g(x(\xi), y(\eta))$
Residual Error
$r_j = f(\xi_j) - \sum_{n=0}^{N} a_n \phi_n(\xi_j);\ j=0,1,\ldots,M$
$r_{i,j} = f(\xi_i,\eta_j) - \sum_{\alpha=0}^{N_x}\sum_{\beta=0}^{N_y} a_{\alpha\beta}\phi_{\alpha\beta}(\xi_i,\eta_j);(i,j)=0,1,\ldots,(M_x,M_y)$
Orthogonal Method
Orthogonal Method & Kronecker Factorization
Weighted Least Square Solution
$a = (\Phi^T W \Phi)^{-1}\Phi^T W f = C_x \boldsymbol{f}$
$a = (\Phi^T W \Phi)^{-1}\Phi^T W f \equiv \overline{C_x} \otimes \overline{C_y} W^{\frac{1}{2}} \boldsymbol{f}$

Figure 2. Orthogonal approximation routine in one and two dimensions.

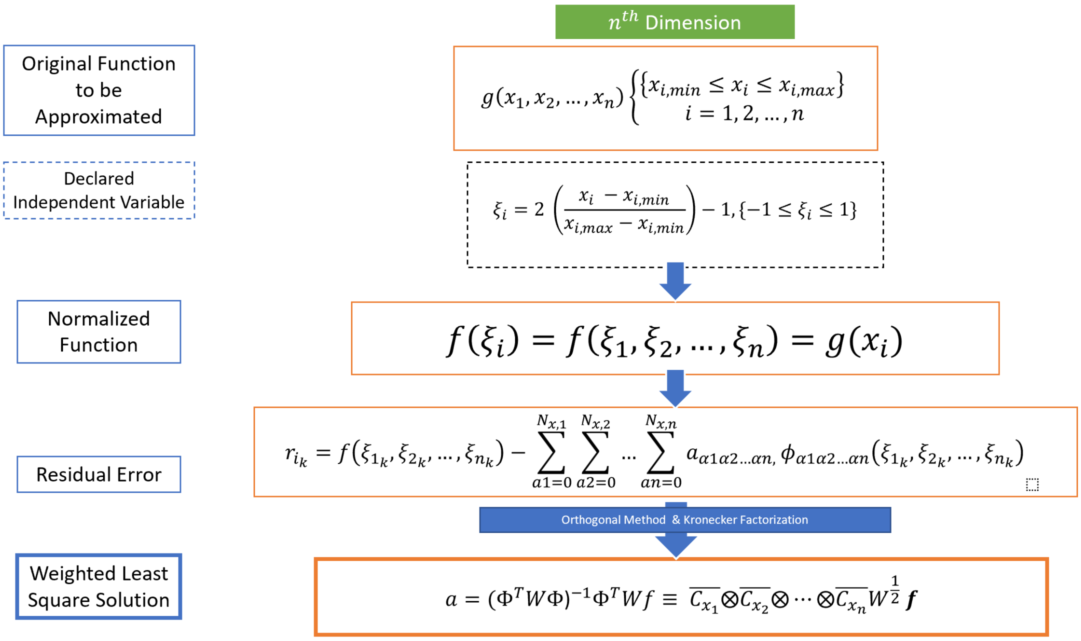

n^{th} Dimension
Original Function to be Approximated
$g(x_1, x_2, \ldots, x_n) \begin{cases} \{x_{i,min} \leq x_i \leq x_{i,max}\} \\ i = 1, 2, \ldots, n \end{cases}$
Declared Independent Variable
$\xi_i = 2\left(\frac{x_i - x_{i,min}}{x_{i,max} - x_{i,min}}\right) - 1, \{-1 \leq \xi_i \leq 1\}$
Normalized Function
$f(\xi_i) = f(\xi_1, \xi_2, \ldots, \xi_n) = g(x_i)$
Residual Error
$r_{i_k} = f(\xi_{1_k}, \xi_{2_k}, \ldots, \xi_{n_k}) - \sum_{\alpha 1=0}^{N_{x,1}} \sum_{\alpha 2=0}^{N_{x,2}} \cdots \sum_{\alpha n=0}^{N_{x,n}} a_{\alpha 1 \alpha 2 \ldots \alpha n,} \phi_{\alpha 1 \alpha 2 \ldots \alpha n}(\xi_{1_k}, \xi_{2_k}, \ldots, \xi_{n_k})$
Orthogonal Method & Kronecker Factorization
Weighted Least Square Solution
$a = (\Phi^T W \Phi)^{-1}\Phi^T W f \equiv \overline{C_{x_1}} \otimes \overline{C_{x_2}} \otimes \cdots \otimes \overline{C_{x_n}} W^{\frac{1}{2}} \boldsymbol{f}$

Figure 3. Orthogonal approximation routine in n-th dimensions.

3. Representations of Lunar Potential Using Orthogonal Finite Elements

In the following discussion, we consider a basic orthogonal FEM approximation to a gravitational acceleration model determined from NASA's lunar Prospector (LP) mission in 100 spherical harmonic degrees [46]. Released in 2000, the LP Gravity model allows considerations of an analogous FEM approximation of the 'LP100K' gravity model. For high-precision orbit propagation, we consider all terms used in the $(m, n) = (100, 100)$ LP model. Using this model, we solve Laplace's equation for gravity using FEM and compare it to the classical spherical harmonic approximations. Initially, a test was done to produce a contour plot of the disturbance gravity acceleration on a small degree order using our FEM approach. This test ensured a reasonable error value on top of a reasonable computation time which encouraged us to continue the procedure with a higher degree order for an accurate representation. Figure 4 shows the radial disturbance acceleration on the lunar surface from our FEM representation with (4×4) degree square using **N = 10** approximation.

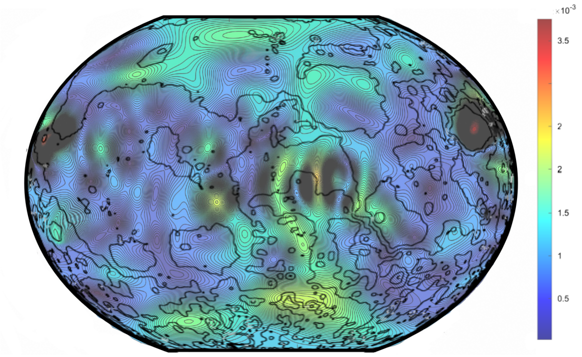


Figure 4. Radial FEM gravity approximation at the lunar Surface for LP (100 x 100) $(\mathrm{m\,s^{-2}})$.

As mentioned in the introduction, the three main challenges faced when using SH for orthogonal approximations are 1) An upper limit discrepancy, 2) inefficiency at orders higher than 2, and 3) Non-free singularities at the poles of the astronomical bodies. With these three challenges in mind, we introduce a Finite Element Method (FEM) model of surface gravity representations in the hopes that lower degree functions produced can be implemented to effectively and sufficiently model and compute lunar gravity. Furthermore, this method can be parallelized to increase computational efficiency. With this, we have to structure the FEM model to be both radially adaptive and efficient, which will result in better interpolation at higher degrees (i.e. at different radial distances above the surface of the Moon).

We will be considering the following gravitational potential model that is made up of the reference and disturbance gravity terms where the reference term includes two-body effects on the lunar gravity (U_{2B}) while the disturbance term ($\Delta U(r, \lambda, \phi)$) includes any other effects on gravity for the higher degree and order terms to take place. With these two terms, the finalized potential, $U(r, \lambda, \phi)$ can be given as:

$$U(r,\lambda,\phi) = \underbrace{U_{2B}(r,\lambda,\phi)}_{Reference\ Term} + \underbrace{\Delta U(r,\lambda,\phi)}_{Disturbance\ Term} \tag{26}$$

Furthermore, the acceleration can be derived from the potential and is represented as:

$$\ddot{r} = -\frac{\partial U}{\partial r} = \underbrace{\frac{\partial}{\partial r}\left\{\frac{GM}{r}\right\}}_{Reference\ Terms} - \underbrace{\frac{\partial}{\partial r}\Delta U}_{Disturbance\ Term} \tag{27}$$

It is important to note that adjusting the model to correct for the two-body interference will be done before we can consider the reference potential compact and efficient. This means only the perturbing acceleration terms are considered in our model.

3.1. Sampling: Radii and Shells

To help illustrate a sample FEM grid, consider a spherical grid with radius R_m (i.e. a sphere with the radius of the Moon at the lunar equator). This sphere is then covered by a 2-D mesh from $0 \leq \lambda \leq 360^o$ and $-88^o \leq \phi \leq 88^o$. It is good to note that the poles are neglected in this routine due to the limitations of the original SH model. As discussed, the poles of astronomical bodies are treated as singularities in SH modeling. However, the FEM model can be used to approximate the surface gravity at the poles using the Cartesian coordinates. This will be implemented in later approximation models. The following section will allow us to discuss the process of finding the most efficient initial conditions like shell size and radius sample.

3.1.1. Radial Sampling

At an arbitrary radius, r between $R_m \leq r \leq 1.11R_m$, let the gravity data$U(r,\lambda,\phi)$ be provided at a constant radius, r, on a specific spherical shell. This data on this shell should be transformed into $U(\xi,\zeta,\eta)$, where $-1 \leq \zeta \leq +1$ and $-1 \leq \eta \leq +1$, we can retrieve an accurate gravity modeling representation of $U(\zeta,\eta)$ on "sufficiently-dense" sets of concentric spherical surfaces. However, sampling radial coordinates is key when considering this unique variation.

To determine a proper radial sample, we consider two different sampling methods: **uniform sampling**, where we approximate the gravity at every radii between the lunar surface to 200 km above in 10 km increments, and cosine sampling. For **cosine sampling**, consider the transformed position r in which the position follows a "cosine-like" transformation as a function of the previously mentioned variable ξ as the following:

$$r = r_{min} + (r_{max} - r_{min})\left[1 - cos\left(\frac{\pi}{4}(1+\xi)\right)\right] \tag{28}$$

where

$$\xi = \frac{4}{\pi}cos^{-1}\left(1 - \frac{r - r_{min}}{r_{max} - r_{min}}\right) - 1 \tag{29}$$

where $-1 \leq \xi \leq +1$ and $r_{min} \leq r \leq r_{max}$, $r_{min} = R_m$ and $r_{max} = 1.11R_m$. At higher radii above the lunar surface, gravity anomalies decline rapidly. With this fact, we seek to format an efficient way to sample surface gravity that allows memory allocation. With cosine sampling, this transformation "naturally" constraints dense measurements at smaller radii ensuring that gravity perturbations are maximized and mimic how gravity would naturally act. This means that no matter the number of points N at lower radii, these nodes then bunch up in comparison to larger radii. This idea is illustrated in Figure 5 for 10 and 21 sample points as we compare this method to uniform sampling. To compact this progression, the transformed radius variable ξ was created to help treat r as an independent variable. It is again important to note that ξ is sampled non-uniformly using another cosine transformation in order to satisfy the Chebyshev orthogonality conditions:

$$\xi_i = -cos(i\pi/2M),\ i = 0,1,2,...,M \tag{30}$$

Not only does cosine sampling simulate a natural progression in gravity, but this sampling method also produces the error magnitude on the order of (10^{-9}) while taking the same amount of computational power as uniform sampling. For these reasons, we will continue using cosine sampling in our analysis at a latitude and longitude of λ = ϕ = 0^o degrees on the Moon.

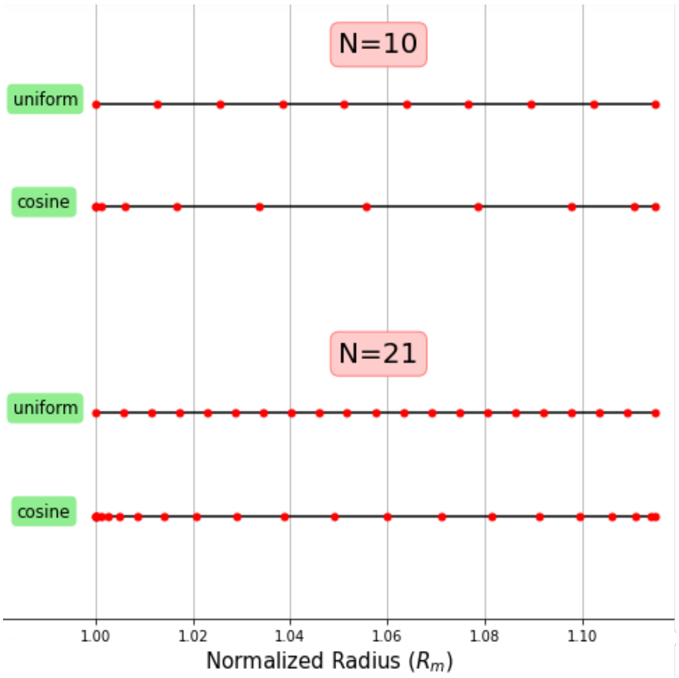


Figure 5. Comparison between uniform radius sampling and cosine radius sampling for 10 and 21 points.

3.1.2. Shells and Polynomial Order

After determining a proper radial sample, determining the required polynomial order as a function of radius is the next step. We want to determine this order that adaptively maintains an efficient error tolerance. The required acceleration error tolerance is determined from the polynomial order N at every radius. For example, if we consider a 4-by-4-degree square area on the lunar surface, the convergence error is the maximum absolute error between the LP SH model's acceleration values and the approximated acceleration from our method.

Consider a shell size of (10×10) square degrees compared to a smaller size of (2×2) square degrees. The (2×2) shell size would require less computational power/time to compute all the coefficients over a small area, while the (10×10) shell size would require a longer computational time to reach a similar error tolerance because of the larger number of coefficient values calculated. The compromise of time would be better justified with the resulting error produced. Although the larger shell size may take a longer period of time to produce results, if these results display a small error tolerance (10^{-9}m s^{-1}), it would be best to use this. Thus, when determining the correct shell size for our model, we want to pursue a shell size that maintains an efficient error tolerance while not compromising computational power.

With a maximum degree order of 100 and shell sizes of ((4×4) and (6×6), we evaluated both sizes and checked for accuracy and computational time across a portion of the near side of the lunar surface (from $0 \leq \lambda \leq 180^o$ and $0^o \leq \phi \leq 88^o$). As discussed previously, the smaller shell size of (4×4) would be more desirable if the computational time is reasonable. However, if the accuracy for the (6×6) case is reasonable compared to the (4×4) shell case, this shell size will be considered. Figure 5 displays a graph of computational time and calculation error as a function of shell size. For a shell size of (4×4) as maximum degree order of 100, it took the machine (see machine configurations in Section V) approximately 3627.43 seconds to produce the relevant FEM approximation coefficients with an error of 6.16×10^{-10}m s^{-1}. For the (6×6) case, it took the same machine approximately 10653.11 seconds to produce the FEM approximation coefficients with an error of 1.40×10^{-16}m s^{-1}. With the shell size of (6×6) providing such a small error compared to typical errors found in other FEM surface gravity models produced in [21], we found that this error is insignificantly smaller given the increased computational time. With this, we have concluded that the (4×4) case provides the best balance between computational time and reasonable accuracy.

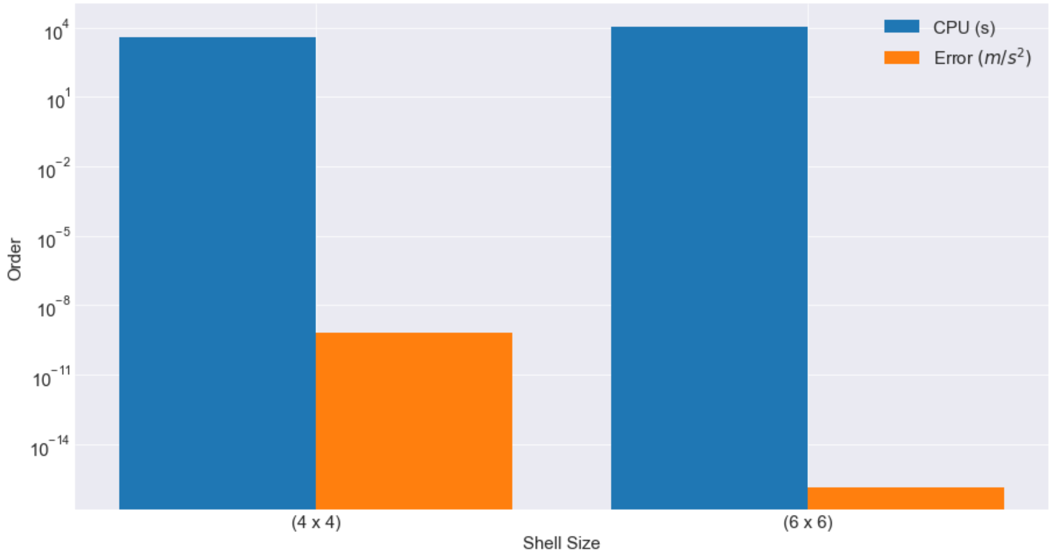


Figure 6. Comparison between shell sizes of (4×4) and (6×6) square degrees.

3.2. Orthogonal Approximation of the Gravitational Acceleration

Using orthogonal approximation, we estimated the lunar gravitational acceleration by transforming gravity into sample points. These sample points are located in our distribution in the least-squares approximation. In this subsection, we will briefly discuss our approach.

Let gravity data $U(r, \lambda, \phi)$ transform to $U(\xi_i, \zeta_j, \eta_k)$ with sample points located as a cosine distribution. For the three components of gravity, the approximation is as follows:

$$\begin{gathered}
g_x(\xi_i, \zeta_j, \eta_k) \cong \sum_{\alpha=0}^{N_x} \sum_{\beta=0}^{N_y} a_{\alpha\beta}(\xi_i)\phi_{\alpha\beta}(\zeta_j, \eta_k) = \Psi_x^T(\zeta_j, \eta_k) a_x(\xi_i) \\
g_y(\xi_i, \zeta_j, \eta_k) \cong \sum_{\alpha=0}^{N_x} \sum_{\beta=0}^{N_y} a_{\alpha\beta}(\xi_i)\phi_{\alpha\beta}(\zeta_j, \eta_k) = \Psi_y^T(\zeta_j, \eta_k) a_y(\xi_i) \\
g_z(\xi_i, \zeta_j, \eta_k) \cong \sum_{\alpha=0}^{N_x} \sum_{\beta=0}^{N_y} a_{\alpha\beta}(\xi_i)\phi_{\alpha\beta}(\zeta_j, \eta_k) = \Psi_z^T(\zeta_j, \eta_k) a_z(\xi_i)
\end{gathered} \tag{31}$$

where the indices (i, j, k) range from 0 to (M_r, M_λ, M_ϕ), and $\Psi_{x,y,z}$ are the basis functions (which are transposed) of the acceleration components.

Furthermore, Equation 32 can be expressed more compactly if the same basis function for the acceleration components by using $\Psi = \Psi_x = \Psi_y = \Psi_z$ which provides us with the following relation: $\Psi = \Psi_x \otimes \Psi_y \otimes \Psi_z$. We then utilize the tensor product of $\Psi(\eta_j, \zeta_k)$ and the identity matrix, I_n, twice to "undo" the transpose acceleration components. This creates the finalized gravitational acceleration approximation equation:

$$g(\xi_i, \zeta_j, \eta_k) = \begin{bmatrix} g_x(\xi_i, \zeta_j, \eta_k) \\ \\ g_y(\xi_i, \zeta_j, \eta_k) \\ \\ g_z(\xi_i, \zeta_j, \eta_k) \end{bmatrix} = [(\Psi(\zeta_j, \eta_k) \otimes I_n) \otimes I_n] \begin{bmatrix} a_x(\xi_i) \\ a_y(\xi_i) \\ a_z(\xi_i) \end{bmatrix} \tag{32}$$

To use the FEM for computing gravitational acceleration, we employ the maximum polynomial order, ensuring consistency levels as compared to the overall model. In particular, at larger radii above the lunar surface, the computational speed decreases by an order of magnitude when only non-zero terms are included.

3.2.1. Parallelization

Recent developments have seen the parallelization of gravitational FEM models, as implemented in [22], utilizing the Open Multi-Processing (OpenMP) application interface, which supports shared-memory parallel programming in C/C++ and Fortran. In this project, the inputs are the spherical coordinates (r, ϕ, λ) with corresponding normalized parameters ξ, ζ, and η which are used to evaluate the Chebyshev polynomials. The master thread is then split into slave threads, each evaluating the multivariate Chebyshev polynomials in parallel. These threads are then joined back together to be evaluated for the FEM acceleration by adding spherical and J_2 terms. A three-magnitude speedup was reported using OpenMP on a geopotential FEM model with 8 threads. Even with a more efficient computation time, the parallelized FEM model still produced less than 10 digits of accuracy compared to the original SH model. With this in mind, OpenMP implementation to the FEM lunar model promises higher speedup gains if more threads are used. This study indicates potential efficiency gains in both memory allocation and computation time for future FEM lunar model developments.

3.3. Results of the FEM LP Analysis

The following discussion delves into the results derived from the above-mentioned FEM routine applied NASA's LP100K SH model. The primary focus is on determining the best parameters for achieving an optimal balance between computational efficiency and error minimization.

For cosine sampling, the number of CGL nodes was highest near the surface and would decrease as the radii increased. The number of sample nodes should decrease as nodes are related to the number of coefficients used in the polynomial approximation at a specific radius. In order to find the best node quantity for each radius, we iterate our routine through uniform node values at each radius until a node value produces an error greater than a tolerance of 10^{-9} m s^{-2}. Because the number of CGL nodes is related to the polynomial order, we want to further our investigation by observing how the polynomial order affects computational time. With this, we find an analogous relationship between CPU time and polynomial order (as expected). Figure 7 illustrates a decrease in time cost as a function of radius alongside of the polynomial order decreasing in the same manner confirming the idea that it takes a longer period of time to approximate at higher polynomial orders than at lower orders.

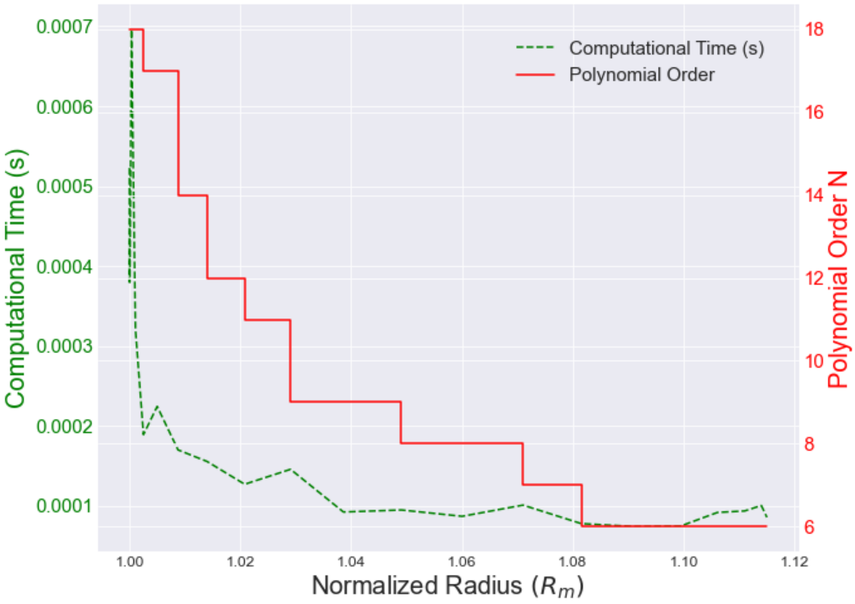


Figure 7. CPU and Polynomial Order vs. Radius.

For higher precision orbit propagation, we consider all terms used in the $(m, n) = (100, 100)$ LP spherical harmonic model. As mentioned in our approximation theory, we select an error tolerance of 10^{-9} m s^{-2} while substituting the high-degree order gravity model with a FEM approximation. Figure 8 showcases the maximum absolute error of the approximated norm of the "disturbance" acceleration as a function of the Chebyshev polynomial order N. Here, we note that as the radius increases, the error declines faster; a smaller polynomial order is sufficient enough to reach the placed error tolerance of 10^{-9}m s^{-2}. However, in this plot, we can see that there are two error tolerance values displayed. In our previous analysis, we identified a polynomial order of 21 that yielded an error of 10^{-9} m s^{-2}. However, after reviewing the coefficient values in this polynomial order, we found that the portion of corresponding errors was much smaller than our allowed error tolerance. This suggests that this polynomial order was significantly implementing a high precision that was computationally expensive. With this, we implemented a "fetching" routine where any error values less than our error tolerance had their corresponding coefficient value to be approximated to 0.

As an outcome, most of the coefficient values were approximated to 0, further suggesting an excessive polynomial order. After revisiting our analysis and the polynomial order of 18 was found for the surface, the same "fetching" routine was done with $N = 18$, finding that none of the error values strayed too far past the error tolerance, which was apparent in the weight of non-zero coefficient values found in that sample. This led us to believe that a maximum polynomial order of 18 was the most reasonable (computationally supportive and precise) option for the surface of the Moon. An error tolerance of 10^{-9}m s^{-2} proved to be too conservative. With this, a more liberal error tolerance value of 10^{-8}m s^{-2} was placed to ease the amount of computational burden while also still maintaining reasonable error.

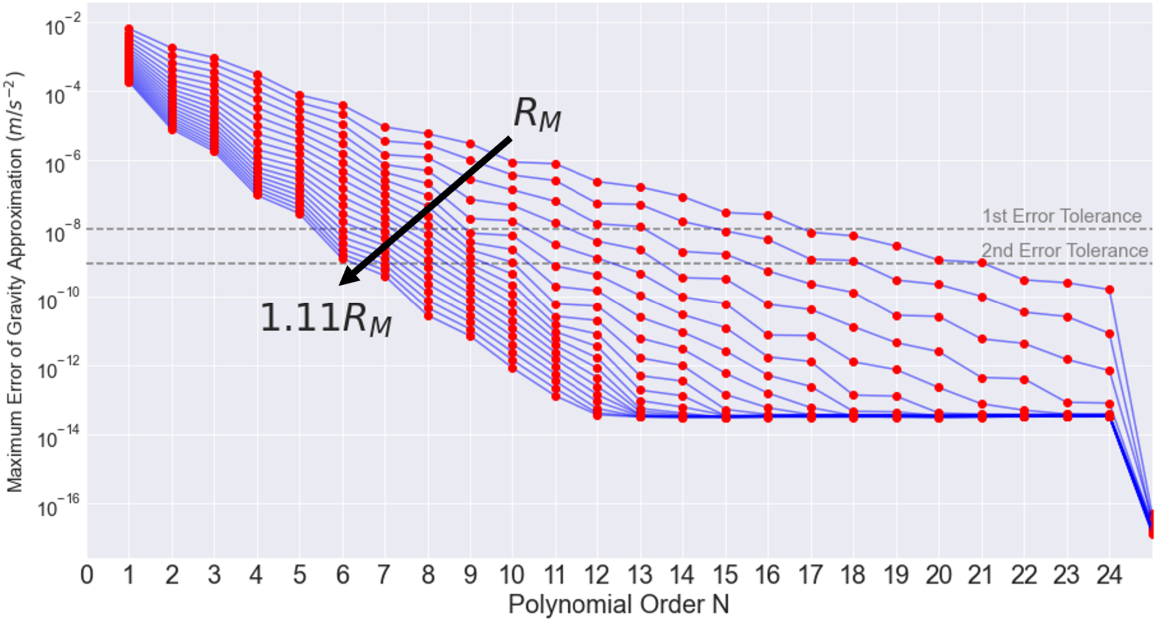
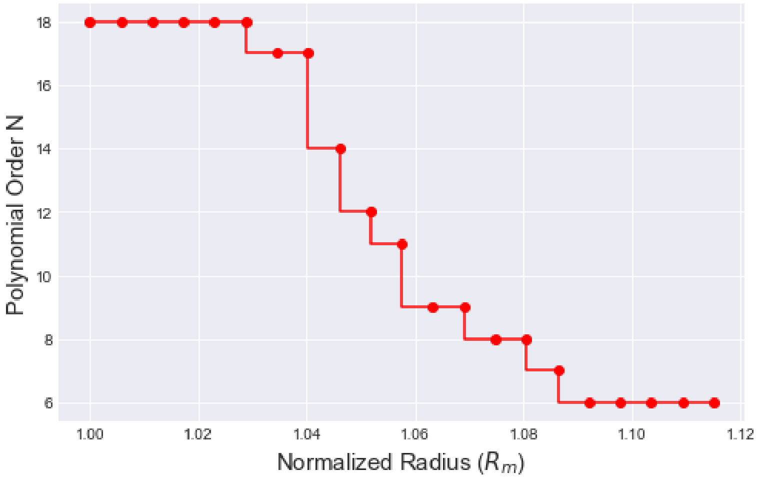


Figure 8. LP Maximum Error of Chebyshev FEM Gravity Approximation (m s^{-2}) versus Polynomial Order (N), for radial distances from $r_{min} = R_M$ to $r_{max} = 1.11R_m$, 200 km above the surface.

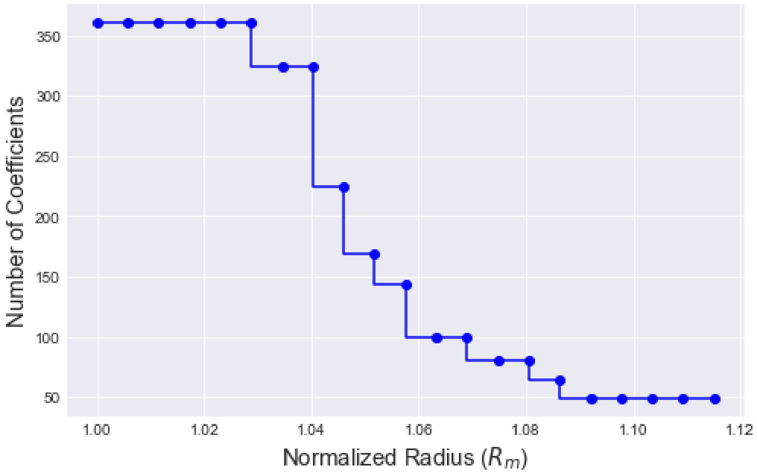
As we get closer to the surface of the Moon, a higher polynomial order is required to reach this error tolerance. The (100, 100) LP model's gravitational acceleration values are used as the truth and the errors between the FEM approximation are reduced by adjusting the degree value of the Chebyshev polynomials until it reaches the error tolerance. For our lunar model, we found that the most efficient maximum approximation error is 10^{-9} m s^{-2} which provides a fair balance between computational time and machine precision at a maximum polynomial order of **N = 18** on the lunar surface with a (4×4) square degree FEM cell size. However, when a polynomial order of 18 was applied to a radius

100 km above the lunar surface, we find that the error produced is negligible. This confirmed our previous discovery about a decreasing radial polynomial order i.e. as you go further from the surface, the polynomial order required to reach the error tolerance decreases as well. Furthermore, a Maximum degree value (M) of 100 and a shell value of (4×4) degree maintains an efficient error tolerance while constraining sensible computational power.

The determination of the correct polynomial order for each radial adaptation allowed for us to determine the number of coefficients required in approximation at each radius. At the lunar surface, the polynomial order of $N = 18$ indicates that the gravity acceleration at the lunar surface can be approximated by $(18 + 1)^2 = 361$ orthogonal polynomial terms (or coefficients). At 200 km above the surface, a polynomial order of $N = 6$ was found to be most efficient, which means it requires 49 coefficient terms. This suggests that as the polynomial order increases, the number of coefficients required in the approximation also increases as well. This relationship can also be supported when observing how an error is related to the node number at each radius. Figure 9 displays two plots; (a) polynomial order as a function of radius and (b) number of coefficients as a function of radius; these plots show a nearly similar result.



(a) Polynomial Order as a function of radii



(b) Number of coefficients as a function of radii

Figure 9. Plots of Polynomial Order N and Number of coefficients as a function of normalized radius R_m.

On the same machine, the FEM model displays three orders of magnitude of computational speedup at the lunar surface compared to the (100, 100) LP SH model. This can be seen in Figure 10.

As we approach this comparison radially, the FEM approach continues to be computationally efficient compared to the original (100, 100) SH model, even going as far as to decrease computational time as it reaches higher radii. At radii R near $(150km + R_m) < R < (200km + R_m)$, we find that the FEM approaches five orders of magnitude of computational efficiency compared to the SH model. If implemented for orbit propagation, the FEM gravity model should not only provide a 9 to 10-digit of accuracy, but it will also produce a 3-5 order of magnitude reduction in CPU time.

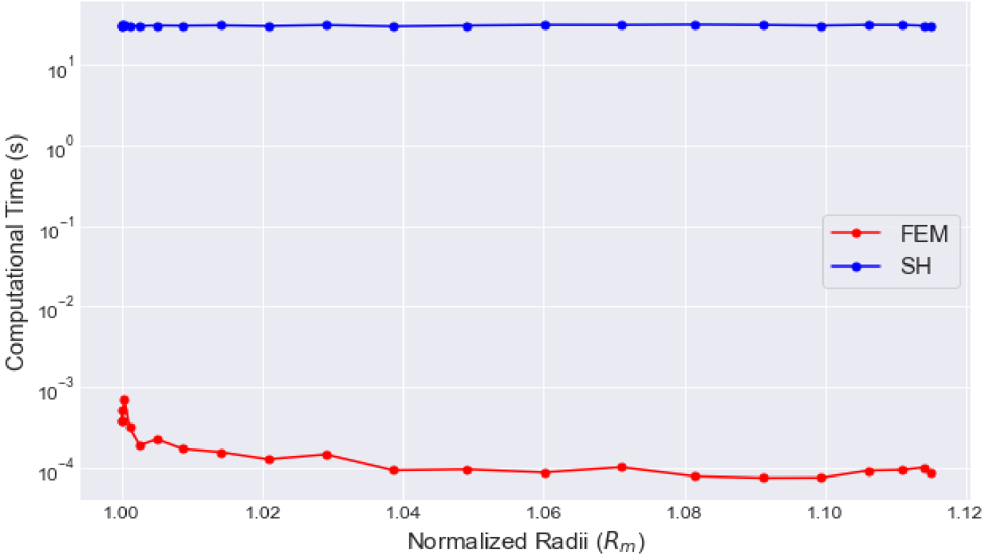


Figure 10. Computational speed of Spherical Harmonic method (blue) vs. FEM (red).

The Chebyshev coefficients were saved for each radial iteration. These first 49 coefficient terms are displayed for our final cosine sampling Figure 13. With cosine sampling, higher-order non-zero coefficients have been found to be comparable to easy-to-fit smooth functions of similar nature. With this, it is observed that the norm of gravity decays rapidly, as it should. Figure 11 qualitatively displays a prompt decay of gravity anomalies in a finite element where the contours created by the gravitational acceleration perturbation at three radii located at the surface, 100 km, and 200 km above the surface. These coefficients can then be used to approximate orbital propagation at different radii on or above the lunar surface. When considering the coefficients to be a function of radius, we can use the two-dimensional approximations on each surface to promote three dimensions by representing each coefficient as an orthogonal Chebyshev function of radius. As a result, an average surface gravity acceleration value of $1.624 \pm 7.626 \times 10^{-9}$ m s^{-2} was found for a polynomial order of 18 and at 200 km above the surface of the Moon, an average value of $0.5017 \pm 1.36595 \times 10^{-9}$ m s^{-2}. These results are illustrated in Figure 12 for all computed radii.

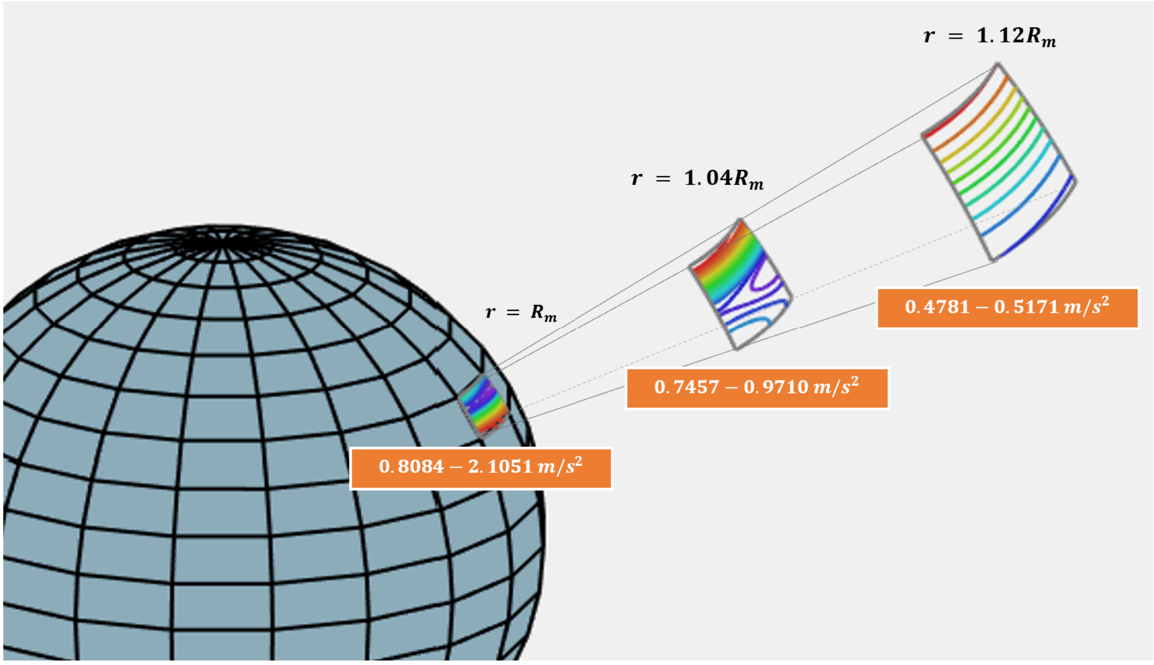


Figure 11. Gravitational acceleration $(\mathrm{m\,s^{-2}})$ from the lunar surface to 200 km above surface.

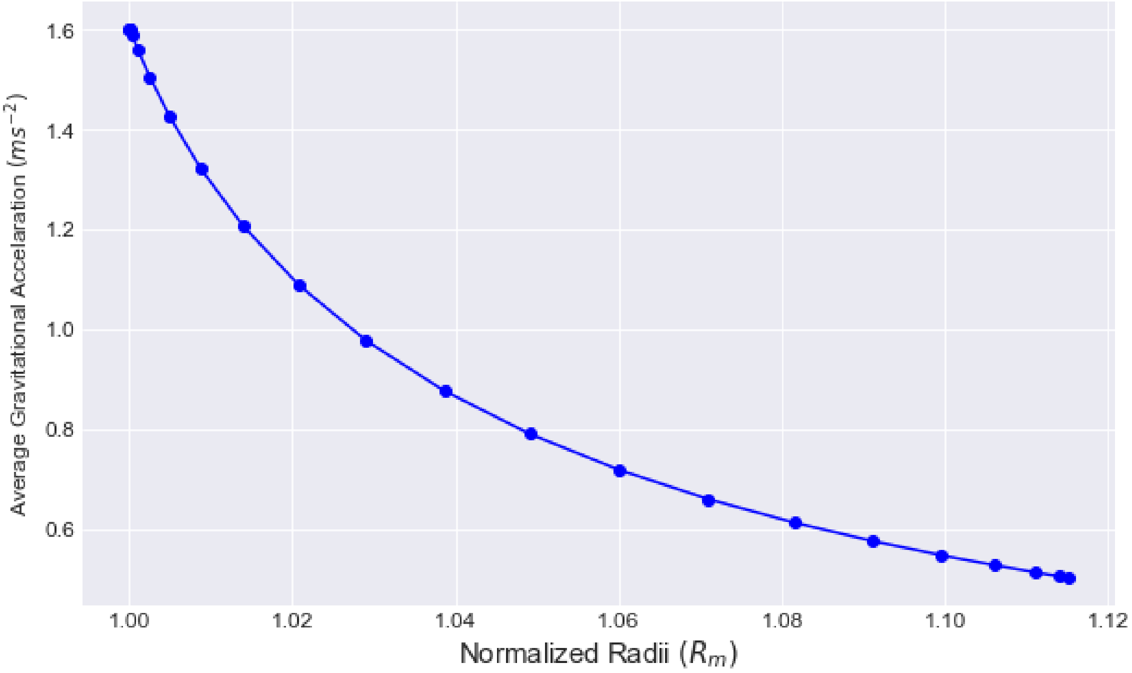
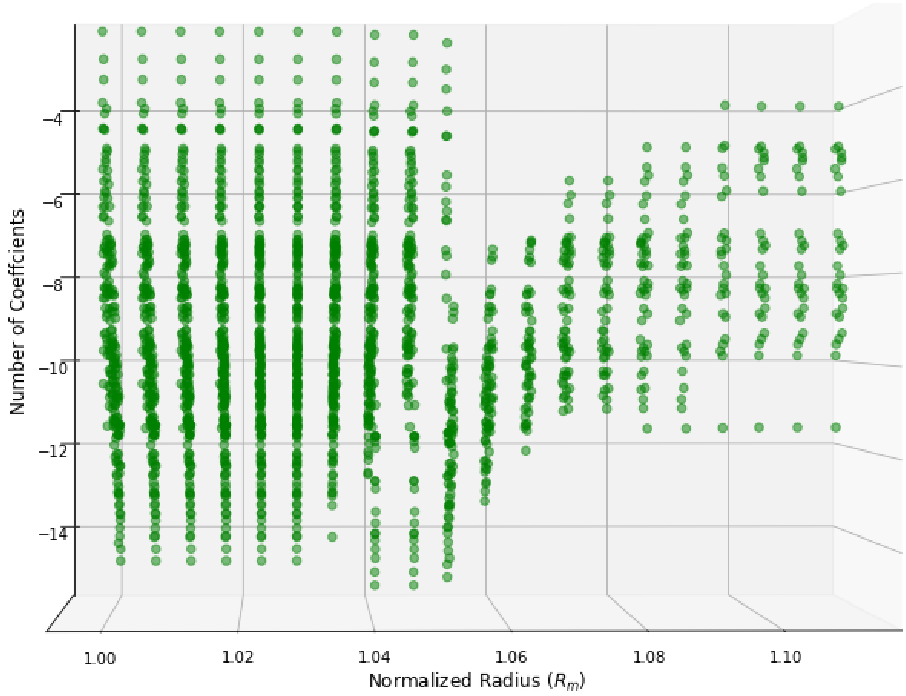
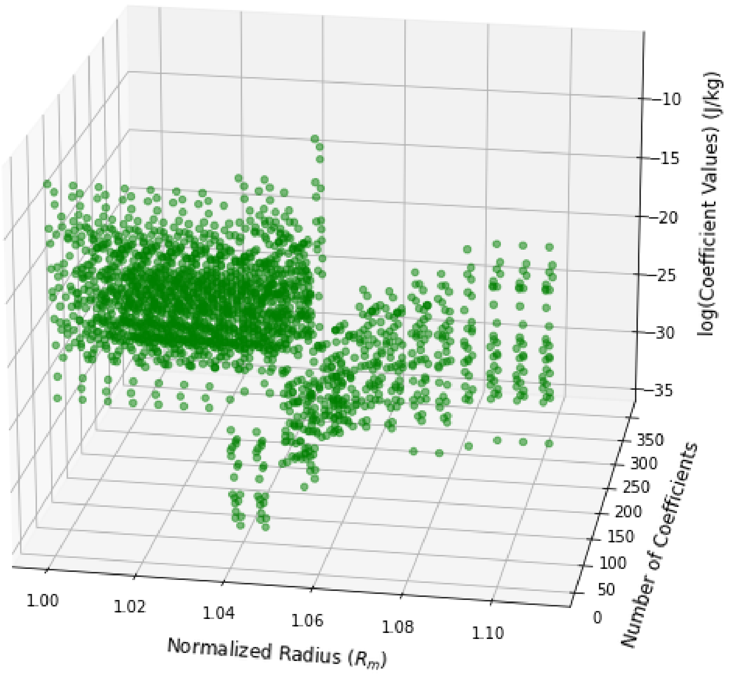


Figure 12. Average gravitational acceleration $(\mathrm{m\,s^{-2}})$ from the lunar surface to 200 km above surface.



(a) Cosine Sampling - Top view: log of coefficient value as a function of normalized radius



(b) Cosine Sampling: Number of coefficients and log of coefficient value as a function of normalized radius

Figure 13. Coefficient representation as a function of normalized radius for Cosine sampling.

4. FEM Application in Astrodynamics: Satellite Trajectory Propagation

In this section, we will discuss how the FEM can be applied to orbit trajectory propagation, and then we will examine a case study to demonstrate how the coefficients produced from our FEM lunar model can aid in satellite trajectory propagation. Finally, we will compare this result to an SH routine to analyze computational time and precision.

Modeling spacecraft trajectories has been one of the most important fields in astrodynamics and space guidance. Satellite orbit propagation predicts how an object will move based on its current state. The satellite's position is defined by a group of approximate equations of motion where the degree of approximation is dependent on orbital information. A trajectory is the path a spacecraft follows through space as it is influenced by the gravitational force of one or more bodies, creating acceleration. The two-body model describes of a satellite with negligible mass orbiting a larger astronomical body. The equations of motion for this model can be solved, resulting in different conic sections depending on their distance away from the larger body. Although not represented in the two-body model, it is worth noting that when considering a satellite orbiting the Moon, we would need to consider the gravity of Earth. Introducing an "n-th" body example is to be investigated in future works.

In the equation of motion, the larger astronomical body is treated as fixed within the two-body frame. We then consider the body as a uniform point mass to further simplify the equation of motion. As stated previously, the spacecraft has negligible mass compared to the main body, which means its acceleration is only due to the gravity of the main body.

The following equation represents the two-body problem we implemented using our FEM coefficients (represented by the disturbance acceleration):

$$\ddot{\vec{r}} = -\frac{\mu}{r^3}\vec{r} + \sum_{i=0}^{k} a_d^i \tag{33}$$

where μ is the gravitational constant of the Moon provided by the original 'LP100K' model: $\mu = 4.902800238 \times 10^{12}\ \mathrm{m\,s^{-2}}$ alongside the disturbance acceleration where a_d^i is the i^{th} coefficient of k gravity spherical harmonic perturbation terms. These terms were previously saved from the original FEM model.

As mentioned previously, we can approximate the position of a spacecraft at any point in its orbit as long as we have at least one reference point to use as a starting approximation. Numerical methods of orbit propagation, also termed special perturbations, compute and approximate solutions of general equations of motion [47–49]. Figure 14 illustrates a general approach to satellite trajectory propagation provided initial conditions. In the following sections, we seek to implement the coefficients acquired from the FEM routine done on a small patch of (4×4) square degrees on various radii on and from the surface of the Moon to interpolate the trajectory of a point source satellite orbiting the Moon. Finally, we check the accuracy and computational time to that of the original LP (100 x 100) SH model. Orbital propagation using FEM has been implemented in a series of projects, some of which can be seen here: [50–52].

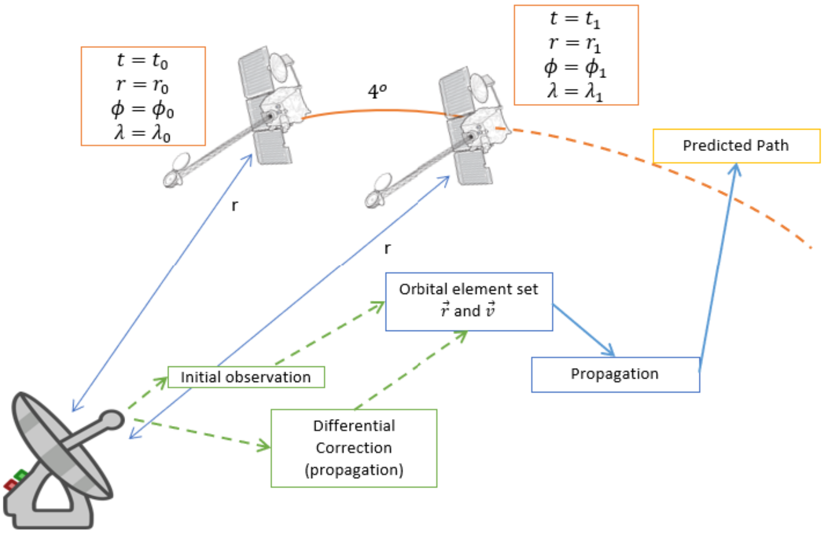


Figure 14. Representation of orbit determination.

4.1. Trajectory Propagation Using FEM versus LP (100 x 100) SH Gravity Model

In this section, we compare the trajectory propagation results achieved using the coefficient values found from our FEM model to the classical SH method. Computations for the following examples were performed using a machine with the following configurations:

- Intel(R) Xeon(R) CPU 3.70GHz 3.70 GHz, 16.0GB of RAM
- Windows 64-bit operating system, x64-based processor
- MATLAB R2019a

Our final goal in this experiment is to justify that the coefficients produced by the original FEM routine can be used in various satellite orbit propagation problems. The steps needed to take for our orbit propagation experiment are the following:

1. **Produce a set of initial values**: ϕ_o, λ_o, and R. These values represent the spacecraft's azimuth, elevation, and radius from the lunar surface, respectively, at $t = 0$.
2. **Set an "ending" position or time.** This can be done by providing ϕ_{final}, λ_{final} or providing a time span.
3. **Produce a small trajectory across a small shell.** Using the coefficient values saved at a particular radius from our previous FEM routine to calculate the acceleration of the spacecraft along this initial trajectory using both the FEM and SH approaches. At this point, we will then compare the resulting computational time and error between the two methods.
4. **Generate satellite positions.** Using MATLAB's "ode45" function to solve the differential two-body problem represented as Equation 4.4, we feed in the acceleration values calculated from step 3, and a time span to find the spacecraft's position as a function of time.
5. **Generate satellite velocities.** To do this, we will implement the MATLAB function "fsolve" to solve the systems of nonlinear equations of several values to solve for the satellite's velocity over the course of the small shell provided initial velocities of V_{x_0}, V_{y_0}, V_{z_0}, the spacecraft's position, and the time span.
6. **Generate a long-span satellite orbit trajectory.** This will be done by, again, utilizing the "ode45" function to generate satellite positions at a longer time span and the resulting final velocities.

Before determining the entire trajectory of a spacecraft, we first have to simulate a smaller portion of its trajectory based on three different radii above the lunar surface (100 km, 150 km, and 200 km). This small trajectory will span over the (4 x 4) square degrees shell at this radius. In this analysis, we consider three tests on orbit propagation. These tests differ in how we produce the initial values:

1. **Test 1**: Randomized position changes. Using MATLAB's "rand" function, we randomize a uniformly distributed pseudorandom set of values for $d\phi$ and $d\lambda$, i.e. the changes in azimuth and elevation, respectively. Using these randomized values, we create two ranges for ϕ and λ:

$$\begin{gathered}[\phi_{min} = 0^o < \phi + d\phi < \phi_{max} = 4^o] \\ [\lambda_{min} = 0^o < \lambda + d\lambda < \lambda_{max} = 4^o]\end{gathered} \tag{34}$$

   With these simulated ranges and a radius 100 km above the lunar surface, we can then convert these values into Cartesian coordinates to get our starting x, y, and z positions of the simulated spacecraft..
2. **Test 2**: A Monte Carlo simulation of initial conditions. With this approach, we repeated randomized values for $d\phi$ and $d\lambda$ under the conditions that $\phi_{min} < \phi_{max}$ and $\lambda_{min} < \lambda_{max}$. This produces various amounts of ranges for ϕ and λ in which we seek to use a randomly picked set of ranges to achieve the initial positions of the simulated spacecraft.

doi:10.20944/preprints202409.0081.v1

3. **Test 3**: Set starting and final positions. This approach is the most simple and straightforward as it is reflective of the most basic problems in orbit propagation. We start of with converting $\phi_o = 0^o$, $\lambda_o = 0^o$, and R into the satellite's initial Cartesian coordinate: x_0, y_0, and z_0. Unlike the previous two approaches, in this test, the satellite's final position is found by converting $\phi_{final} = 4^o$, $\lambda_{final} = 4^o$, and R into the satellite's final Cartesian coordinate: x_{final}, y_{final}, and z_{final}. All that is left to do is to find the satellite's positions during $t = 0s$ and $t = t_{final}$.

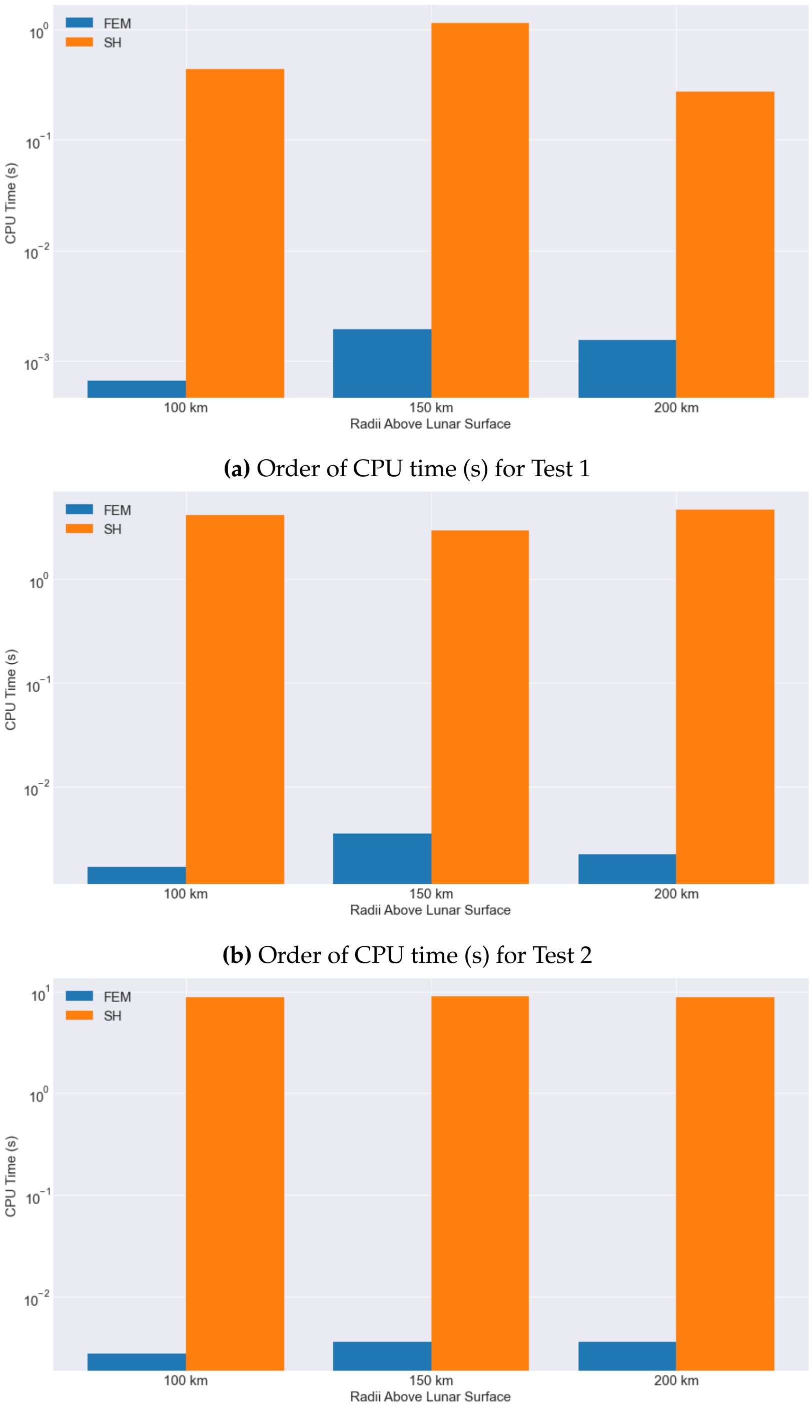
For all three approaches, we use the initial ranges of ϕ and λ and R to generate satellite accelerations using our FEM routine and the LP (100×100) SH routine. In the FEM analysis, the coefficients saved from our original model at $R = 100\,km + R_m$ were multiplied by the Chebyshev polynomials approximated at each elevation and azimuth. These results are then compared to SH results for computational time and error. Figure 15 displays the computational time for both approaches for each of the three tests for each of the three radii. For all three tests, FEM timing is 3–4 orders of magnitude smaller than that of SH.

For all three tests, we calculate the FEM model's error by subtracting it from the "truth" value which we have specified before to be the LP (100×100) model. The resulting errors are illustrated in Figure 16. In Tests 1 and 2, the resulting FEM error from all three radii is in the order of 10^{-10} m s^{-2}. With the previously mentioned increase in computational speedup, we find that these two approaches produced above-adequate results for approximating the acceleration beginning produced in the (4×4) square degree shell 100 km above the lunar surface. However, for the Monte Carlo simulation (Test 2), we find a higher order of points in the iteration compared to Test 1, which is reflected in the computational time required for each of the tests. Figure 17 displays the initial trajectories created from these three tests over the (4×4) square degree section. In Test 1, the trajectory is smooth and laminar, which is expected from a small spacecraft trajectory, while Test 2 displays a randomized path which is representative of a typical Monte Carlo simulation. It is good to note that the trajectory found in Test 2 does not suggest that the model failed to produce a correct spacecraft orbit. This proves that the model was able to approximate a disorderly trajectory, as apparent in the small error.

The error found in Test 3 is in the order of 10^{-9} m s^{-2} across all three radii. It is also good to note that the error is lowest at the "end" points of the calculation due to the initial values set at each starting position. As illustrated in Figure 17c, the trajectory across the (4×4) square degree shell is similar to that of the trajectory for Test 1; however, this Test produced more data points.

Finally, a long space orbit propagation was done for all three tests. For this, a time span of 360 minutes was used. Figure 18 shows the long-span predicted trajectory of the satellite as apparent from Test 3 for a radius of 100 km above the lunar surface as an example. This long-span trajectory is shown in red, while the initial small shell trajectory is in blue. The shape of this conic section is representative of typical trajectories found with two-body applications: [53–55].

This application of FEM shows promising results and demonstrates a potential path to significant advancements in computational speedup and precision for differential equations in astrodynamics.

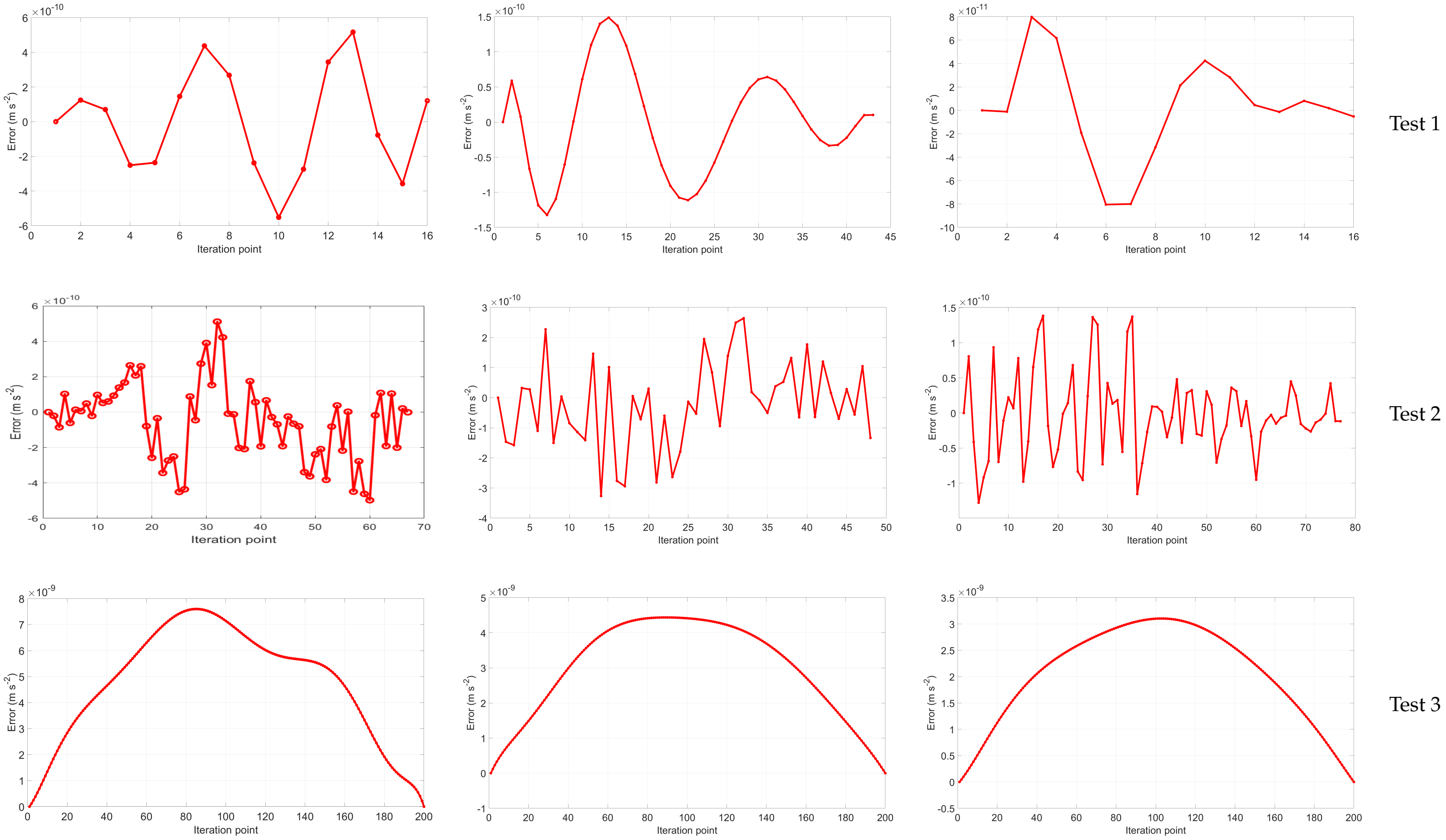


(a) Order of CPU time (s) for Test 1

(b) Order of CPU time (s) for Test 2

(c) Order of CPU time (s) for Test 3

Figure 15. Computational time for the three tests for FEM and LP (100 x 100) SH approaches for each radius above the lunar surface.

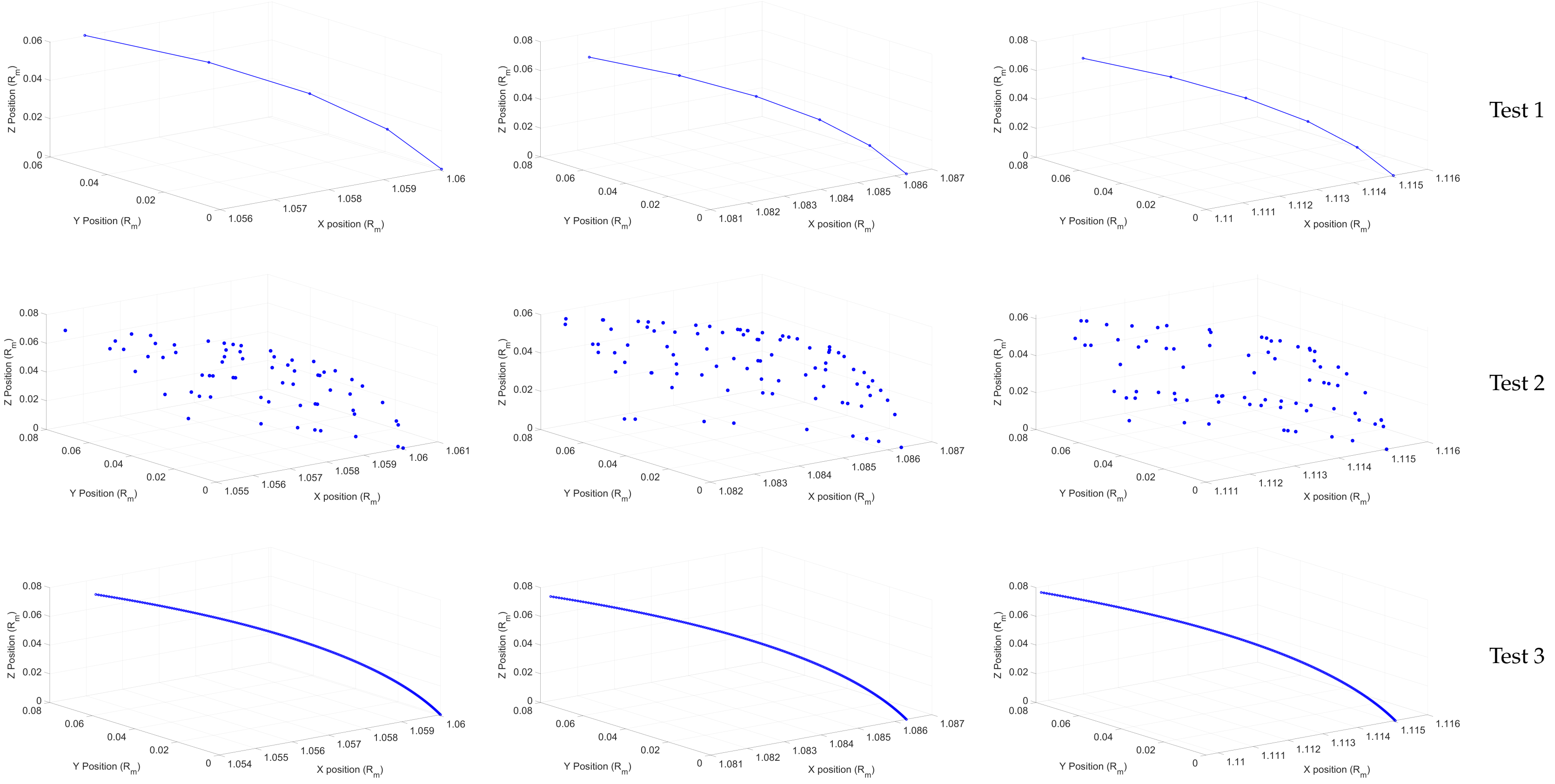


(a) 100 km

(b) 150 km

(c) 200 km

Figure 16. FEM error for three orbital propagated initial positions tests.



(a) 100 km (b) 150 km (c) 200 km

Figure 17. Spacecraft Trajectory across (4 x 4) square degrees for three tests.

doi:10.20944/preprints202409.0081.v1

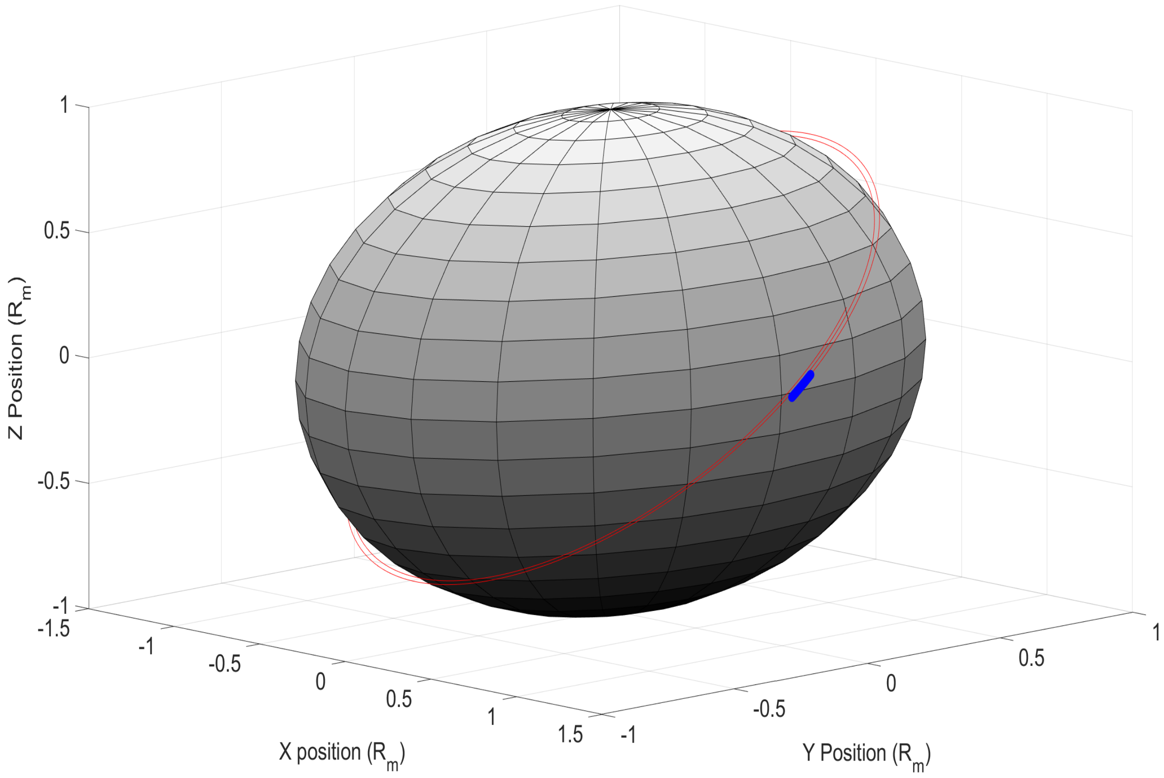


Figure 18. Long Span Satellite Orbit Trajectory Approximated by FEM for 100 km above the surface.

5. Conclusion

This research presented a novel approach for modeling the Moon's surface gravity anomaly, enhancing approximation techniques for Laplace's differential equations of gravitational forces. Using the orthogonal Finite Element Method (FEM), we approximated a gravitational acceleration model based on data from NASA's Lunar Prospector mission encompassing 100 spherical harmonic degrees. The FEM model was then compared to the classical SH solution in two regards: accuracy and computational expense.

Our findings revealed the efficiency of orthogonal approximation techniques, including modifying the regression matrix, to be Kronecker factorable when approximating higher dimensions using algebraic identities on the one-dimension orthogonal least-squares operators. The use of Chebyshev polynomials in our approximation strategy has also allowed the selection of specific orthogonal basis functions, which supports an efficient sampling mechanism.

We further evaluated two radial sampling techniques and established that cosine sampling alleviated the challenges associated with the Runge Phenomenon. Placing a conservative error tolerance of 10^{-9} m s^{-2} on the resulting disturbance acceleration, we were able to save the corresponding coefficient values for reaching radial iteration from the lunar surface to 200 km above. Through our FEM technique, we found a correlation between the number of coefficients resolved in each radial iteration and the polynomial order. For instance, with a polynomial order N=18N=18, we deduced 361 orthogonal polynomial terms on the lunar surface. As a result, an average surface gravity acceleration value of $1.624 \pm 7.626 \times 10^{-9}$ m s^{-2} was found for a polynomial order of 18 and at 200 km above the surface of the Moon, an average value of $0.5017 \pm 1.36595 \times 10^{-9}$ m s^{-2}.

A notable highlight was the computational speedup achieved using the FEM approach. We showed that a five-order of magnitude speedup in the computational time can be obtained using the FEM model compared to the original SH approach. As a consequence, a push to apply these techniques on precise long-term orbit propagation for spacecraft and satellites in lunar-based missions has been made. Three numerical orbital tests were conducted for a satellite orbiting the Moon. In all three tests, the coefficient values produced from the original FEM model were used to approximate the satellite's trajectory as it goes across a (4 x 4) square shell 100 km above the surface. These accelerations showed small errors [$10^{-6} - 10^{-10}$ m s^{-2}] across all three tests with a four-order of magnitude speedup in computation time. A long-span trajectory was also achieved using FEM for one of the tests, displaying promising results in the interpolation of orbital mechanics representing the two-body problem.These findings are crucial for precise orbit propagation in various astrodynamics simulations.

Anticipating future endeavors, this research paves the way for advancements in orthogonal approximation theory and orbit propagation. Potential avenues include code parallelization for upgrades in computation efficiency, FEM machine learning benchmarks, implementing n-body orbital propagation for satellites with treated masses, and FEM modeling of the entire lunar surface for more precision in constraining the disturbance gravitational acceleration. With this, FEM surface gravity models can be developed for other spherical and non-spherical astronomical bodies such as Mars, Europa, and comets whose surface gravity anomaly is not well modeled or studied. Overall, this method is well-versed and shows extreme assurance for the realization of new developments in the field of Astronomy and Astrodynamics.

Acknowledgments: This work was supported by San Diego State University's Aerospace Engineering and Astronomy departments. We would like to thank Prof. John Junkins for his tremendous support and insight as he pioneered FEM gravity modeling for us.

References

1. NASA, Goddard & Arizona State University. *Lunar Reconnaissance Orbiter Wide Angle Camera*; NASA, 2011.
2. Drake, N. *A rogue rocket part collided with the Moon*, National Geographic, 2022.
3. Chin, G.; Brylow, S.; Foote, M.; Garvin, J.; Kasper, J.; Keller, J.; Litvak, M.; Mitrofanov, I.; Paige, D.; Raney, K.; Robinson, M.; Sanin, A.; Smith, D.; Spence, H.; Spudis, P.; Stern, S.A.; Zuber, M. Lunar Reconnaissance Orbiter Overview: The Instrument Suite and Mission. *Space Science Reviews* **2007**, *129*, 391–419. doi:10.1007/s11214-007-9153-y.
4. James, P.; Ermakov, A.; Sori, M. Requirements for gravity measurements on the anticipated Artemis III mission, 2020. doi:10.48550/ARXIV.2009.03514.
5. Goossens, S.; Sabaka, T.; Wieczorek, M.; Neumann, G.; Mazarico, E.; Lemoine, F.; Nicholas, J.; Smith, D.; Zuber, M. High-Resolution Gravity Field Models from GRAIL Data and Implications for Models of the Density Structure of the Moon's Crust. *Journal of Geophysical Research: Planets* **2020**, *125*. doi:10.1029/2019JE006086.
6. Konopliv, A.S.; Asmar, S.W.; Carranza, E.; Sjogren, W.L.; Yuan, D.N. Recent Gravity Models as a Result of the Lunar Prospector Mission. *Icarus* **2001**, *150*, 1–18. doi:10.1006/icar.2000.6573.
7. Blitzer, L.; Weisfeld, M.; Wheelon, A.D. Perturbations of a Satellite's Orbit Due to the Earth's Oblateness. *Journal of Applied Physics* **1956**, *27*, 1141–1149. doi:10.1063/1.1722218.
8. Bills, B.G.; Ferrari, A.J. A harmonic analysis of lunar gravity. *Journal of Geophysical Research: Solid Earth* **1980**, *85*, 1013–1025, [https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/JB085iB02p01013]. doi:https://doi.org/10.1029/JB085iB02p01013.
9. Ishihara, Y.; Namiki, N.; Sugita, S.; Matsumoto, K.; Goossens, S.; Araki, H.; Noda, H.; Sasaki, S.; Iwata, T.; Hanada, H. Localized Gravity/Topography Correlation and Admittance Spectra one the Moon. EGU General Assembly Conference Abstracts, 2009, EGU General Assembly Conference Abstracts, p. 3927.
10. Konopliv, A.S.; Binder, A.B.; Hood, L.L.; Kucinskas, A.B.; Sjogren, W.L.; Williams, J.G. Improved Gravity Field of the Moon from Lunar Prospector. *Science* **1998**, *281*, 1476–1480, doi:10.1126/science.281.5382.1476.
11. Casotto, S.; Fantino, E. Evaluation of methods for spherical harmonic synthesis of the gravitational potential and its gradients. *Advances in Space Research* **2007**, *40*, 69–75. doi:10.1016/j.asr.2007.01.021.
12. Jancaitis, J.R.; Junkins, J.L. Modeling in n dimensions using a weighting function approach. *Journal of Geophysics Research* **1974**, *79*, 3361–3366. doi:10.1029/JB079i023p03361.
13. Klipstein, W.M.; Arnold, B.W.; Enzer, D.G.; Ruiz, A.A.; Tien, J.Y.; Wang, R.T.; Dunn, C.E. The Lunar Gravity Ranging System for the Gravity Recovery and Interior Laboratory (GRAIL) Mission. *Space Science Reviews* **2013**, *178*, 57–76. doi:10.1007/s11214-013-9973-x.
14. Balmino, G.; Vales, N.; Sylvain, B.; Briais, A. Spherical harmonic modelling to ultra-high degree of Bouguer and isostatic anomalies. *Journal of Geodesy* **2011**, *86*, 499–520. doi:10.1007/s00190-011-0533-4.

15. Rexer, M.; Hirt, C. Ultra-high-Degree Surface Spherical Harmonic Analysis Using the Gauss-Legendre and the Driscoll/Healy Quadrature Theorem and Application to Planetary Topography Models of Earth, Mars and Moon. *Surveys in Geophysics* **2015**, *36*, 803–830. doi:10.1007/s10712-015-9345-z.
16. Goossens, S.; Matsumoto, K.; Liu, Q.; Kikuchi, F.; Sato, K.; Hanada, H.; Ishihara, Y.; Noda, H.; Kawano, N.; Namiki, N.; Iwata, T.; Lemoine, F.G.; Rowlands, D.D.; Harada, Y.; Chen, M. Lunar gravity field determination using SELENE same-beam differential VLBI tracking data. *Journal of Geodesy* **2011**, *85*, 205–228. doi:10.1007/s00190-010-0430-2.
17. Junkins, J.L. Investigation of Finite-Element Representations of the Geopotential. *AIAA Journal* **1976**, *14*, 803–808, [https://doi.org/10.2514/3.61420]. doi:10.2514/3.61420.
18. Fahroo, F.; Ross, I.M. Direct Trajectory Optimization by a Chebyshev Pseudospectral Method. *Journal of Guidance, Control, and Dynamics* **2002**, *25*, 160–166, [https://doi.org/10.2514/2.4862]. doi:10.2514/2.4862.
19. Gong, Q.; Fahroo, F.; Ross, I.M. Spectral Algorithm for Pseudospectral Methods in Optimal Control. *Journal of Guidance, Control, and Dynamics* **2008**, *31*, 460–471, [https://doi.org/10.2514/1.32908]. doi:10.2514/1.32908.
20. Tang, X.; Chen, J. Direct Trajectory Optimization and Costate Estimation of Infinite-horizon Optimal Control Problems Using Collocation at the Flipped Legendre-Gauss-Radau Points. *IEEE/CAA Journal of Automatica Sinica* **2016**, *3*, 174–183. doi:10.1109/JAS.2016.7451105.
21. Bani Younes, A.H.A. Orthogonal polynomial approximation in higher dimensions: Applications in Astrodynamics. PhD thesis, Texas A & M University, 2013.
22. Atallah, A.M.; Younes, A.B. Parallel Evaluation of Chebyshev Approximations: Applications in Astrodynamics. *Journal of the Astronautical Sciences* **2022**. doi:10.1007/s40295-022-00302-6.
23. Engels, R.C.; Junkins, J.L. Local Representation of the Geopotential by Weighted Orthonormal Polynomials. *Journal of Guidance and Control* **1980**, *3*, 55–61, [https://doi.org/10.2514/3.55947]. doi:10.2514/3.55947.
24. Junkins, J.L.; Miller, G.W.; Jancaitis, J.R. A weighting function approach to modeling of irregular surfaces. *Journal of Geophysical Research (1896-1977)* **1973**, *78*, 1794–1803, doi:10.1029/JB078i011p01794.
25. Beylkin, G.; Cramer, R. Toward Multiresolution Estimation and Efficient Representation of Gravitational Fields. *Celestial Mechanics and Dynamical Astronomy* **2002**, *84*, 87–104.
26. Lekien, F.; Marsden, J. Tricubic interpolation in three dimensions. *International Journal for Numerical Methods in Engineering* **2005**, *63*, 455–471, [https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.1296]. doi:10.1002/nme.1296.
27. Colombi, A.; Hirani, A.N.; Villac, B.F. Adaptive Gravitational Force Representation for Fast Trajectory Propagation Near Small Bodies. *Journal of Guidance, Control, and Dynamics* **2008**, *31*, 1041–1051, [https://doi.org/10.2514/1.32559]. doi:10.2514/1.32559.
28. Hujsak, R. Almost-Optimal Sensor Tasking using Auction Methods. Advanced Maui Optical and Space Surveillance Technologies Conference; Ryan, S., Ed., 2010, p. E29.
29. Oltrogge, D. *AstroHD: Astrodynamics Modeling With a Distinctly Digital Flavor*; AIAA, 2008; https://doi.org/10.2514/6.2008-7065.
30. Boni, L.; Bassetto, M.; Mengali, G.; Quarta, A. Electric sail static structural analysis with finite element approach. *Acta Astronautica* **2020**, *175*. doi:10.1016/j.actaastro.2020.06.009.
31. Sophal Pou, L.; Garcia, R.; Mimoun, D.; Murdoch, N.; Karatekin, O. Determination of the gravitational potential and tidal stress of asteroids using the finite element method. EGU General Assembly Conference Abstracts, 2018, EGU General Assembly Conference Abstracts, p. 4755.
32. Zheng, G.; Nie, H.; Chen, J.; Chuanzhi, C.; Lee, H. Dynamic analysis of lunar lander during soft landing using explicit finite element method. *Acta Astronautica* **2018**, *148*. doi:10.1016/j.actaastro.2018.04.014.
33. Atallah, A.; Bani Younes, A. *Parallel Finite Element Gravity Model*; AIAA, 2020; doi:10.2514/6.2020-1465.
34. Atallah, A.; Bani Younes, A. *A Comparative Study of Orbit Propagation Using Clenshaw-Curtis and Gauss-Legendre Quadrature Methods*; AIAA, 2021; doi:10.2514/6.2022-2511.
35. Atallah, A.; Bani Younes, A. *Correction: A Comparative Study of Orbit Propagation Using Clenshaw-Curtis and Gauss-Legendre Quadrature Methods*; AIAA, 2022; [https://arc.aiaa.org/doi/pdf/10.2514/6.2022-2511.c1]. doi:10.2514/6.2022-2511.c1.
36. Bai, X. Modified chebyshev-picard iteration methods for solution of initial value and boundary value problems. PhD thesis, Texas A & M University, 2011.
37. Battin, R.H. *An introduction to the mathematics and methods of astrodynamics.*; AIAA, 1987.

38. Chebyshev, P.L. *"Theorie De Mecanismes Connus Sous Le Nom De Paralllogrammes*; M'emoires des Savants 'etrangers pr'esent'es 'a l'Acad'emie de Saint P'etersbourg, 1857.
39. Grant, J.A. Chebyshev Polynomials in Numerical Analysis. By L. Fox and I. B. Parker. Pp. ix, 205. 42s. 1968. (Oxford.). *The Mathematical Gazette* **1970**, *54*, 96–97. doi:10.2307/3613223.
40. Mason, J.C.; Handscomb, D.C. *Chebyshev Polynomials*; Chapman and Hall/CR, 2002.
41. Schaub, H.; Junkins, J.L. *Analytical Mechanics of Space Systems*; AIAA, 2003. doi:10.2514/4.861550.
42. Singla, P.; Junkins, J. *Multi-resolution methods for modeling and control of dynamical systems*; Chapman and Hall/CRC, 2008; pp. 1–299.
43. Snay, R.A. *Applicability of array algebra*; U.S. Dept. of Commerce, National Oceanic and Atmospheric Administration, National Ocean Survey, 1978; p. 459–461.
44. Crassidis, J.L.; Junkins, J.L. *Optimal Estimation of Dynamic Systems, Second Edition (Chapman & Hall/CRC Applied Mathematics & Nonlinear Science)*, 2nd ed.; Chapman and Hall/CRC, 2011.
45. MATLAB. *version 7.10.0 (R2010a)*; The MathWorks Inc.: Natick, Massachusetts, 2010.
46. Konopliv, A.S.; Yuan, D.N. Lunar Prospector 100th Degree Gravity Model Development. Lunar and Planetary Science Conference, 1999, Lunar and Planetary Science Conference, p. 1067.
47. Danby, J. *Fundamentals of celestial mechanics, Richmond: Willman-Bell*; Willmann-Bell, Inc., 1992.
48. Montenbruck, O.; Gill, E.; Lutze, F. *Satellite orbits: models, methods, and applications, Appl. Mech. Rev., 55, 2, B27-B28*; Springer, 2002.
49. Vallado, D.A. *Fundamentals of astrodynamics and applications*; Vol. 12, Springer Science & Business Media, 2001.
50. Bradley, B.K. Numerical Algorithms for Precise and Efficient Orbit Propagation and Positioning. PhD thesis, University of Colorado at Boulder, 2015.
51. García-Pérez, A.; Sorribes-Palmer, F.; Alonso, G.; Ravanbakhsh, A. FEM simulation of space instruments subjected to shock tests by mechanical impact. *International Journal of Impact Engineering* **2019**, *126*, 11–26. doi:10.1016/j.ijimpeng.2018.12.008.
52. Read, J.; Bani Younes, A.; Junkins, J.; Turner, J. State Transtion Matrix for Perturbed Orbital Motion Using Modified Chebyshev Picard Iteration. *Journal of the Astronautical Sciences* **2015**.
53. Branco, C.A. Conical Orbital Mechanics: A Rework of Classic Orbit Transfer Mechanics. PhD thesis, Old Dominion University, 2020.
54. Cheng, Z.Y.; Yan, L.S. The Area Factors of the Conic Section and Its Application on Orbit Determination of Visual Binary. New Frontiers in Binary Star Research; Leung, K.C.; Nha, I.S., Eds., 1993, Vol. 38, *Astronomical Society of the Pacific Conference Series*, p. 410.
55. Trenti, M.; Hut, P. Gravitational N-body Simulations, 2008. doi:10.48550/ARXIV.0806.3950.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.