Normalization of Gravitational Acceleration Models

Randy A. Eckman[1]
Aaron J. Brown[2]
Daniel R. Adamo[3]

January 24, 2011

[1]Aerospace engineering co-op, NASA Johnson Space Center
[2]Aerospace engineer, NASA Johnson Space Center, Mail Code DM34
[3]Independent astrodynamics consultant

THIS PAGE IS INTENTIONALLY LEFT BLANK.

Contents

i

List of Tables

THIS PAGE IS INTENTIONALLY LEFT BLANK.

List of Figures

THIS PAGE IS INTENTIONALLY LEFT BLANK.

Chapter 1

Introduction

Unlike the uniform density spherical shell appoximations of Newton, the consequence of spaceflight in the real universe is that gravitational fields are sensitive to the nonsphericity of their generating central bodies. The gravitational potential of a nonspherical central body is typically resolved using spherical harmonic approximations. However, attempting to directly calculate the spherical harmonic approximations results in at least two singularities which must be removed in order to generalize the method and solve for any possible orbit, including polar orbits. Three unique algorithms have been developed to eliminate these singularities by Samuel Pines [1], Bill Lear [2], and Robert Gottlieb [3].

This paper documents the methodical normalization of two[1] of the three known formulations for singularity-free gravitational acceleration (namely, the Lear [2] and Gottlieb [3] algorithms) and formulates a general method for defining normalization parameters used to generate normalized Legendre Polynomials and ALFs for any algorithm. A treatment of the conventional formulation of the gravitational potential and acceleration is also provided, in addition to a brief overview of the philosophical differences between the three known singularity-free algorithms.

---

[1]The Pines algorithm (Section 4.1) has been previously normalized and thoroughly investigated by Lundberg and Schutz [4] and subsequently implemented by DeMars [5]. See Section 4.1.2.

THIS PAGE IS INTENTIONALLY LEFT BLANK.

Chapter 2

Gravitational Potential and Acceleration

2.1 Coordinates

Gravitational potential V is typically resolved using an equatorial central-body-fixed Cartesian coordinate system $\begin{bmatrix} x_b & y_b & z_b \end{bmatrix}^T$, where

- x_b passes through the center of mass, equator, and prime meridian of the central body

- z_b is the north polar axis of the central body

- y_b completes the right-handed triad, where y_b is positive in the eastern hemisphere

Spherical coordinates $\begin{bmatrix} r & \theta & \phi \end{bmatrix}^T$ can then be defined in terms of the central-body-fixed coordinates

- $r = \sqrt{x_b^2 + y_b^2 + z_b^2} = $ magnitude of position vector

- $\theta = \arctan2(y_b, x_b) = $ (positive east) longitude or right ascension

- $\phi = \arcsin(\frac{z_b}{r}) = $ (positive north) latitude or declination

If $x_b = y_b = 0$, then θ may be set to any value. For convenience, it is typically set to $\theta = 0$ in this case. Similarly, if $r = 0$, then ϕ may be set to any value but is typically set to $\phi = 0$.

The relationship between central-body-fixed and spherical coordinates is shown in Figure 2.1. The spherical coordinate angles themsevles are rarely used directly since they typically appear as arguments in trigonometric functions. See Section 2.1.2 for more information.
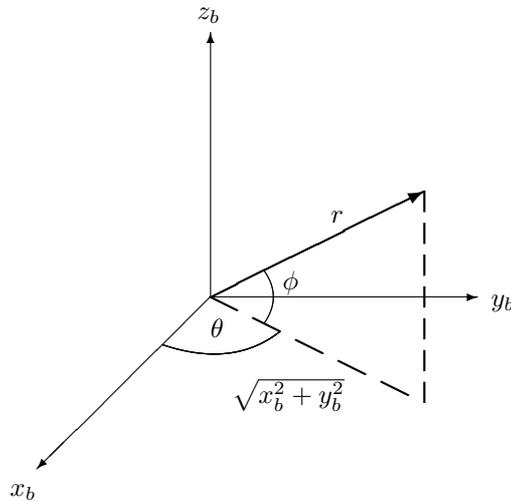
Figure 2.1: Body-fixed coordinates and spherical coordinates illustrated

2.1.1   The Gravitational Acceleration

The gradient of the gravitational potential V is the force of gravity per unit mass, which is the *acceleration.*

The gravitational acceleration in the central-body-fixed Cartesian coordinate system $\begin{bmatrix} x_b & y_b & z_b \end{bmatrix}^T$ is

$$a = \begin{bmatrix} a_{x_b} & a_{y_b} & a_{z_b} \end{bmatrix}^T = \nabla_b V = \begin{bmatrix} \frac{\partial V}{\partial x_b} & \frac{\partial V}{\partial y_b} & \frac{\partial V}{\partial z_b} \end{bmatrix}^T \qquad (2.1)$$

where ∇_b is the gradient with respect to the central-body-fixed coordinates.

Taking the gradient with respect to a central-body-fixed (i.e. accelerating) coordinate system means that we must include the Coriolis, centripetal, tangential, and relative accleration terms [6, pg. 54-55]. To eliminate the complication of including these terms, an alternative inertial (non-accelerating) coordinate system centered in the central body is used to compute the gradient.

Consider a displacement $\Delta\theta$ of the position vector in spherical coordinates in the direction of increasing θ (away from the x_b axis). While displacement of θ is actually the arc of a circle, a limit is approached as $\Delta\theta \rightarrow 0$ when taking the gradient in spherical coordinates and thus the displacement arc becomes arbitrarily close to its subtending chord. The instantaneous positive displacement of θ for a vector in spherical coordinates is thus perpendicular to the radial vector in the direction of increasing θ. This logic can be similarly applied for ϕ. The directions of increasing θ and ϕ form the basis of the new coordinate system [7, pg. 143].

The new x axis is parallel to the position vector and is defined as the r_o axis. The new y axis is located on the central-body-fixed $x_b y_b$ plane at a right-
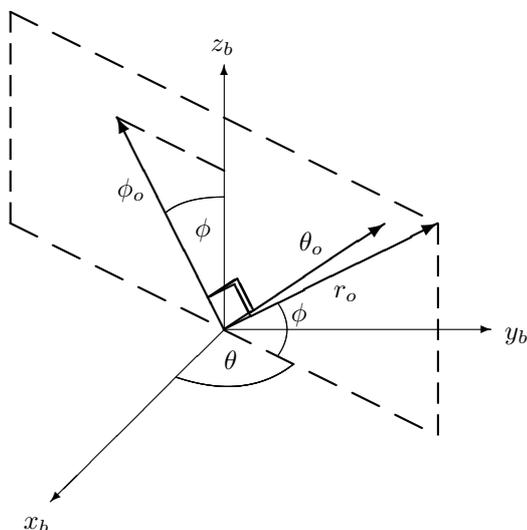
Figure 2.2: Orthogonal spherical coordinates $\begin{bmatrix} r_o & \theta_o & \phi_o \end{bmatrix}^T$ and central-body-fixed coordinates $\begin{bmatrix} x_b & y_b & z_b \end{bmatrix}^T$ shown with the $r_o\phi_o$ plane

angle to the r_o axis in the direction of increasing θ and is defined as the θ_o axis. The new z axis completes the right-handed triad and is defined as the ϕ_o axis. The ϕ_o axis forms the angle ϕ with the z_b axis in the direction of increasing ϕ (away from the z_b axis). The new coordinates are referred to as the *orthogonal spherical coordinates* $\begin{bmatrix} r_o & \theta_o & \phi_o \end{bmatrix}^T$, not to be confused with the conventional spherical coordinates $\begin{bmatrix} r & \theta & \phi \end{bmatrix}^T$. The relationship between central-body-fixed coordinates and orthogonal spherical coordinates is shown in Figure 2.2.

The transformation matrix from the orthogonal spherical coordinates to the central-body-fixed coordinates is [2]

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = \begin{bmatrix} \cos\phi\cos\theta & -\sin\theta & -\sin\phi\cos\theta \\ \cos\phi\sin\theta & \cos\theta & -\sin\phi\sin\theta \\ \sin\phi & 0 & \cos\phi \end{bmatrix} \begin{bmatrix} r_o \\ \theta_o \\ \phi_o \end{bmatrix} \qquad (2.2)$$

All of the trigonometric functions in the transformation can be precomputed using the trigonometric relationships outlined in Section 2.1.2.

The gravitational acceleration in the orthogonal spherical coordinate system is [6, pg. 54]

$$a = \begin{bmatrix} a_{r_o} & a_{\theta_o} & a_{\phi_o} \end{bmatrix}^T = \nabla V = \begin{bmatrix} \frac{\partial V}{\partial r} & \frac{1}{r\cos\phi}\frac{\partial V}{\partial \theta} & \frac{1}{r}\frac{\partial V}{\partial \phi} \end{bmatrix}^T \qquad (2.3)$$

The result of Equation 2.3 can then be transformed back to central-body-fixed coordinates via Equation 2.2.

2.1.2   Trigonometric Relationships

Instead of solving for the spherical coordinate angles ϕ and θ, it is more efficient to precompute the values of trigonometric functions of these angles given by the definitions of sine and cosine.

$$\sin\theta \;=\; \frac{y_b}{\sqrt{x_b^2 + y_b^2}} \tag{2.4}$$

$$\cos\theta \;=\; \frac{x_b}{\sqrt{x_b^2 + y_b^2}} \tag{2.5}$$

$$\sin\phi \;=\; \frac{z_b}{r} \tag{2.6}$$

$$\cos\phi \;=\; \frac{\sqrt{x_b^2 + y_b^2}}{r} \tag{2.7}$$

A logical check should be included prior to computing these values to identify the cases $x_b = y_b = 0$ or $r = 0$. If $x_b = y_b = 0$, then the equations $\sin\theta = 0$ and $\cos\theta = 1$ are typically substituted. Similarly, if $r = 0$, then the equations $\sin\phi = 0$ and $\cos\phi = 1$ are typically substituted.

2.2   Constants and Coefficients

$S_{n,m}$ and $C_{n,m}$ are known in the literature as the spherical harmonic mass coefficients of the central body, which we will refer to simply as the *mass coefficients*. Certain subsets of the mass coeffients are given special names.

- $C_{n,0}$ = zonal coefficients

- $S_{n,n}, C_{n,n}$ = sectorial coefficients

- $S_{n,m}, C_{n,m}$ = tesseral coefficients

Tables of mass coefficients are usually provided with the gravitational parameter $\mu = GM$ (where G is the Newtonian gravitational constant and M is the mass of the central body) and the scaling radius a_{eq} for which the coefficients are calibrated.

2.3   Legendre polynomials and ALFs

The various functions denoted P in this paper are the Legendre polynomials and the Associated Legendre Functions (ALFs). Legendre polynomials of the first kind are denoted P_n with argument $\sin\phi$. ALFs of the first kind are similarly denoted $P_{n,m}$ also with argument $\sin\phi$. Legendre polynomials and ALFs have subscipts n and m which are called the degree and order of the polynomial, respectively. ALFs with $m > n$ are defined as zero. The Legendre polynomials are equivalent to the ALF of the same degree but with $m = 0$.

In the remainder of this document, the argument of $\sin\phi$ will be assumed and omitted for brevity for all Legendre polynomials and ALFs.

2.4 The Gravitational Potential Function

The gravitational potential V satisfies Laplace's equation

$$\frac{\partial^2 V}{\partial x_b{}^2} + \frac{\partial^2 V}{\partial y_b{}^2} + \frac{\partial^2 V}{\partial z_b{}^2} = 0 \tag{2.8}$$

The potential function V can be written in spherical coordinates as an orthogonal expansion using spherical harmonics [6, Pg. 52].

$$V = \frac{\mu}{r} \left[1 + \sum_{n=1}^{\infty} \sum_{m=0}^{n} \left(\frac{a_{eq}}{r} \right)^n P_{n,m} \left(S_{n,m} \sin m\theta + C_{n,m} \cos m\theta \right) \right] \tag{2.9}$$

Equation 2.9 can also be written with the zonal terms separated out

$$V = \frac{\mu}{r} \left[1 + \sum_{n=1}^{\infty} \left(\frac{a_{eq}}{r} \right)^n P_n C_{n,0} \right.$$
$$\left. + \sum_{n=1}^{\infty} \sum_{m=1}^{n} \left(\frac{a_{eq}}{r} \right)^n P_{n,m} \left(S_{n,m} \sin m\theta + C_{n,m} \cos m\theta \right) \right] \tag{2.10}$$

2.4.1 Square Gravity Models

The $S_{n,m}$, $C_{n,m}$, and $P_{n,m}$ values can be used to form lower triangular matricies, where the terms above the diagonal are all zero, n is the row, and m is the column.

When the origin of the central body-fixed coordinate system is located at the center of mass, the $C_{1,0}$, $S_{1,1}$, and $C_{1,1}$ coefficients are all zero. This convention will be adopted throughout this paper. Therefore, in computing V, the sums may be made starting with $n = 2$. Models where maximum degree and order are equal are referred to as *square models*. Let n_d be the desired maximum degree and order of the gravity model. Then

$$V = \frac{\mu}{r} \left[1 + \sum_{n=2}^{n_d} \sum_{m=0}^{n} \left(\frac{a_{eq}}{r} \right)^n P_{n,m} \left(S_{n,m} \sin m\theta + C_{n,m} \cos m\theta \right) \right] \tag{2.11}$$

With the zonal terms separated out, Equation 2.11 becomes

$$V = \frac{\mu}{r} \left[1 + \sum_{n=2}^{n_d} \left(\frac{a_{eq}}{r} \right)^n P_n C_{n,0} \right.$$
$$\left. + \sum_{n=2}^{n_d} \sum_{m=1}^{n} \left(\frac{a_{eq}}{r} \right)^n P_{n,m} \left(S_{n,m} \sin m\theta + C_{n,m} \cos m\theta \right) \right] \tag{2.12}$$

2.4.2   Non-Square Gravity Models

Whereas Equations 2.11 and 2.12 assume square models, it is also possible to approximate the gravitational potential using *non-square models*. This can be accomplished by changing the bounds of the sums in Equations 2.11 and 2.12 to n_d and m_d, the desired degree and desired order, respectively. The outcome of a non-square model can then be written as

$$V = \frac{\mu}{r} \left[1 + \sum_{n=2}^{n_d} \sum_{\substack{m=0 \\ m \leq m_d}}^{n} \left(\frac{a_{eq}}{r} \right)^n P_{n,m} \left(S_{n,m} \sin m\theta + C_{n,m} \cos m\theta \right) \right] \quad (2.13)$$

resembling Equation 2.11, or separating out zonal terms,

$$V = \frac{\mu}{r} \left[1 + \sum_{n=2}^{n_d} \left(\frac{a_{eq}}{r} \right)^n P_n C_{n,0} \right.$$

$$\left. + \sum_{n=2}^{n_d} \sum_{\substack{m=1 \\ m \leq m_d}}^{n} \left(\frac{a_{eq}}{r} \right)^n P_{n,m} \left(S_{n,m} \sin m\theta + C_{n,m} \cos m\theta \right) \right] \quad (2.14)$$

resembling Equation 2.12.

Because most algorithms for generating the Legendre polynomials and ALFs are optimized to generate values for square models, it is best to pass only the desired degree of the non-square gravity model to these function-generating subroutines. In calculating a non-square potential, the excess ALFs beyond the desired order are then simply unused.

The fact that Equations 2.11 and 2.12 (and thus Equations 2.13 and 2.14) are orthogonal expansions of V means that any lower order expansion is merely a truncated higher order expansion. No refit of the coefficients is necessary [2].

Chapter 3

Normalization

3.1   The Normalization Factor

Using a normalization factor allows mass coefficients to be electronically represented as tractable values well within the valid range for IEEE 754 double-precision floating point variables, even when degree n and order m are relatively large. The mass coefficients of a central body are "normalized" when they are divided by this normalization factor $N_{n,m}$, typically defined as [8, pg. 544]

$$N_{n,m} = \sqrt{\frac{(n-m)!(2n+1)(2-\delta_{0,m})}{(n+m)!}}, \ N_n = N_{n,0} \tag{3.1}$$

where $\delta_{0,m}$ is the Kronecker delta function that returns one if $m = 0$ and zero otherwise. Table 3.1 lists the normalization factors through degree and order of four.

Historically, mass coefficients were "unnormalized" by the end user of the model by multiplying the mass coefficients with their corresponding normalization factor. This was done because conventional gravitational potential algorithms required unnormalized coefficients in their formulations. A fundamental

		$m \rightarrow$			
$n \downarrow$	0	1	2	3	4
0	1	0	0	0	0
1	$\sqrt{3}$	$\sqrt{3}$	0	0	0
2	$\sqrt{5}$	$\sqrt{\frac{5}{3}}$	$\frac{1}{2}\sqrt{\frac{5}{3}}$	0	0
3	$\sqrt{7}$	$\sqrt{\frac{7}{6}}$	$\frac{1}{2}\sqrt{\frac{7}{15}}$	$\frac{1}{6}\sqrt{\frac{7}{10}}$	0
4	3	$3\sqrt{\frac{1}{10}}$	$\frac{1}{2}\sqrt{\frac{1}{5}}$	$\frac{1}{2}\sqrt{\frac{1}{70}}$	$\frac{1}{8}\sqrt{\frac{1}{35}}$

Table 3.1: Normalization factors through 4×4

9

problem encountered by attempting to unnormalize mass coefficients with very high degrees and orders is that the normalization factors become prone to overflow in modern computers. This is due to the factorial in the denominator of Equation 3.1, $(n + m)!$. The user is restricted to $n_d + m_d < 171$ using IEEE 754 double-precision floating point variables to calculate normalization factors because of overflow beyond this boundary (see Section 3.2).

It was later established that the additional work of unnormalizing mass coefficients becomes unnecessary by "normalizing" ALFs in gravitational potential formulations. Legendre polynomials and ALFs are said to be "normalized" when multiplied by $N_{n,m}$. This normalization scheme is ideal because the product of an ALF and its corresponding coefficient is equal to the product of the normalized ALF and its corresponding normalized coefficient. In this paper, an overbar will be used to indicate normalized quantities, i.e. $\bar{C}_{n,m}$ and $\bar{P}_{n,m}$. By the aforementioned normalization conventions, Equation 3.2 shows that the product of the ALF and corresponding coefficient holds true for normalized quantities.

$$\bar{P}_{n,m}\bar{C}_{n,m} = (P_{n,m}N_{n,m}) \left(\frac{C_{n,m}}{N_{n,m}} \right) = P_{n,m}C_{n,m} \qquad (3.2)$$

This eliminates the restrictions imposed by requiring unnormalized mass coefficients for gravitational potential formulations. The relationship from Equation 3.2 can be seen with $S_{n,m}$ and $\bar{S}_{n,m}$ as well.

3.2   Recursive Mass Coefficient Normalization

Gravity models evaluating spherical harmonic associated Legendre functions at high degree n and order m require normalized Legendre coefficients to accurately compute terms ranging over hundreds of orders of magnitude. The standard normalization scheme is to divide the unnormalized coefficients by the Kaula normalization factor $N_{n,m}$, defined in Equation 3.1. If $N_{n,m}$ is calculated directly, computation difficulties arise when $n + m > 170$ because 171! triggers an overflow condition in IEEE double precision (64-bit) real numbers. The overflow limit is $\pm 1.797693134862316e+308$[1].

Suppose $N_{n,m}$ values are directly calculated and grouped as elements in a lower-triangular matrix, as is customary, by incrementing m from 0 until $m = n$ before n is incremented to begin another matrix row. In performing this task, the first $(n + m)!$ overflow will be encountered for $N_{86,85}$. Table 3.2 provides $N_{n,m}$ values neighboring the $N_{86,85}$ element. In Table 3.2, $N_{n,m}$ values which *can* be computed are nowhere near an overflow condition because they are quotients with large denominators. This suggests a recursive computation will succeed in generating accurate $N_{n,m}$ values far beyond element $N_{86,85}$. The recursion will only fail when the $N_{n,m}$ quotient overflows.

To document the recursion, suppose the value of $N_{n',m'}$ is given as x. As-

---

[1]This number was obtained from the `realmax('double')` command in MATLAB.

Degree n	Order m		
	84	85	86
85	1.117258027e+151	1.456726244e+152	N/A
86	1.024089587e+152	Overflow	Overflow
87	Overflow	Overflow	Overflow

Table 3.2: Normalization factors neighboring $N_{86,85}$

Degree	Order		
	$m'-1$	m'	$m'+1$
$n'-1$		$x\sqrt{\frac{(n'-m')(2n'+1)}{(n'+m')(2n'-1)}}$	
n'	$x\sqrt{\frac{2-\delta_{0,m'}}{(n'+m')(n'-m'+1)}}$	x	$x\sqrt{\frac{(n'+m'+1)(n'-m')}{2-\delta_{0,m'}}}$
$n'+1$		$x\sqrt{\frac{(n'+1+m')(2n'+1)}{(n'+1-m')(2n'+3)}}$	

Table 3.3: Recursions for generating normalization factors

suming $n'-m' > 0$ for "upward" or "rightward" recursion[2], Table 3.3 supplies recursive formulae for adjacent elements in terms of x and its associated degree n' and order m'. Each formula has been verified using the three finite $N_{n,m}$ elements appearing in Table 3.2. For example, invoking the "downward" recursion,

$$N_{86,84} = N_{85,84}\sqrt{\frac{170 \cdot 171}{2 \cdot 173}} \approx 9.16609737241 N_{85,84}$$

Most gravity models are formally published with normalized coefficients. Those wishing to use normalized coefficients at $n+m > 170$ in an unnormalized model will find the Table 3.3 recursions useful in their work. At $n+m \leq 170$, the Table 3.3 recursions offer computational efficiencies over Equation 3.1 evaluations, but these will be of little consequence if coefficients are to be unnormalized in a single pass with the results stored for all subsequent use.

3.3   Normalization Ratios

Consider a recursion formula for Legendre Polynomials [9, pg. 114, sec. 3]

$$P_n = \frac{1}{n}[(2n-1)\sin\phi\,P_{n-1} - (n-1)P_{n-2}] \tag{3.3}$$

The corresponding normalized Legendre polynomial $\bar{P}_n$ is found by multiplying Equation 3.3 by the normalization factor N_n.

$$\bar{P}_n = N_n P_n = \frac{1}{n}[(2n-1)\sin\phi\,N_n P_{n-1} - (n-1)N_n P_{n-2}] \tag{3.4}$$

---

[2]These forbidden recursions would otherwise step outside lower-triangular matrix limits if x corresponded to a diagonal element with $n'-m' = 0$.

Because Equation 3.4 returns normalized polynomials and must recur over its own inputs, the equation needs to be written as a function of normalized polynomials by replacing the conventional polynomials with their normalized equivalents. By substituting

$$P_n = \frac{\bar{P}_n}{N_n} \tag{3.5}$$

Equation 3.4 becomes

$$\bar{P}_n = \frac{1}{n}\left[(2n-1)\sin\phi\,\frac{N_n}{N_{n-1}}\bar{P}_{n-1} - (n-1)\frac{N_n}{N_{n-2}}\bar{P}_{n-2}\right] \tag{3.6}$$

Equation 3.6 now contains *ratios* of normalization factors, namely $\frac{N_n}{N_{n-1}}$ and $\frac{N_n}{N_{n-2}}$.

3.4   The Recursion Normalization Parameter

The ratios of the normalization factors, which are referred to as the *normalization parameters* and denoted by λ, can be pre-computed since many ratios will be needed more than once while normalizing the algorithms. The ratios of Equation 3.6 are written as λ_{n-1} and λ_{n-2}, where the subscripts of the parameters refer to the subscript of the corresponding polynomial which they normalize in Equation 3.6.

When $n = 2$, $\lambda_{n-1} = \lambda_1$. However, when $n = 3$, $\lambda_{n-2} \neq \lambda_1$. For this reason, the parameters are written as functions of the current values of n and m, such as $\lambda_{n-1}(n)$ and $\lambda_{n-2,m}(n,m)$, and ignore the actual value of the subscripts of the parameters. The subscripts are merely notation used to identify a specific parameter for normalizing the ALF with the same subscripts.

This notation is used in Equation 3.6 to obtain

$$\bar{P}_n = \frac{1}{n}[(2n-1)\sin\phi\,\lambda_{n-1}(n)\bar{P}_{n-1} - (n-1)\lambda_{n-2}(n)\bar{P}_{n-2}] \tag{3.7}$$

An equation for the parameter λ_{n-1} can be found by substituting the definition of the normalization factors (Equation 3.1) in the ratio and simplifying.

$$\lambda_{n-1}(n) = \frac{N_n}{N_{n-1}} = \frac{\sqrt{2n+1}}{\sqrt{2(n-1)+1}} = \sqrt{\frac{2n+1}{2n-1}} \tag{3.8}$$

The Lear algorithm (Section 4.2) requires five normalization parameters to recursively compute normalized Legendre polynomials and ALFs. Each of these parameters can similarly be derived by simplifying the definition of the normalization factor for each of the corresponding ratios.

Chapter 4

The Three Singularity-Free Algorithms

By inspection, we can identify two potential singularities in Equation 2.3. They can occur when $\cos\phi = 0$ (such as in a polar orbit) or when taking the partial derivative of ALFs with $m = 1$. Three unique algorithms have been developed to eliminate these singularities by Samuel Pines [1], Bill Lear [2], and Robert Gottlieb [3].

4.1 Pines Algorithm

4.1.1 Basis of the Pines Approach

Pines approached the problem by first transforming a position vector to central-body-fixed coordinates. By reallocating factors in each term of the potential, he defined a series of special functions in terms of the unit position vector components which can then be solved recursively without singularities. A set of polynomials he referred to as the *derived ALFs* were created by modifying the conventional definition of ALFs. The derived ALFs have similar recursive behaviors to their conventional counterparts but have no discontinuities in their partial derivatives.

4.1.2 Pines Algorithm Implementations

The Pines acceleration algorithm has previously been normalized by Lundberg and Schutz [4]. Within the normalized algorithm, Pines' derived ALFs can be recursively generated in a number of ways. To evaluate the various approaches, Lundberg and Schutz developed seven different recursions algorithms and then performed numerical analyses of the stability of normalized and unnormalized versions of each recursion algorithm. Lundberg and Schutz concluded that a simple row-wise or column-wise recursion provides the most stability of all the

13

$$\lambda_{n-1}(n) \quad = \quad \frac{N_n}{N_{n-1}} = \sqrt{\frac{2n+1}{2n-1}} \tag{4.1}$$

$$\lambda_{n-2}(n) \quad = \quad \frac{N_n}{N_{n-2}} = \sqrt{\frac{2n+1}{2n-3}} \tag{4.2}$$

$$\lambda_{n-1,n-1}(n) \quad = \quad \frac{N_{n,n}}{N_{n-1,n-1}} = \frac{1}{2n-1}\sqrt{\frac{2n+1}{2n}} \tag{4.3}$$

$$\lambda_{n-1,m}(n,m) \quad = \quad \frac{N_{n,m}}{N_{n-1,m}} = \sqrt{\frac{(n-m)(2n+1)}{(n+m)(2n-1)}} \tag{4.4}$$

$$\lambda_{n-2,m}(n,m) \quad = \quad \frac{N_{n,m}}{N_{n-2,m}} = \sqrt{\frac{(n-m)(n-m-1)(2n+1)}{(n+m)(n+m-1)(2n-3)}} \tag{4.5}$$

Table 4.1: Normalization parameters λ for a normalized Lear algorithm

algorithms, normalized or unnormalized. In this analysis, the normalized and unnormalized versions of the column-wise recursion by Lundberg and Schutz [4, Recursion I] are utilized for implementations of Pines. The implementations used here are based on a normalized implementation provided by DeMars [5].

4.2   Lear Algorithm

4.2.1   Basis of the Lear Approach

Lear transformed the position vector to the orthogonal spherical coordinate system. This results in several $\sec\phi$ factors that emerge in the equations for the acceleration. Lear found that stable recursions could be developed by assimilating the $\sec\phi$ factors in the ALF recursion equations. Lear utilized the conventional (unmodified) ALFs and thus used traditional recursion equations for ALFs, across which he simply distributed the $\sec\phi$ factors. Terms in the potential which include an ALF but no $\sec\phi$ could easily eliminate the secant (which has been combined in the value of the ALF) by multiplying the term by $\cos\phi$, a value with no discontinuity.

4.2.2   Normalized Lear Algorithm

Each of the recursion equations of the Lear algorithm is now normalized using the normalization parameters of Section 3.4. The parameters required in the Lear algorithm are defined in Table 4.1. Each of the original equations can be found in Ref. [2]. Only the normalized equations are presented here and should replace their analogues in the original algorithm. The added normalization parameters are denoted with an underbrace or overbrace to bring attention to the changes from the equations in the original document.

In each of the following recursions, let n_d be the desired degree and order of the gravity model.

Recursion for Zonal Legendre Polynomials $\bar{P}_n$

This recursion utilizes Parameters 4.1 and 4.2. For $n = 2$ through n_d

$$\bar{P}_n = \frac{1}{n}[(2n-1)\sin\phi \underbrace{\lambda_{n-1}(n)}_{4.1} \bar{P}_{n-1} - (n-1)\underbrace{\lambda_{n-2}(n)}_{4.2} \bar{P}_{n-2}] \qquad (4.6)$$

where $\bar{P}_0 = 1$ and $\bar{P}_1 = N_1 P_1 = \sqrt{3}\sin\phi$.

Recursion for Zonal Legendre Polynomial Derivatives $\bar{P}'_n$

This recursion utilizes Parameter 4.1. For $n = 2$ through n_d

$$\bar{P}'_n = \underbrace{\lambda_{n-1}(n)}_{4.1}[\sin\phi\,\bar{P}'_{n-1} + n\bar{P}_{n-1}] \qquad (4.7)$$

where $\bar{P}'_1 = \sqrt{3}$.

Recursion for Sectorial ALFs $(\sec\phi\,\bar{P}_{n,n})$

This recursion utilizes Parameter 4.3. For $n = 2$ through n_d

$$(\sec\phi\,\bar{P}_{n,n}) = (2n-1)\cos\phi \underbrace{\lambda_{n-1,n-1}(n)}_{4.3}(\sec\phi\,\bar{P}_{n-1,n-1}) \qquad (4.8)$$

where $(\sec\phi\,\bar{P}_{1,1}) = \sqrt{3}$.

Recursion for Tesseral ALFs $(\sec\phi\,\bar{P}_{n,m})$

This recursion utilizes Parameters 4.4 and 4.5. For $n = 2$ through n_d and (inner loop) $m = 1$ through $n - 1$

$$\begin{aligned}(\sec\phi\,\bar{P}_{n,m}) = [(2n-1)\sin\phi \overbrace{\lambda_{n-1,m}(n,m)}^{4.4}(\sec\phi\,\bar{P}_{n-1,m}) \\ - (n+m-1)\underbrace{\lambda_{n-2,m}(n,m)}_{4.5}(\sec\phi\,\bar{P}_{n-2,m})]\frac{1}{n-m}\end{aligned} \qquad (4.9)$$

where $(\sec\phi\,\bar{P}_{n-1,n}) = 0$ for $n = 1$ through n_d.

Recursion for Sectorial ALF Derivatives $(\cos\phi\,\bar{P}'_{n,n})$

This recursion has no normalization parameters because the input and output are the same degree and order. For $n = 1$ through n_d

$$(\cos\phi\,\bar{P}'_{n,n}) = -n\sin\phi\,(\sec\phi\,\bar{P}_{n,n}) \qquad (4.10)$$

Recursion for Tesseral ALF Derivatives $(\cos\phi\,\bar{P}'_{n,m})$

This recursion utilizes Parameter 4.4. For $n = 2$ through n_d and (inner loop) $m = 1$ through $n - 1$

$$
\begin{aligned}
(\cos\phi\,\bar{P}'_{n,m}) = &-n\sin\phi\,(\sec\phi\,\bar{P}_{n,m}) \\
&+ (n+m)\,\underbrace{\lambda_{n-1,m}(n,m)}_{4.4}(\sec\phi\,\bar{P}_{n-1,m})
\end{aligned} \tag{4.11}
$$

4.2.3   Example of Normalized Lear Recursions

This section analytically demonstrates using normalized recursion relationships to generate Legendre polynomials and ALFs for a gravity model with a degree and order of four. Each of the final answers are written first simplified and then with its normalization factored out to demonstrate equivalence with its unnormalized value. These can be found in the examples outlined in Ref. [2].

Zonal Legendre Polynomials

$$
\begin{aligned}
\bar{P}_2 &= \frac{1}{2}\left[3\sin\phi\,\lambda_{n-1}(2)\bar{P}_1 - \lambda_{n-2}(2)\bar{P}_0\right] \\
&= \frac{1}{2}\left[3\sin\phi\,\sqrt{\frac{5}{3}}\sqrt{3}\sin\phi - \sqrt{5}\right] \\
&= \frac{3}{2}\sqrt{5}\sin^2\phi - \frac{1}{2}\sqrt{5} = \sqrt{5}\left(\frac{3}{2}\sin^2\phi - \frac{1}{2}\right)
\end{aligned}
$$

$$
\begin{aligned}
\bar{P}_3 &= \frac{1}{3}\left[5\sin\phi\,\lambda_{n-1}(3)\bar{P}_2 - 2\lambda_{n-2}(3)\bar{P}_1\right] \\
&= \frac{1}{3}\left[5\sin\phi\,\sqrt{\frac{7}{5}}\left(\frac{3}{2}\sqrt{5}\sin^2\phi - \frac{1}{2}\sqrt{5}\right) - 2\sqrt{\frac{7}{3}}\sqrt{3}\sin\phi\right] \\
&= \frac{1}{3}\left[\frac{5\cdot3}{2}\sqrt{7}\sin^3\phi - \frac{5}{2}\sqrt{7}\sin\phi - 2\sqrt{7}\sin\phi\right] \\
&= \frac{5}{2}\sqrt{7}\sin^3\phi - \frac{3}{2}\sqrt{7}\sin\phi = \sqrt{7}\left(\frac{5}{2}\sin^3\phi - \frac{3}{2}\sin\phi\right)
\end{aligned}
$$

$$
\begin{aligned}
\bar{P}_4 &= \frac{1}{4}\left[7\sin\phi\,\lambda_{n-1}(4)\bar{P}_3 - 3\lambda_{n-2}(4)\bar{P}_2\right] \\
&= \frac{1}{4}\left[7\sin\phi\,\sqrt{\frac{9}{7}}\left(\frac{5}{2}\sqrt{7}\sin^3\phi - \frac{3}{2}\sqrt{7}\sin\phi\right)\right. \\
&\quad\left. - 3\sqrt{\frac{9}{5}}\left(\frac{3}{2}\sqrt{5}\sin^2\phi - \frac{1}{2}\sqrt{5}\right)\right]
\end{aligned}
$$

$$= \frac{1}{4} \left[\frac{7 \cdot 5 \cdot 3}{2} \sin^4 \phi - \frac{7 \cdot 3 \cdot 3}{2} \sin^2 \phi - \frac{3 \cdot 3 \cdot 3}{2} \sin^2 \phi + \frac{3 \cdot 3}{2} \right]$$

$$= \frac{105}{8} \sin^4 \phi - \frac{90}{8} \sin^2 \phi + \frac{9}{8} = 3 \left(\frac{35}{8} \sin^4 \phi - \frac{30}{8} \sin^2 \phi + \frac{3}{8} \right)$$

Zonal Legendre Polynomial Derivatives

$$\bar{P}'_2 = \lambda_{n-1}(2) \left[\sin \phi \bar{P}'_1 + 2\bar{P}_1 \right]$$

$$= \sqrt{\frac{5}{3}} \left[\sin \phi \sqrt{3} + 2\sqrt{3} \sin \phi \right]$$

$$= 3\sqrt{5} \sin \phi = \sqrt{5} \left(3 \sin \phi \right)$$

$$\bar{P}'_3 = \lambda_{n-1}(3) \left[\sin \phi \bar{P}'_2 + 3\bar{P}_2 \right]$$

$$= \sqrt{\frac{7}{5}} \left[\sin \phi \left(3\sqrt{5} \sin \phi \right) + 3 \left(\frac{3}{2} \sqrt{5} \sin^2 \phi - \frac{1}{2} \sqrt{5} \right) \right]$$

$$= 3\sqrt{7} \sin^2 \phi + \frac{3 \cdot 3}{2} \sqrt{7} \sin^2 \phi - \frac{3}{2} \sqrt{7}$$

$$= \frac{15}{2} \sqrt{7} \sin^2 \phi - \frac{3}{2} \sqrt{7} = \sqrt{7} \left(\frac{15}{2} \sin^2 \phi - \frac{3}{2} \right)$$

$$\bar{P}'_4 = \lambda_{n-1}(4) \left[\sin \phi \bar{P}'_3 + 4\bar{P}_3 \right]$$

$$= \sqrt{\frac{9}{7}} \left[\sin \phi \left(\frac{15}{2} \sqrt{7} \sin^2 \phi - \frac{3}{2} \sqrt{7} \right) + 4 \left(\frac{5}{2} \sqrt{7} \sin^3 \phi - \frac{3}{2} \sqrt{7} \sin \phi \right) \right]$$

$$= \frac{3 \cdot 15}{2} \sin^3 \phi - \frac{3 \cdot 3}{2} \sin \phi + \frac{3 \cdot 4 \cdot 5}{2} \sin^3 \phi - \frac{4 \cdot 3 \cdot 3}{2} \sin \phi$$

$$= \frac{105}{2} \sin^3 \phi - \frac{45}{2} \sin \phi = 3 \left(\frac{35}{2} \sin^3 \phi - \frac{15}{2} \sin \phi \right)$$

Sectorial (Diagonal) ALFs

$$(\sec \phi \, \bar{P}_{2,2}) = 3 \cos \phi \, \lambda_{n-1,n-1}(2)(\sec \phi \, \bar{P}_{1,1})$$

$$= 3 \cos \phi \frac{1}{3} \sqrt{\frac{5}{4}} \sqrt{3}$$

$$= \frac{1}{2} \sqrt{15} \cos \phi = \frac{1}{2} \sqrt{\frac{5}{3}} \left(3 \cos \phi \right)$$

$$(\sec \phi \, \bar{P}_{3,3}) = 5 \cos \phi \, \lambda_{n-1,n-1}(3)(\sec \phi \, \bar{P}_{2,2})$$

$$= 5\cos\phi\,\frac{1}{5}\sqrt{\frac{7}{6}}\left(\frac{1}{2}\sqrt{15}\cos\phi\right)$$

$$= \frac{1}{2}\sqrt{\frac{35}{2}}\cos^2\phi = \frac{1}{6}\sqrt{\frac{7}{10}}\left(15\cos^2\phi\right)$$

$$(\sec\phi\,\bar{P}_{4,4}) = 7\cos\phi\,\lambda_{n-1,n-1}(4)(\sec\phi\,\bar{P}_{3,3})$$

$$= 7\cos\phi\,\frac{1}{7}\sqrt{\frac{9}{8}}\left(\frac{1}{2}\sqrt{\frac{35}{2}}\cos^2\phi\right)$$

$$= \frac{3}{8}\sqrt{35}\cos^3\phi = \frac{1}{8}\sqrt{\frac{1}{35}}\left(105\cos^3\phi\right)$$

Tesseral ALFs: Row 2 $(n=2)$

$$(\sec\phi\,\bar{P}_{2,1}) = 3\sin\phi\,\lambda_{n-1,m}(2,1)(\sec\phi\,\bar{P}_{1,1}) - 2\lambda_{n-2,m}(2,1)(\sec\phi\,\bar{P}_{0,1})$$

$$= 3\sin\phi\,\sqrt{\frac{5}{3\cdot 3}}\sqrt{3} - 2\sqrt{0}\cdot 0$$

$$= \sqrt{15}\sin\phi = \sqrt{\frac{5}{3}}\left(3\sin\phi\right)$$

Tesseral ALFs: Row 3 $(n=3)$

$$(\sec\phi\,\bar{P}_{3,1}) = \left[5\sin\phi\,\lambda_{n-1,m}(3,1)(\sec\phi\,\bar{P}_{2,1}) - 3\lambda_{n-2,m}(3,1)(\sec\phi\,\bar{P}_{1,1})\right]\frac{1}{2}$$

$$= \frac{1}{2}\left[5\sin\phi\,\sqrt{\frac{2\cdot 7}{4\cdot 5}}\left(\sqrt{15}\sin\phi\right) - 3\sqrt{\frac{2\cdot 7}{4\cdot 3\cdot 3}}\sqrt{3}\right]$$

$$= \frac{5}{2}\sqrt{\frac{7\cdot 3}{2}}\sin^2\phi - \frac{1}{2}\sqrt{\frac{7\cdot 3}{2}}$$

$$= \frac{5}{2}\sqrt{\frac{21}{2}}\sin^2\phi - \frac{1}{2}\sqrt{\frac{21}{2}} = \sqrt{\frac{7}{6}}\left(\frac{15}{2}\sin^2\phi - \frac{3}{2}\right)$$

$$(\sec\phi\,\bar{P}_{3,2}) = 5\sin\phi\,\lambda_{n-1,m}(3,2)(\sec\phi\,\bar{P}_{2,2}) - 4\lambda_{n-2,m}(3,2)(\sec\phi\,\bar{P}_{1,2})$$

$$= 5\sin\phi\,\sqrt{\frac{7}{5\cdot 5}}\left(\frac{1}{2}\sqrt{15}\cos\phi\right) - 4\sqrt{0}\cdot 0$$

$$= \frac{5}{2}\sqrt{\frac{7\cdot 5\cdot 3}{5\cdot 5}}\sin\phi\cos\phi$$

$$= \frac{1}{2}\sqrt{105}\sin\phi\cos\phi = \frac{1}{2}\sqrt{\frac{7}{15}}\left(15\sin\phi\cos\phi\right)$$

Tesseral ALFs: Row 4 ($n = 4$)

The example in the Lear document which corresponds to the following example contains a typographical error. The final result for the ALF should be

$$(\sec\phi\, P_{4,1}) = \frac{35}{2}\sin^3\phi - \frac{15}{2}\sin\phi$$

$$
\begin{aligned}
(\sec\phi\,\bar{P}_{4,1}) &= \left[7\sin\phi\,\lambda_{n-1,m}(4,1)(\sec\phi\,\bar{P}_{3,1}) - 4\lambda_{n-2,m}(4,1)(\sec\phi\,\bar{P}_{2,1})\right]\frac{1}{3} \\
&= \frac{7}{3}\sin\phi\sqrt{\frac{3\cdot 9}{5\cdot 7}}\left(\frac{5}{2}\sqrt{\frac{21}{2}}\sin^2\phi - \frac{1}{2}\sqrt{\frac{21}{2}}\right) \\
&\quad -\frac{4}{3}\sqrt{\frac{3\cdot 2\cdot 9}{5\cdot 4\cdot 5}}\left(\sqrt{15}\sin\phi\right) \\
&= \frac{7\cdot 5}{3\cdot 2}\sqrt{\frac{3\cdot 9\cdot 7\cdot 3}{5\cdot 7\cdot 2}}\sin^3\phi - \frac{7}{3\cdot 2}\sqrt{\frac{3\cdot 9\cdot 7\cdot 3}{5\cdot 7\cdot 2}}\sin\phi \\
&\quad -\frac{4}{3}\sqrt{\frac{3\cdot 2\cdot 9\cdot 3\cdot 5}{5\cdot 4\cdot 5}}\sin\phi \\
&= \frac{21}{2}\sqrt{\frac{5}{2}}\sin^3\phi - \frac{9}{2}\sqrt{\frac{5}{2}}\sin\phi \\
&= 3\sqrt{\frac{1}{10}}\left(\frac{35}{2}\sin^3\phi - \frac{15}{2}\sin\phi\right)
\end{aligned}
$$

$$
\begin{aligned}
(\sec\phi\,\bar{P}_{4,2}) &= \left[7\sin\phi\,\lambda_{n-1,m}(4,2)(\sec\phi\,\bar{P}_{2,2}) - 5\lambda_{n-2,m}(4,2)(\sec\phi\,\bar{P}_{2,2})\right]\frac{1}{2} \\
&= \frac{7}{2}\sin\phi\sqrt{\frac{2\cdot 9}{6\cdot 7}}\left(\frac{1}{2}\sqrt{105}\sin\phi\cos\phi\right) \\
&\quad -\frac{5}{2}\sqrt{\frac{2\cdot 9}{6\cdot 5\cdot 5}}\left(\frac{1}{2}\sqrt{15}\cos\phi\right) \\
&= \frac{7}{4}\sqrt{\frac{2\cdot 3\cdot 3\cdot 3\cdot 5\cdot 7}{2\cdot 3\cdot 7}}\sin^2\phi\cos\phi - \frac{5}{4}\sqrt{\frac{2\cdot 3\cdot 5\cdot 3\cdot 3}{2\cdot 3\cdot 5\cdot 5}}\cos\phi \\
&= \frac{21}{4}\sqrt{5}\sin^2\phi\cos\phi - \frac{3}{4}\sqrt{5}\cos\phi \\
&= \frac{1}{2}\sqrt{\frac{1}{5}}\left(\frac{105}{2}\sin^2\phi\cos\phi - \frac{15}{2}\cos\phi\right)
\end{aligned}
$$

$$
\begin{aligned}
(\sec\phi\,\bar{P}_{4,3}) &= 7\sin\phi\,\lambda_{n-1,m}(4,3)(\sec\phi\,\bar{P}_{1,1}) - 6\lambda_{n-2,m}(4,3)(\sec\phi\,\bar{P}_{2,3}) \\
&= 7\sin\phi\sqrt{\frac{9}{7\cdot 7}}\left(\frac{1}{2}\sqrt{\frac{35}{2}}\cos^2\phi\right) - 6\sqrt{0}\cdot 0 \\
&= \frac{7}{2}\sqrt{\frac{3\cdot 3\cdot 5\cdot 7}{2\cdot 7\cdot 7}}\sin\phi\cos^2\phi
\end{aligned}
$$

$$= \frac{3}{2}\sqrt{\frac{35}{2}}\sin\phi\cos^2\phi = \frac{1}{2}\sqrt{\frac{1}{70}}\left(105\sin\phi\cos^2\phi\right)$$

Sectorial (Diagonal) ALF Derivatives

$$
\begin{aligned}
(\cos\phi\,\bar{P}'_{1,1}) &= -\sin\phi\,(\sec\phi\,\bar{P}_{1,1}) \\
&= -\sqrt{3}\sin\phi = \sqrt{3}\,(-\sin\phi)
\end{aligned}
$$

$$
\begin{aligned}
(\cos\phi\,\bar{P}'_{2,2}) &= -2\sin\phi\,(\sec\phi\,\bar{P}_{2,2}) \\
&= -2\sin\phi\left(\frac{1}{2}\sqrt{15}\cos\phi\right) \\
&= -\sqrt{15}\sin\phi\cos\phi = \frac{1}{2}\sqrt{\frac{5}{3}}\left(-6\sin\phi\cos\phi\right)
\end{aligned}
$$

$$
\begin{aligned}
(\cos\phi\,\bar{P}'_{3,3}) &= -3\sin\phi\,(\sec\phi\,\bar{P}_{3,3}) \\
&= -3\sin\phi\left(\frac{1}{2}\sqrt{\frac{35}{2}}\cos^2\phi\right) \\
&= -\frac{3}{2}\sqrt{\frac{35}{2}}\sin\phi\cos^2\phi = \frac{1}{6}\sqrt{\frac{7}{10}}\left(-45\sin\phi\cos^2\phi\right)
\end{aligned}
$$

$$
\begin{aligned}
(\cos\phi\,\bar{P}'_{4,4}) &= -4\sin\phi\,(\sec\phi\,\bar{P}_{4,4}) \\
&= -4\sin\phi\left(\frac{3}{8}\sqrt{35}\cos^3\phi\right) \\
&= -\frac{3}{2}\sqrt{35}\sin\phi\cos^3\phi = \frac{1}{8}\sqrt{\frac{1}{35}}\left(-420\sin\phi\cos^3\phi\right)
\end{aligned}
$$

Tesseral ALF Derivatives: Row 2 $(n = 2)$

$$
\begin{aligned}
(\cos\phi\,\bar{P}'_{2,1}) &= -2\sin\phi\,(\sec\phi\,\bar{P}_{2,1}) + 3\lambda_{n-1,m}(2,1)(\sec\phi\,\bar{P}_{1,1}) \\
&= -2\sin\phi\left(\sqrt{15}\sin\phi\right) + 3\sqrt{\frac{5}{3\cdot 3}}\sqrt{3} \\
&= -2\sqrt{15}\sin^2\phi + \sqrt{15} = \sqrt{\frac{5}{3}}\left(-6\sin^2\phi + 3\right)
\end{aligned}
$$

Tesseral ALF Derivatives: Row 3 $(n = 3)$

$$(\cos\phi\,\bar{P}'_{3,1}) = -3\sin\phi\,(\sec\phi\,\bar{P}_{3,1}) + 4\lambda_{n-1,m}(3,1)(\sec\phi\,\bar{P}_{2,1})$$

$$= -3\sin\phi\left(\frac{5}{2}\sqrt{\frac{21}{2}}\sin^2\phi - \frac{1}{2}\sqrt{\frac{21}{2}}\right) + 4\sqrt{\frac{2\cdot7}{4\cdot5}}\left(\sqrt{15}\sin\phi\right)$$

$$= -\frac{15}{2}\sqrt{\frac{21}{2}}\sin^3\phi + \frac{3}{2}\sqrt{\frac{21}{2}}\sin\phi + \sqrt{2\cdot7\cdot4\cdot3}\sin\phi$$

$$= -\frac{15}{2}\sqrt{\frac{21}{2}}\sin^3\phi + \frac{11}{2}\sqrt{\frac{21}{2}}\sin\phi$$

$$= \sqrt{\frac{7}{6}}\left(-\frac{45}{2}\sin^3\phi + \frac{33}{2}\sin\phi\right)$$

$$(\cos\phi\,\bar{P}'_{3,2}) = -3\sin\phi\,(\sec\phi\,\bar{P}_{3,2}) + 5\lambda_{n-1,m}(3,2)(\sec\phi\,\bar{P}_{2,2})$$

$$= -3\sin\phi\left(\frac{1}{2}\sqrt{105}\sin\phi\cos\phi\right) + 5\sqrt{\frac{7}{5\cdot5}}\left(\frac{1}{2}\sqrt{15}\cos\phi\right)$$

$$= -\frac{3}{2}\sqrt{105}\sin^2\phi\cos\phi + \frac{1}{2}\sqrt{105}\cos\phi$$

$$= \frac{1}{2}\sqrt{\frac{7}{15}}\left(-45\sin^2\phi\cos\phi + 15\cos\phi\right)$$

Tesseral ALF Derivatives: Row 4 ($n=4$)

The example in the Lear document which corresponds to the following example contains a typographical error. The final result for the ALF should be

$$(\cos\phi\,P'_{4,1}) = -70\sin^4\phi + \frac{135}{2}\sin^2\phi - \frac{15}{2}$$

$$(\cos\phi\,\bar{P}'_{4,1}) = -4\sin\phi\,(\sec\phi\,\bar{P}_{4,1}) + 5\lambda_{n-1,m}(4,1)(\sec\phi\,\bar{P}_{3,1})$$

$$= -4\sin\phi\left(\frac{21}{2}\sqrt{\frac{5}{2}}\sin^3\phi - \frac{9}{2}\sqrt{\frac{5}{2}}\sin\phi\right)$$

$$+ 5\sqrt{\frac{3\cdot9}{5\cdot7}}\left(\frac{5}{2}\sqrt{\frac{21}{2}}\sin^2\phi - \frac{1}{2}\sqrt{\frac{21}{2}}\right)$$

$$= -21\sqrt{10}\sin^4\phi + 9\sqrt{10}\sin^2\phi + \frac{5\cdot5}{2}\sqrt{\frac{3\cdot9\cdot7\cdot3}{2\cdot5\cdot7}}\sin^2\phi$$

$$- \frac{5}{2}\sqrt{\frac{3\cdot9\cdot7\cdot3}{2\cdot5\cdot7}}$$

$$= -21\sqrt{10}\sin^4\phi + \frac{81}{2}\sqrt{\frac{5}{2}}\sin^2\phi - \frac{9}{2}\sqrt{\frac{5}{2}}$$

$$= 3\sqrt{\frac{1}{10}}\left(-70\sin^4\phi + \frac{135}{2}\sin^2\phi - \frac{15}{2}\right)$$

$$(\cos\phi\,\bar{P}'_{4,2}) = -4\sin\phi\,(\sec\phi\,\bar{P}_{4,2}) + 6\lambda_{n-1,m}(4,2)(\sec\phi\,\bar{P}_{3,2})$$

$$= -4\sin\phi\left(\frac{21}{4}\sqrt{5}\sin^2\phi\cos\phi - \frac{3}{4}\sqrt{5}\cos\phi\right)$$

$$+ 6\sqrt{\frac{2\cdot 9}{6\cdot 7}}\left(\frac{1}{2}\sqrt{105}\sin\phi\cos\phi\right)$$

$$= -21\sqrt{5}\sin^3\phi\cos\phi + 3\sqrt{5}\sin\phi\cos\phi$$

$$+ 3\sqrt{\frac{9\cdot 3\cdot 5\cdot 7}{3\cdot 7}}\sin\phi\cos\phi$$

$$= -21\sqrt{5}\sin^3\phi\cos\phi + 12\sqrt{5}\sin\phi\cos\phi$$

$$= \frac{1}{2}\sqrt{\frac{1}{5}}\left(-210\sin^3\phi\cos\phi + 120\sin\phi\cos\phi\right)$$

$$(\cos\phi\,\bar{P}'_{4,3}) = -4\sin\phi\left(\sec\phi\,\bar{P}_{4,3}\right) + 7\lambda_{n-1,m}(4,3)(\sec\phi\,\bar{P}_{3,3})$$

$$= -4\sin\phi\left(\frac{3}{2}\sqrt{\frac{35}{2}}\sin\phi\cos^2\phi\right) + 7\sqrt{\frac{9}{7\cdot 7}}\left(\frac{1}{2}\sqrt{\frac{35}{2}}\cos^2\phi\right)$$

$$= -3\sqrt{70}\sin^2\phi\cos^2\phi + \frac{3}{2}\sqrt{\frac{35}{2}}\cos^2\phi$$

$$= \frac{1}{2}\sqrt{\frac{1}{70}}\left(-420\sin^2\phi\cos^2\phi + 105\cos^2\phi\right)$$

4.3   Gottlieb Algorithm

4.3.1   Basis of the Gottlieb Approach

Mueller [10] developed an efficient algorithm for solving the gravitational potential function which, like Pines, defined special functions by reallocating the factors in each term of the potential function. Gottlieb [3] then defined the gradient of Mueller's potential function in terms of the partial derivatives with respect to these special functions. Recursions for the partial derivatives of the special functions were then developed to resolve the gravitational acceleration.

4.3.2   Normalized Gottlieb Algorithm

Each of the recursion equations of the Gottlieb algorithm is now normalized using the normalization parameters of Section 3.4. Each of the original equations can be found in Ref. [3]. Only the normalized equations are presented here and should replace their analogues in the original algorithm. The added normalization parameters are denoted with an underbrace or overbrace to bring attention to the changes from the equations in the original document.

Normalizing the Gottlieb recursions requires a few new normalization parameters in addition to the parameters derived for a normalized Lear implementation. The new normalization parameters are outlined in Table 4.2. In

$$\lambda_{n,m+1}(n,m) \quad = \quad \frac{N_{n,m}}{N_{n,m+1}} = \sqrt{\frac{(n+m+1)(n-m)(2-\delta_{0,m})}{2}} \quad (4.12)$$

$$\lambda_{n,m+1}(n,0) \quad = \quad \frac{N_{n,0}}{N_{n,1}} = \sqrt{\frac{(n+1)n}{2}} \quad (4.13)$$

$$\lambda_{n-1,m-1}(n,m) \quad = \quad \frac{N_{n,m}}{N_{n-1,m-1}}$$

$$= \quad \sqrt{\frac{(2n+1)(2-\delta_{0,m})}{(2n-1)(n+m)(n+m-1)(2-\delta_{0,m-1})}} \quad (4.14)$$

$$(4.15)$$

Table 4.2: Normalization parameters λ needed for a normalized Gottlieb algorithm

this section, $\sin\phi$ has been replaced by ϵ to keep a consistent notation with the original document.

Recursion for Zonal Legendre Polynomials $\bar{P}_n^0 = \bar{P}_n$

This recursion utilizes Parameters 4.1 and 4.2. For $n = 2$ through n_d

$$\bar{P}_n^0 = \bar{P}_n = \frac{1}{n}[(2n-1)\epsilon \underbrace{\lambda_{n-1}(n)}_{4.1} \bar{P}_{n-1} - (n-1)\underbrace{\lambda_{n-2}(n)}_{4.2} \bar{P}_{n-2}] \quad (4.16)$$

where $\bar{P}_0 = 1$ and $\bar{P}_1 = N_1 P_1 = \sqrt{3}\,\epsilon$.

Recursion for Sectorial and Tesseral ALFs $\bar{P}_n^m$

This recursion utilizes Parameters 4.5 and 4.14. For $n = 2$ through n_d and (inner loop) $m = 1$ through n_d

$$\bar{P}_n^m = \underbrace{\lambda_{n-2,m}(n,m)}_{4.5} \bar{P}_{n-2}^m + (2n-1)\underbrace{\lambda_{n-1,m-1}(n,m)}_{4.14} \bar{P}_{n-1}^{m-1} \quad (4.17)$$

where $\bar{P}_1^1 = 1$.

Recursion for Intermediate Sum H_n

This recursion utilizes Parameters 4.13 and 4.12. For $n = 2$ through n_d

$$H_n = C_{n,0} \overbrace{\lambda_{n,m+1}(n,0)}^{4.13} \bar{P}_n^1$$

$$+ \sum_{m=1}^{n} \underbrace{\lambda_{n,m+1}(n,m)}_{4.12} \frac{\bar{P}_n^{m+1}}{r^m}(C_{n,m}C_m + S_{n,m}S_m) \quad (4.18)$$

THIS PAGE IS INTENTIONALLY LEFT BLANK.

Chapter 5

Verifications and Conclusions

Each of the algorithms addressed in this paper were at first coded directly from the equations or code provided in the original sources and then tested against normalized implementations of each. Surprisingly, the results showed a fairly large discrepancy between the results of the normalized and unnormalized implementations for two of the three algorithms. Through numerical analysis and extremely scrutinized debugging, it became evident that the generator for the ALFs was the source of error.

Since the normalized Pines implementation was provided to the authors already coded but its results appeared more stable than the unnormalized code, it was suspected that the two codes were in some way different. As an experiment, the unnormalized Pines code was replaced by a copy of the normalized version of the code which had been "unnormalized," revealing that the ALF generator was in fact different between the two Pines implementations utilized in the first test. Further experimentation verified that the stability of the algorithms depended largely on the stability of the ALF generator used. Finally, the results of all testing appeared to indicate that normalization amplifies any inherent noise and error in each of the algorithms, a conclusion which further drove the development of additional conclusions and recommendations.

5.1   Preliminary Test Conditions

To test both the normalized and unnormalized implementations of each algorithm (six total) for agreement, acceleration vectors were computed at a set of positions around a given central body. The Moon was chosen for the central body, and the LP150Q mass coefficients were utilized for the test. The mass coefficients were unnormalized using the recursions in Section 3.2 to pass to the unnormalized algorithms. Tests utilized central-body-fixed position vectors with latitudes ranging from $-90°$ to $+90°$ in $30°$ increments and at each longi-

tude from $-150°$ to $+180°$ in $30°$ increments. Each position vector was given a 200-kilometer altitude above the lunar reference radius of 1738 kilometers.

At each of these positions, the acceleration was computed with all six algorithms with a variety of gravity model sizes. As a first check to ensure the algorithms functioned properly, a 0×0 model was tested to obtain the central body acceleration. Square models 2×2 through 50×50 were then tested, followed by the non-square models 50×0 through 50×49. Finally, the "extreme" cases of 125×125 and 150×150 were tested to ensure the normalized models in fact converged at relatively high degrees and orders.

The DeMars implementation of normalized Pines was considered the baseline model because it was based on the extensive stability studies of Lundberg and Schutz [5]. Error, defined for each tested iteration as the magnitude of the delta vector between the DeMars-calculated acceleration vector and the calculated accleration vectors from each of the other algorithms, was considered acceptable if the order of magnitude was 10^{-18} or smaller. This is the order of magnitude of ten times machine epsilon for the acceleration magnitudes tested, which was obtained by passing various acceleration vector magnitudes from the DeMars subroutine as arguments to the MATLAB `eps` function.

5.2   Preliminary Results

The output of the first MATLAB script to test the algorithms is listed in Appendix A. The preliminary results showed that Lear (both normalized and unnormalized) and Pines unnormalized had negligible error for all gravity models and locations around the central body. Gottlieb had small but noticeable error in equatorial locations in the square models 40×40 and larger. All of the Gottlieb non-square models had large but consistent error everywhere. The error became especially pronounced in the large 125×125 and 150×150 models at the equator. These findings necessitated further analysis of the error which developed at the equator.

5.3   Further Analysis

The tests outlined in Section 5.1 were modified to only calculate accelerations with all the square models from 2×2 through 150×150. This would allow a trend to emerge when plotting the error versus the degree and order of the model, as shown in Figure 5.1. At the equator, the error in both the normalized and unnormalized Gottlieb models grows slowly with increasing degree and order and suddenly diverges to positive infinity when the degree and order approaches 150. The rate of growth of error in Gottlieb is drastically larger as seen by comparing the scales of the y-axis of Figures 5.1 and 5.2. The beginning of the divergence of Gottlieb from the other two models, which remain closely in line with each other, in the unstable implementations can be clearly seen in Figure 5.3.

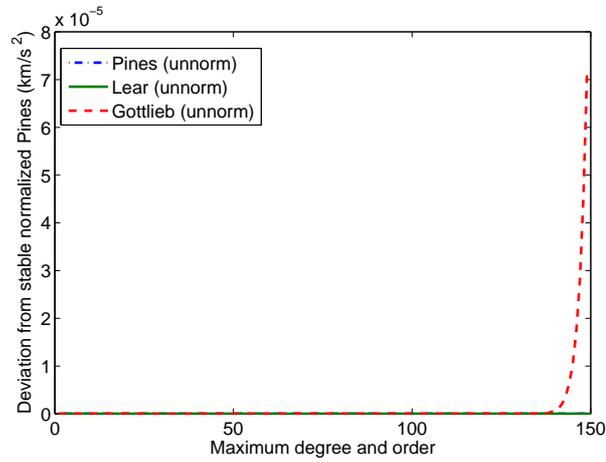As an experiment, this same test was run using a known unstable ALF

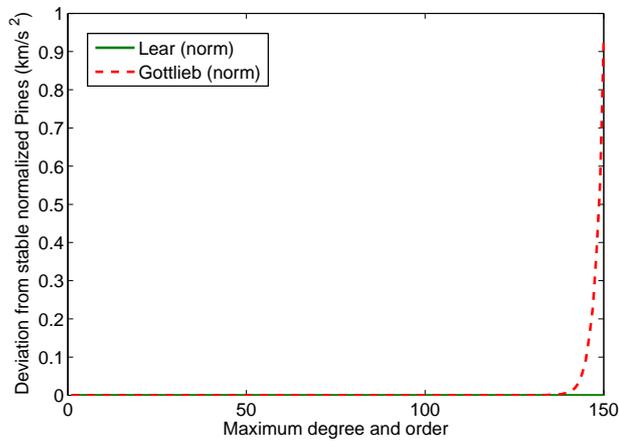Figure 5.1: Acceleration error magnitude for unnormalized models at equator $(\phi = 0°)$

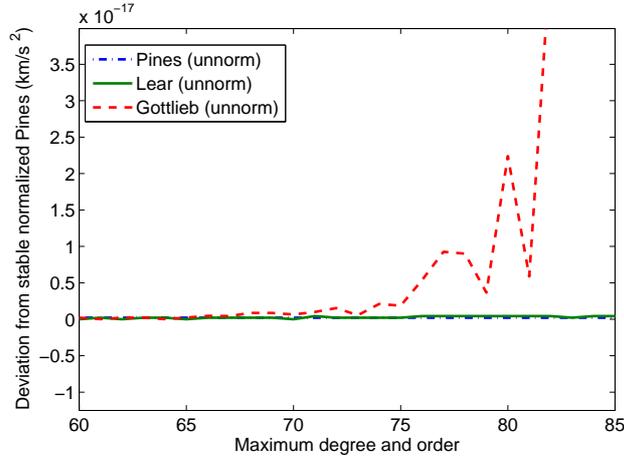Figure 5.2: Acceleration error magnitude for normalized models at equator $(\phi = 0°)$

Figure 5.3: Acceleration error magnitude for unnormalized models at equator ($\phi = 0°$) with degree and order 60–85

generator[1] in the Pines algorithm. The error behavior of this known unstable algorithm at the poles mirrored the behavior of the Gottlieb error at the equator, as shown in Figure 5.4.

Normalized implementations of the unstable Pines showed that the error was much larger than the error of the unnormalized unstable Pines, as seen in Figure 5.5. This large disparity between normalized and unnormalized error resembled the error of the two Gottlieb implementations as well.

5.4   Conclusions

Four primary conclusions can be drawn from the data presented in this paper:

- **Pines (as implemented by DeMars) and Lear algorithms are stable because they use a stable ALF recursion.** It is worth noting that virtually the same recursion equation [4, Recursion I] is used for generating ALFs in the unnormalized implementations of the Lear and Pines (DeMars) algorithms. The very similar behavior between the two algorithms can thus be explained by the similarity in their ALF generators.

- **Gottlieb and Pines algorithms, as originally published, are unstable due to unstable ALF recursions.** The apparently unstable behavior of the Gottlieb algorithm is presumed to be the result of an unstable ALF generator used in the algorithm. This conclusion is motivated by the similar signature of the error data with a known unstable ALF generator, the Pines algorithm with Unnormalized Recursion IV from Lundberg and

---
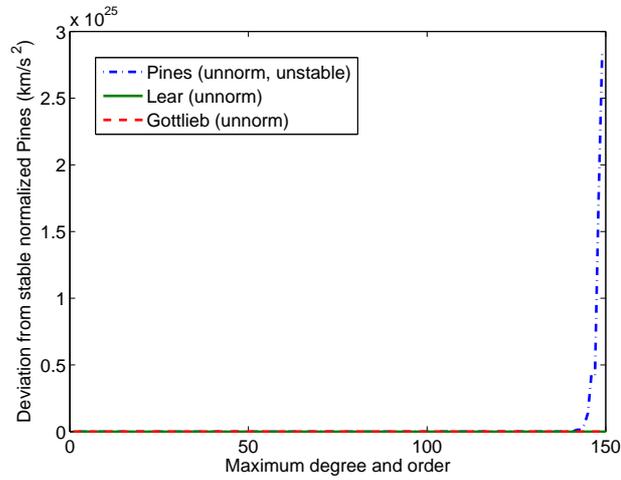
[1]Unnormalized Recursion IV from Lundberg and Schutz [4].

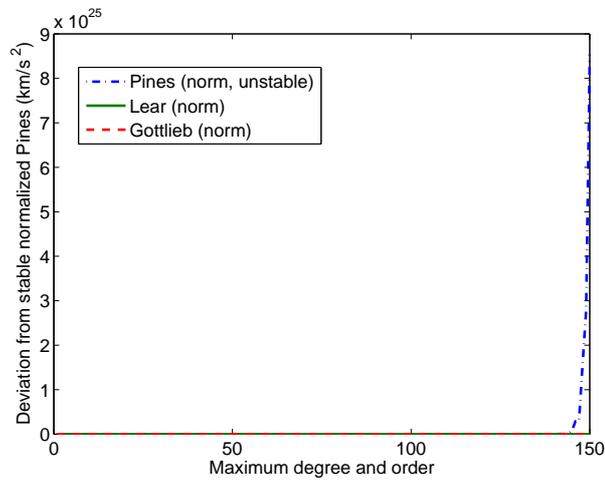Figure 5.4: Acceleration error magnitude for unnormalized models (unstable Pines) at south pole ($\phi = -90°$)

Figure 5.5: Acceleration error magnitude for normalized models (unstable Pines) at south pole ($\phi = -90°$)

Schutz [4] with which it was originally published. The error is location specific, much like the error that develops in the inherently unstable Pines implementation, albeit in a different location. The fact that both error signatures are latitude-dependent implies the ALF generator is the source of the instability, since the argument of ALFs used for spherical harmonic expansion is always $\sin \phi$.

- **Normalization of recursions amplifies numerical noise.** Assuming Gottlieb is, in fact, using an unstable ALF generator, it would appear that normalization of unstable ALF generators increases this instability and amplifies the error dramatically. This conclusion is supported by observing the same amplified error in both a known unstable ALF generator in Pines and the presumed-unstable ALF generator in Gottlieb in their normalized implementations relative to their unnormalized equivalents.

- **Unnormalized algorithms provide perfectly valid results at high degree and order as long as coefficients can be reliably unnormalized.** Normalized and unnormalized implementations of Pines as well as Lear algorithms agree very well with each other. This leads to the conclusion that it is safe to use unnormalized algorithms as long as proper unnormalization of the coefficients is performed, such as using the recursions in Section 3.2. If a relatively small gravity model is always desired, such as in current Mission Control Center software where the Earth gravity model is traditionally limited to degree and order 7, it is perfectly acceptable to continue the practice of implementing unnormalized algorithms for calculating gravitational acceleration.

5.5   Recommendations

Two recommendations are presented by the authors:

- **Gottlieb and Pines algorithms should not be implemented directly as published.** Unless a more stable ALF generation scheme is implemented, it is recommended that the Gottlieb algorithm be omitted from any implementation which incorporates large models due to the potential for instability. Developing an improved Gottlieb algorithm which implements a stable ALF generator is left for future work.

  The Pines algorithm has been stabilized by Lundberg and Schutz [4], so their recursions should be implemented with a Pines algorithm instead of the ALF generator used in the original Pines paper.

- **Normalized implementations are better suited to software packages than unnormalized algorithms.** Normalized algorithms are not only better suited for acceleration calculation with larger gravity models, their consistency with unnormalized algorithms makes them desirable for implementation in software packages seeking to maintain versatility

and robustness when computing gravitational acceleration. Normalized algorithms, when properly implemented, should always return a valid acceleration given any set of normalized coefficients and a valid degree and order.

THIS PAGE IS INTENTIONALLY LEFT BLANK.

Appendix A

Preliminary Results

This section contains the output of the initial MATLAB test script. These initial tests of the algorithms were used to identify test cases which needed further analysis. Cases with unusually large error are boxed $\boxed{\text{like this}}$. Note: Error is defined as the magnitude of the delta vector between each algorithm's acceleration vector and the Normalized Pines acceleration vector. See Section 5.1 for additional information.

```
Maximum central body unnormalized Pines error: 2.65574e-019
at lat/lon: -30/-60
degree x order: 0x0

Maximum central body unnormalized Lear error: 4.8487e-019
at lat/lon: 0/-120
degree x order: 0x0

Maximum central body unnormalized Gottlieb error: 6.50521e-019
at lat/lon: -90/-150
degree x order: 0x0

Maximum central body normalized Lear error: 0
at lat/lon: -90/-150
degree x order: 0x0

Maximum central body normalized Gottlieb error: 0
at lat/lon: -90/-150
degree x order: 0x0

Maximum square unnormalized Lear error: 1.99033e-018
at lat/lon: 0/30
degree x order: 48x48

Maximum square unnormalized Gottlieb error: 2.7959e-018
at lat/lon: 30/-150
degree x order: 49x49

Maximum non-square unnormalized Lear error: 2.28713e-018
at lat/lon: 30/-90
degree x order: 50x9
```

33

```
Maximum non-square unnormalized Gottlieb error:  8.29053e-008
```

at lat/lon: 60/-150
degree x order: 50x0

Maximum square Pines error: 2.65574e-019
at lat/lon: -30/-150
degree x order: 1x1

Maximum square Lear error: 2.48422e-019
at lat/lon: 60/60
degree x order: 46x46

```
Maximum square Gottlieb error:  1.32303e-016
```

at lat/lon: 0/-150
degree x order: 49x49

Maximum non-square Pines error: 4.96844e-019
at lat/lon: -60/-60
degree x order: 50x13

Maximum non-square Lear error: 2.65574e-019
at lat/lon: -30/150
degree x order: 50x25

```
Maximum non-square Gottlieb error:  1.43628e-016
```

at lat/lon: 0/-150
degree x order: 50x26

Maximum square normalized Lear error 125: 9.00606e-019
at lat/lon: 30/-90
degree x order: 125x125

```
Maximum square normalized Gottlieb error 125:  5.96756e-005
```

at lat/lon: 0/-120
degree x order: 125x125

Maximum square normalized Lear error 150: 2.7959e-018
at lat/lon: 30/150
degree x order: 150x150

```
Maximum square normalized Gottlieb error 150:  1.17375
```

at lat/lon: 0/-30
degree x order: 150x150

Bibliography

[1] Pines, S., "Uniform Representation of the Gravitational Potential and its Derivatives," *AIAA Journal*, Vol. 11, No. 11, Nov. 1973, pp. 1508–1511.

[2] Lear, W. M., "The Gravitational Acceleration Equations," JSC Internal Note 86-FM-15 (JSC-22080), NASA, April 1986.

[3] Gottlieb, R. M., "A Fast Recursive Singularity Free Algorithm for Calculating the First and Second Derivatives of the Geopotential," JSC Internal Note 89-FM-10 (JSC-23762), NASA, July 1990.

[4] Lundberg, J. B. and Schutz, B. E., "Recursion Formulas of Legendre Functions for Use with Nonsingular Geopotential Models," *Journal of Guidance, Control, and Dynamics*, Vol. 11, Jan.-Feb. 1988, pp. 31–38.

[5] DeMars, K. J., Bishop, R. H., Crain, T. P., and Condon, G. L., "Engineering Analysis of Guidance and Navigation Performance in the Uncertain Lunar Environment to Support Human Exploration," *Thirty-First Annual AAS Guidance and Control Conference*, No. AAS 08-046, American Astronautical Society, Breckenridge, CO, Feb. 2008.

[6] Tapley, B. D., Schutz, B. E., and Born, G. H., *Statistical Orbit Determination*, Elsevier, 2004.

[7] Schey, H. M., *Div, Grad, Curl, and All That: An Informal Text on Vector Calculus*, W. W. Norton, 4th ed., 2005.

[8] Vallado, D. A., *Fundamentals of Astrodynamics and Applications*, Microcosm Press/Springer, 3rd ed., 2007.

[9] Jahnke, E. and Emde, F., *Tables of Functions with Formulae and Curves*, Dover, 4th ed., 1945.

[10] Mueller, A. C., "A Fast Recursive Algorithm for Calculating the Forces Due to the Geopotential (Program: GEOPOT)," JSC Internal Note 75-FM-42 (JSC-09731), NASA, June 1975.