Autonomous Vehicle Simulation (AVS) Laboratory, University of Colorado

Basilisk Technical Memorandum

Document ID: Basilisk-SphericalHarmonics

SPHERICAL HARMONICS C++ MODEL

Prepared by	S. Carnahan

Status: Tested
Scope/Contents
The gravity effector module is responsible for calculating the effects of gravity from a body on a spacecraft. A spherical harmonics model and implementation is developed and described. A unit test has been written and run which tests basic input/output, single-body gravitational acceleration, and multi-body gravitational acceleration.

Rev	Change Description	By	Date
1.0	First version - Mathematical formulation and implementation	M. Diaz Ramos	20170101
1.1	Added test documentation	S. Carnahan	20170713

Contents

---

1 Model Description

The gravity effector module is responsible for calculating the effects of gravity from a body on a spacecraft. A spherical harmonics model and implementation is developed and described below. The iterative methods used for the software algorithms are also described. Finally, the results of the code unit tests are presented and discussed.

1.1 Mathematical model

1.1.1 Gravity models

Gravity models are usually based on solutions of the Laplace equation ($\nabla^2 U(\bar{\mathbf{r}}) = 0$). It is very important to state that this equation only models a gravity potential outside a body. For computing a potential inside a body the Poisson equation is used instead.

The spherical harmonic potential is a solution of the Laplace equation using orthogonal spherical harmonics. It can be derived solving the Laplace equation in spherical coordinates, using the separation of variables technique and solving a Sturm-Liouville problem. In this work, the solution will be found using another technique, which essentially follows Vallado's book.[5]

For each element of mass dm_{Q} the potential can be written as

$$\mathrm{d}U(\bar{\mathbf{r}}) = G\frac{\mathrm{d}m_{\mathrm{Q}}}{\rho_{\mathrm{Q}}} \tag{1}$$

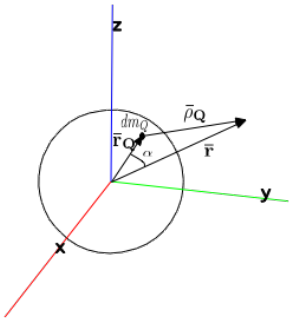

Fig. 1: Geometry of the Spherical Harmonics Representation.

where ρ_Q is the distance between the element of mass and the position vector $\bar{\mathbf{r}}$ where the potential is computed. This position vector is usually given in a body-fixed frame. The relation between the position vector $\bar{\mathbf{r}}$, the position of the element of mass $\bar{\mathbf{r}}_Q$ and ρ_Q can be given using the cosine theorem and the angle α between the two position vectors, as can be appreciated in Figure 1.

$$\rho_Q = \sqrt{r^2 + r_Q^2 - 2rr_Q\cos(\alpha)} = r\sqrt{1 - 2\frac{r_Q}{r}\cos(\alpha) + \left(\frac{r_Q}{r}\right)^2} = r\sqrt{1 - 2\gamma\cos(\alpha) + \gamma^2} \tag{2}$$

where $\gamma = r_Q/r$.

The potential can be obtained by integrating dU through the whole body.

$$U(\bar{\mathbf{r}}) = G\int_{body}\frac{\mathrm{d}m_Q}{r\sqrt{1 - 2\gamma\cos(\alpha) + \gamma^2}} \tag{3}$$

If the potential is computed outside the body, γ will always be less than 1, and the inverse of the square root can be approximated using the Legendre polynomials $P_l[\beta]$.[5] Even though this derivation does not use the Laplace equation, it still assumes that the potential is computed outside the body.

The Legendre polynomials can be written as

$$P_l[\beta] = \frac{1}{2^l l!}\frac{d^l}{d\beta^l}(\beta^2 - 1)^l \tag{4}$$

The potential is

$$U(\bar{\mathbf{r}}) = \frac{G}{r}\int_{body}\sum_{l=0}^{\infty}\gamma^l P_l[\cos(\alpha)]\mathrm{d}m_Q \tag{5}$$

The angle α must be integrated. However, the cosine of the angle α can be decomposed using the geocentric latitude and the longitude associated to vectors $\bar{\mathbf{r}}$ and $\bar{\mathbf{r}}_Q$. These angles will be called (ϕ, λ) and (ϕ_Q, λ_Q) respectively. Using the addition theorem it is possible to write.[5]

$$P_l[cos(\alpha)] = P_l[\sin(\phi_Q)]P_l[\sin(\phi)] + 2\sum_{m=1}^{l}\frac{(l-m)!}{(l+m)!}(a_{l,m}a'_{l,m} + b_{l,m}b'_{l,m}) \tag{6}$$

where

$$a_{l,m} = P_{l,m}[\sin(\phi_Q)]\cos(m\lambda_Q) \tag{7}$$
$$b_{l,m} = P_{l,m}[\sin(\phi_Q)]\sin(m\lambda_Q) \tag{8}$$
$$a'_{l,m} = P_{l,m}[\sin(\phi)]\cos(m\lambda) \tag{9}$$
$$b'_{l,m} = P_{l,m}[\sin(\phi)]\sin(m\lambda) \tag{10}$$

where $P_{l,m}[x]$ are the associated Legendre functions. "l" is called degree and "m", order. The polynomials can be computed as

$$P_{l,m}[\beta] = (1-\beta^2)^{\frac{m}{2}} \frac{d^m}{d\beta^m} P_l[\beta] \tag{11}$$

As can be seen, $a_{l,m}$ and $b_{l,m}$ must be integrated, but $a'_{l,m}$ and $a'_{l,m}$ can be taken outside the integral. Therefore, it is possible to define

$$C'_{l,m} = \int_{body} (2-\delta_m) r_{\mathsf{Q}}^l \frac{(l-m)!}{(l+m)!} a_{l,m} \mathrm{d}m_{\mathsf{Q}} \tag{12}$$

$$S'_{l,m} = \int_{body} (2-\delta_m) r_{\mathsf{Q}}^l \frac{(l-m)!}{(l+m)!} b_{l,m} \mathrm{d}m_{\mathsf{Q}} \tag{13}$$

where δ_m is the Kronecker delta.

Then

$$U(\bar{\mathbf{r}}) = \frac{G}{r} \sum_{l=0}^{\infty} C'_{l,0} \frac{P_l[\sin(\phi)]}{r^l} + \frac{G}{r} \sum_{l=0}^{\infty} \sum_{m=1}^{l} \frac{P_{l,m}[\sin(\phi)]}{r^l} \big[C'_{l,m} \cos(m\lambda) + S'_{l,m} \sin(m\lambda) \big] \tag{14}$$

Non-dimensional coefficients $C_{l,m}$ and $S_{l,m}$ are usually used

$$C'_{l,m} = C_{l,m} R_{\mathsf{ref}}^l m_{\mathsf{Q}} \tag{15}$$

$$S'_{l,m} =_{\mathsf{CoM}} S_{l,m} R_{\mathsf{ref}}^l m_{\mathsf{Q}} \tag{16}$$

where m_{Q} is the total mass of the body and R_{ref} is a reference radius. If the coefficients $C_{l,m}$ and $S_{l,m}$ are given, the reference radius must be specified. Usually, the reference is chosen as the maximum radius or the mean radius.[4]

The potential is then

$$U(\bar{\mathbf{r}}) = \frac{\mu}{r} \sum_{l=0}^{\infty} C_{l,0} \left(\frac{R_{\mathsf{ref}}}{r} \right)^l P_l[\sin(\phi)] + \frac{\mu}{r} \sum_{l=0}^{\infty} \sum_{m=1}^{l} \left(\frac{R_{\mathsf{ref}}}{r} \right)^l P_{l,m}[\sin(\phi)] \big[C_{l,m} \cos(m\lambda) + S_{l,m} \sin(m\lambda) \big] \tag{17}$$

Since $P_l[x] = P_{l,0}[x]$ the potential can be written in a more compact way

$$U(\bar{\mathbf{r}}) = \frac{\mu}{r} \sum_{l=0}^{\infty} \sum_{m=0}^{l} \left(\frac{R_{\mathsf{ref}}}{r} \right)^l P_{l,m}[\sin(\phi)] \big[C_{l,m} \cos(m\lambda) + S_{l,m} \sin(m\lambda) \big] \tag{18}$$

Some coefficients have a very interesting interpretation.

$$C_{0,0} = 1 \tag{19}$$

$$S_{l,0} = 0 \quad \forall l \geqslant 0 \tag{20}$$

$$C_{1,0} = \frac{Z_{\mathsf{CoM}}}{R_{\mathsf{ref}}} \tag{21}$$

$$C_{1,1} = \frac{X_{\mathsf{CoM}}}{R_{\mathsf{ref}}} \tag{22}$$

$$S_{1,1} = \frac{Y_{\mathsf{CoM}}}{R_{\mathsf{ref}}} \tag{23}$$

where $[X_{\text{CoM}}, Y_{\text{CoM}}, Z_{\text{CoM}}]$ represents the center of mass of the celestial body. Therefore, if the origin of the coordinate system coincides with the center of mass, all these coefficients are identically zero. Similarly, the second order coefficients are related to the second order moments (moments of inertia).

Finally, the coefficients and Legendre polynomials are usually normalized to avoid computational issues. The factor $N_{l,m}$ is called the normalization factor

$$N_{l,m} = \sqrt{\frac{(l-m)!(2-\delta_m)(2l+1)}{(l+m)!}} \tag{24}$$

The normalized coefficients are

$$\bar{C}_{l,m} = \frac{C_{l,m}}{N_{l,m}} \tag{25}$$

$$\bar{S}_{l,m} = \frac{S_{l,m}}{N_{l,m}} \tag{26}$$

The normalized associated Legendre functions are

$$\bar{P}_{l,m}[x] = P_{l,m}[x] N_{l,m} \tag{27}$$

The potential may be written as

$$U(\bar{\mathbf{r}}) = \frac{\mu}{r} \sum_{l=0}^{\infty} \sum_{m=0}^{l} \left(\frac{R_{\text{ref}}}{r}\right)^l \bar{P}_{l,m}[\sin(\phi)]\left[\bar{C}_{l,m}\cos(m\lambda) + \bar{S}_{l,m}\sin(m\lambda)\right] \tag{28}$$

1.1.2 Pines' Representation of Spherical Harmonics Gravity

There are many ways to algorithmically compute the potential and its first and secondary derivatives. One of such algorithms is the one proposed by Pines.[2]

The spherical harmonics representation as it was presented has a singularity at the poles for the gravity field. The Pines' formulation avoids this problem and is more numerically stable for high degree and high order terms.

Unfortunately, this formulation does not contain the normalization factor which is necessary if the coefficients are normalized. In a paper written by Lundberg and Schutz,[1] a normalized representation of the Pines' formulation is given, but it contains an approximation.

For this work, and in order to code the spherical harmonics formulation, a formulation similar to Pines' using the Lundberg-Schutz paper will be derived. However, no approximations will be used. Therefore, the algorithm will be developed here without using the exact formulations given in those papers. For the sake of brevity, not every single derivation will be carried out, but it is possible to get the results following the expressions obtained in this section.

In the Pines' formulation the radius and the director cosines are used as coordinates. The potential will be given as $U[r, s, t, u]$, where

$$r = \sqrt{x^2 + y^2 + z^2} \tag{29}$$

$$s = \frac{x}{r} \tag{30}$$

$$t = \frac{y}{r} \tag{31}$$

$$u = \frac{z}{r} \tag{32}$$

For a function of these coordinates, the dependance will be given using square brackets (e.g. $f[r,s,t,u]$).

Since $u = \sin(\phi) = \cos(90^\circ - \phi)$, it is possible to write

$$P_{l,m}[\sin(\phi)] = P_{l,m}[u] \tag{33}$$

The derived Legendre functions $A_{l,m}[u]$ are defined such that

$$P_{l,m}[u] = (1-u^2)^{\frac{m}{2}} A_{l,m}[u] \tag{34}$$

From the definition of $P_{l,m}$ (Equation 11), it is possible to write

$$A_{l,m}[u] = \frac{d^m}{du^m} P_l[u] = \frac{1}{2^l l!} \frac{d^{l+m}}{du^{l+m}} (u^2-1)^l \tag{35}$$

The term $(1-u^2)^{\frac{m}{2}}$ can be written as $(1-\sin^2(\phi))^{\frac{m}{2}} = |\cos(\phi)|^m = \cos^m(\phi)$.

If the complex number ξ is defined such that (j is the imaginary unit)

$$\xi = \cos(\phi)\cos(\lambda) + j\cos(\phi)\sin(\lambda) = \frac{x}{r} + j\frac{y}{r} = s + jt \tag{36}$$

it is possible to write

$$\xi^m = \cos^m(\phi) e^{jm\lambda} = (s+jt)^m \tag{37}$$

The following sequences may be defined

$$R_m[s,t] = Re\{\xi^m\} \tag{38}$$

$$I_m[s,t] = Im\{\xi^m\} \tag{39}$$

Putting all together, it is possible to write

$$U(\bar{\mathbf{r}}) = \frac{\mu}{r} \sum_{l=0}^{\infty} \sum_{m=0}^{l} \left(\frac{R_{\text{ref}}}{r}\right)^l A_{l,m}[u]\{C_{l,m}R_m[s,t] + S_{l,m}I_m[s,t]\} \tag{40}$$

In order to normalize the coefficients ($\bar{C}_{l,m}$ and $\bar{S}_{l,m}$) and the derived Legendre functions ($\bar{A}_{l,m} = N_{l,m}A_{l,m}$), each term is divided an multiplied by the normalization factor $N_{l,m}$. Then

$$U(\bar{\mathbf{r}}) = \frac{\mu}{r} \sum_{l=0}^{\infty} \sum_{m=0}^{l} \left(\frac{R_{\text{ref}}}{r}\right)^l \bar{A}_{l,m}[u]\{\bar{C}_{l,m}R_m[s,t] + \bar{S}_{l,m}I_m[s,t]\} \tag{41}$$

The sets $D_{l,m}[s,t]$, $E_{l,m}[s,t]$, and $F_{l,m}[s,t]$, are defined as

$$D_{l,m}[s,t] = \bar{C}_{l,m}R_m[s,t] + \bar{S}_{l,m}I_m[s,t] \tag{42}$$

$$E_{l,m}[s,t] = \bar{C}_{l,m}R_{m-1}[s,t] + \bar{S}_{l,m}I_{m-1}[s,t] \tag{43}$$

$$F_{l,m}[s,t] = \bar{S}_{l,m}R_{m-1}[s,t] - \bar{C}_{l,m}I_{m-1}[s,t] \tag{44}$$

The value $\rho_l[r]$ is also defined as

$$\rho_l[r] = \frac{\mu}{r}\left(\frac{R_{\text{ref}}}{r}\right)^l \tag{45}$$

The gravity potential may be finally computed as

$$U(\bar{\mathbf{r}}) = \sum_{l=0}^{\infty} \sum_{m=0}^{l} \rho_l[r]\bar{A}_{l,m}[u]D_{l,m}[s,t] \tag{46}$$

This is the final expression that will be used to compute the gravity potential.

1.1.3 Recursion Formulas

Several recursion formulas are needed in order to algorithmically implement the Pines' formulation. They will be given without proof, but they are easily derived using the definitions above.

- Recursion formula for $\rho_l[r]$
  Initial condition: $\rho_0[r] = \frac{\mu}{r}$
  $$\rho_l[r] = \rho \cdot \rho_{l-1}[r] \tag{47}$$
  where $\rho = R_{\text{ref}}/r$.

- Recursion formula for $R_m[s,t]$
  Initial condition: $R_0[s,t] = 1$
  $$R_m[s,t] = sR_{m-1}[s,t] - tI_{m-1}[s,t] \tag{48}$$

- Recursion formula for $I_m[s,t]$
  Initial condition: $I_0[s,t] = 0$
  $$I_m[s,t] = sI_{m-1}[s,t] + tR_{m-1}[s,t] \tag{49}$$

- Recursion formula for $\bar{A}_{l,m}[u]$
  From Equation (35), it is possible to see that
  $$A_{l,l}[u] = (2l-1)A_{l-1,l-1}[u] \tag{50}$$
  $$A_{l,l-1}[u] = uA_{l,l}[u] \tag{51}$$

There are several recursion formulas for computing Legendre polynomials $A_{l,m}[u]$, for $m < l-1$. The following formula, which is stable for high degrees,[1] will be used:

$$A_{l,m}[u] = \frac{1}{l-m}((2l-1)uA_{l-1,m}[u] - (l+m-1)A_{l-2,m}[u]) \tag{52}$$

Using Equations (50), (51), and (52), and the definition $\bar{A}_{l,m}[u] = N_{l,m}A_{l,m}[u]$, the following recursion formulas can be derived.

Initial condition: $\bar{A}_{0,0}[u] = 1$

The diagonal terms are computed as

$$\bar{A}_{l,l}[u] = \sqrt{\frac{(2l-1)(2-\delta_l)}{(2l)(2-\delta_{l-1})}}\bar{A}_{l-1,l-1}[u] \tag{53}$$

The low diagonal terms are then calculated as

$$\bar{A}_{l,l-1}[u] = u\sqrt{\frac{(2l)(2-\delta_{l-1})}{2-\delta_l}}\bar{A}_{l,l}[u] \tag{54}$$

Finally, for $l \geqslant (m+2)$, $N1_{l,m}$ and $N2_{l,m}$ are defined such that

$$N1_{l,m} = \sqrt{\frac{(2l+1)(2l-1)}{(l-m)(l+m)}} \tag{55}$$

$$N2_{l,m} = \sqrt{\frac{(l+m-1)(2l+1)(l-m-1)}{(l-m)(l+m)(2l-3)}} \tag{56}$$

and $\bar{A}_{l,m}[u]$ computed using

$$\bar{A}_{l,m}[u] = uN1_{l,m}\bar{A}_{l-1,m}[u] - N2_{l,m}\bar{A}_{l-2,m}[u] \tag{57}$$

1.1.4 Derivatives

The first order derivatives of many of the values given are necessary to compute the gravity field (second order derivatives are needed if the Hessian is to be computed).

It is easy to show that

$$\frac{\partial D_{l,m}}{\partial s}[s,t] = mE_{l,m}[s,t] \tag{58}$$
$$\frac{\partial D_{l,m}}{\partial t}[s,t] = mF_{l,m}[s,t] \tag{59}$$

$$\frac{d\rho_l}{dr}[r] = -\frac{(l+1)}{R_{\text{ref}}}\rho_{l+1}[r] \tag{60}$$

$$\frac{\partial R_m}{\partial s}[s,t] = mR_{m-1}[s,t] \tag{61}$$
$$\frac{\partial R_m}{\partial t}[s,t] = -mI_{m-1}[s,t] \tag{62}$$
$$\frac{\partial I_m}{\partial s}[s,t] = mI_{m-1}[s,t] \tag{63}$$
$$\frac{\partial I_m}{\partial t}[s,t] = mR_{m-1}[s,t] \tag{64}$$

$$\frac{d\bar{A}_{l,m}}{du}[u] = \frac{N_{l,m}}{N_{l,m+1}}\bar{A}_{l,m+1}[u] \tag{65}$$

The gravity field can be computed using all the equations given. However, the gradient of the potential is needed. As a change of variables was realized, the chain rule must be applied. In order to avoid filling up pages with math derivations, the results will be given. With patience, the following results can be obtained applying the chain rule and using all the derivatives given.

The gravity field can be computed as

$$\bar{\mathbf{g}} = (a_1[r,s,t,u] + s\cdot a_4[r,s,t,u])\hat{\mathbf{i}} + (a_2[r,s,t,u] + t\cdot a_4[r,s,t,u])\hat{\mathbf{j}} + (a_3[r,s,t,u] + u\cdot a_4[r,s,t,u])\hat{\mathbf{k}} \tag{66}$$

where

$$a_1[r,s,t,u] = \sum_{l=0}^{\infty}\sum_{m=0}^{l}\frac{\rho_{l+1}[r]}{R_{\text{ref}}}m\bar{A}_{l,m}[u]E_{l,m}[s,t] \tag{67}$$

$$a_2[r,s,t,u] = \sum_{l=0}^{\infty}\sum_{m=0}^{l}\frac{\rho_{l+1}[r]}{R_{\text{ref}}}m\bar{A}_{l,m}[u]F_{l,m}[s,t] \tag{68}$$

$$a_3[r,s,t,u] = \sum_{l=0}^{\infty}\sum_{m=0}^{l}\frac{\rho_{l+1}[r]}{R_{\text{ref}}}m\frac{N_{l,m}}{N_{l,m+1}}\bar{A}_{l,m+1}[u]D_{l,m}[s,t] \tag{69}$$

$$a_4[r,s,t,u] = \sum_{l=0}^{\infty}\sum_{m=0}^{l}\frac{\rho_{l+1}[r]}{R_{\text{ref}}}m\frac{N_{l,m}}{N_{l+1,m+1}}\bar{A}_{l+1,m+1}[u]D_{l,m}[s,t] \tag{70}$$

In order to avoid computing factorials, it is easy to see that

$$\frac{N_{l,m}}{N_{l,m+1}} = \sqrt{\frac{(l-m)(2-\delta_m)(l+m+1)}{2-\delta_{m+1}}} \tag{71}$$

$$\frac{N_{l,m}}{N_{l+1,m+1}} = \sqrt{\frac{(l+m+2)(l+m+1)(2l+1)(2-\delta_m)}{(2l+3)(2-\delta_{m+1})}} \tag{72}$$

Using all these expressions, the potential and the gravity field can be computed.

2 Model Functions

The mathematical description of gravity effects are implemented in gravityEffector.cpp. This code performs the following primary functions

- **GravBody Creation**: The code creates gravity bodies which are capable of affecting spacecraft. It does not effect a spacecraft unless that spacecraft explicitly adds the body as a gravity effector.
- **Orbital Energy**: The code can calculate the total orbital energy as well as orbital kinetic and orbital potential energy of a spacecraft about a gravity body.
- **Simple Gravity**: The code can compute a gravity acceleration between two bodies according to Newton's law of universal gravitation given μ and the distance between the bodies.
- **Spherical Harmonics**: The code can compute gravity acceleration between two bodies using the more-complex method of spherical harmonics. To do this, it must be provided with the same inputs as for calculating simple gravity. In addition, it needs to be provided a "degree" of spherical harmonics to be used and spherical harmonics coefficients useful up to that degree.
- **Multiple Body Effects**: The code can stack the effects of multiple gravity bodies on top of each other to determine the net effect on a spacecraft. The user must indicate in the spacecraft set-up which gravitational bodies should be taken into account.
- **Interface: Spacecraft States**: The code sends and receives spacecraft state information via the DynParamManager.
- **Interface: Energy Contributions**: The code sends spacecraft energy contributions via updateEnergyContributions() which is called by the spacecraft.
- **Interface: GravBody States**: The code outputs GravBody states(ephemeris information) via the Basilisk messaging system.

3 Model Assumptions and Limitations

The limitations of spherical harmonics gravity model are well-known and clearly explained in Schaub and Junkins' book.[3] The limitations include:

- **Coefficient Accuracy**: The coefficients used in the spherical harmonics equations are typically calculated based on gravitational data gathered by satellites. Therefore, the accuracy of the model is determined by the accuracy of the satellite instrumentation and precision of the stored data. Furthermore, for some bodies, there may not be sufficient information available to provide accurate coefficients or higher-degree coefficients.

- **Maximum Degree**: The spherical harmonics equation is a series expansion. Therefore, any implementation must truncate the equation at some point. The truncated portion of the equation necessarily defines some amount of error in the final calculation. This error is, however, small after the first handful of terms. Additionally, a larger distance between gravity body and spacecraft requires fewer terms of the series to achieve equal accuracy as compared to a case with less distance. This code allows the user to request a maximum number of terms to evaluate rather than a specific accuracy. This could lead to less-than-desirable accuracy with small separation distances and greater-than-necessary run times with large separation distances.
- **Planetary Ephemeris Data**: This code generally relies on an external package for planetary ephemeris information. Errors included in this package will translate into error in the gravity calculations, but those errors should be small. Because the ephemeris data is tabulated, this code should not be used to try to project the orbits of the celestial bodies in question. This could be done, though, by treating any celestial body as a "spacecraft".

4 Test Description and Success Criteria

The unit test, test_gravityDynEffector.py, validates the internal aspects of the Basilisk spherical harmonics gravity effector module by comparing module output to expected output. It utilizes spherical harmonics for calulcation given the gravitation parameter for the massive body, a reference radius, and the maximum degree of spherical harmonics to be used. The unit test verifies basic set-up, single-body gravitational acceleration, and multi-body gravitational acceleration.

4.1 Model Set-Up Verification

This test verifies, via three checks, that the model is appropriately initialized when called.

- 1.1 The first check verifies that the normalized coefficient matrix for the spherical harmonics calculations is initialized appropriately as a 3×3 identity matrix.
- 1.2 The second check verifies that the magnitude of the gravity being calculated is reasonable(i.e. between 9.7 and $9.9\ m/s^2$).
- 1.3 The final check ensures that the maximum degrees value is truly acting as a ceiling on the maximum number of degrees being used in the spherical harmonics algorithms. For example, if the maximum degrees value is set to 20, an attempt is made to call the spherical harmonics algorithms with a degrees value of 100 and another attempt is made with a degrees value of 20. The results are compared and should be equal due to the enforcement of the maximum degrees value.

4.2 Single-Body Gravity Calculations

This test compares calculated gravity values around the Earth with ephemeris data from the Hubble telescope. The simulation begins shortly after 0200 UTC May 1, 2012 and carries on for two hours, evaluating the gravitational acceleration at two second intervals.

4.3 Multi-Body Gravity Calculations

This test includes gravity effects from Earth, Mars, Jupiter, and the Sun. Calculations are verified by comparing against ephemeris data from the DAWN mission. The simulation begins at midnight, October 24th, 2008 and carries on for just under 22 minutes, evaluating the gravitational acceleration at two second intervals.

5 Test Parameters

This section summarizes the specific error tolerances for each test. Error tolerances are determined based on whether the test results comparison should be exact or approximate due to integration or other reasons. Error tolerances for each test are summarized in table 4.

Table 2: Error tolerance for each test.

Test	Tolerated Error
Gravity Magnitude (check 1.2)	1.0e-01
Single-Body Gravity	1.0e-04
Multi-Body Gravity	5.0e-04

6 Test Results

All checks within test_gravityDynEffector.py passed as expected. Table 3 shows the test results.

Table 3: Test results.

Test	Pass/Fail	Notes
Setup Test	PASSED	
Single-Body Gravity	PASSED	
Multi-Body Gravity	PASSED	

7 User Guide

This section contains descriptions of how the gravity effector code works and includes descriptions of and notes on variables. It should be helpful to users who wish to use the gravity effector module.

7.1 Code Diagram

The diagram in Fig. 2 demonstrates the basic iterative logic of the gravity effector module. There is extensive additional code that deals with things from the messaging system to transforming the spacecraft position from one frame to another. In general, the inputs shown must be given for each gravity body to be considered. There are, however, convenience functions which add the standard values of these inputs for common celestial bodies. An example is simIncludeGravity.addEarth() in the test_scenarioOrbitManeuver.py tutorial.

After the inputs are given for each gravity body, the computeGravityInertial() method calculates the 0^{th} degree gravity term for each body and its effects on the spacecraft. If useSphericalHarmParams is True for a given body, then computeField() is called to calculate higher degree spherical harmonics terms for that body.

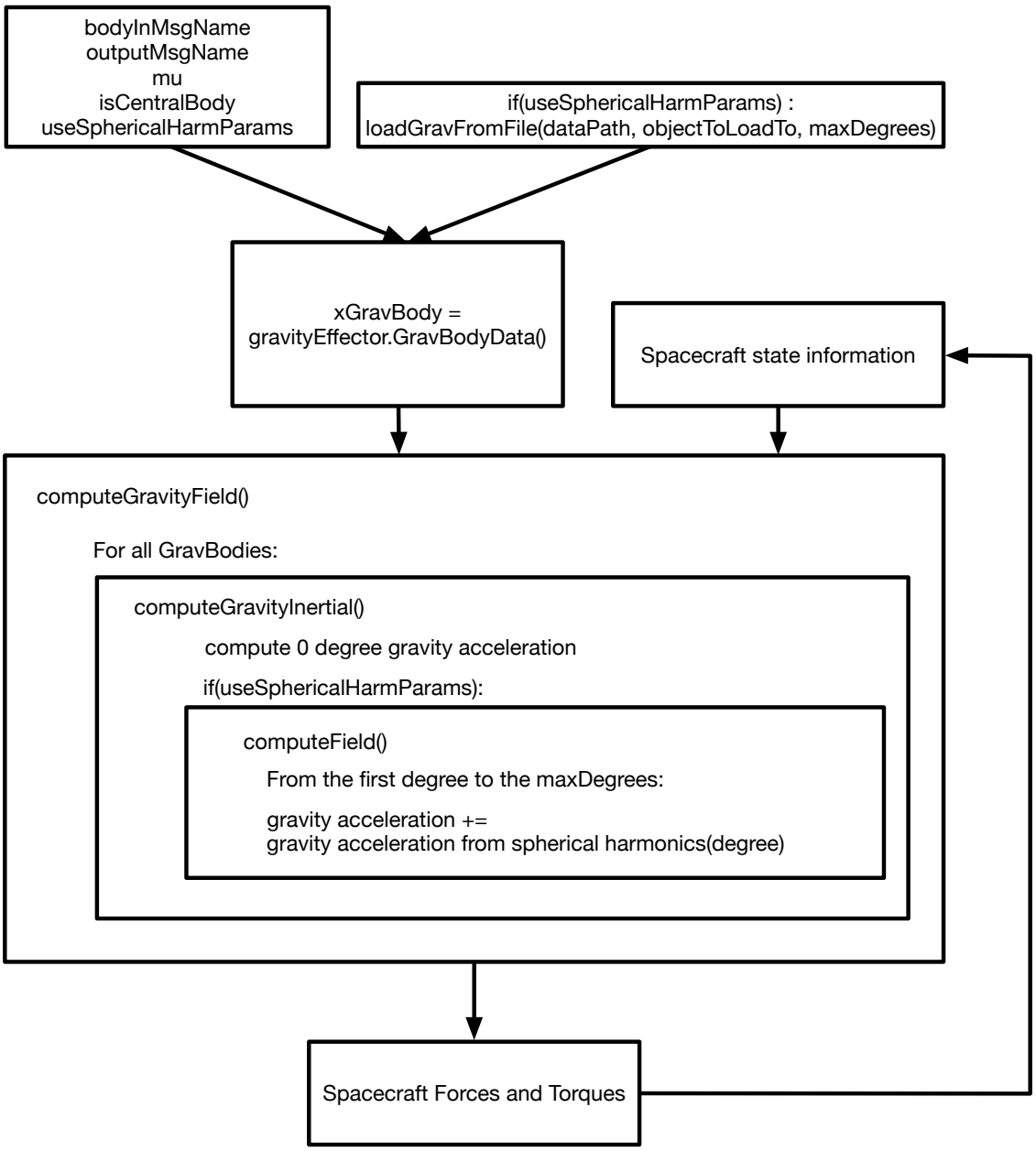

Fig. 2: A pseudo-code diagram demonstrating the flow of inputs and outputs in the gravity effector module.

7.2 Variable Definition and Code Description

The variables in Table 4 are available for user input. Variables used by the module but not available to the user are not mentioned here. Variables with default settings do not necessarily need to be changed by the user, but may be.

Table 4: Definition and Explanation of Variables Used.

Variable	LaTeX Equivalent	Variable Type	Notes
spherHarm.maxDeg	l_{max}	double	Default setting: 0"inertial_state number of degree to use when calculating gravity effects using sperical harmonics.
radEquator	R^l_{ref}	double	[m] Default setting: 0.0. This is the reference radius of the gravity body.
muBody	μ	double	[m3/s2] Default setting: 0.0f. This is the gravitational parameter of the body. Required Input to get any non-zero values out of the code.
isCentralBody	N/A	bool	Default setting: False. Determines whether the body in question is the central (inertial) body.
isDisplayBody	N/A	bool	Default setting: False. Determines whether the body in question is the focus of the visualization
ephemTime	N/A	double	[s] Default setting: 0. The ephemeris time for the body in question
ephIntTime	N/A	double	[s] Default setting: 0. Required Input. The integration time asociated with the ephem data.
bodyInMsgName	N/A	string	Required Input. The name of the message with gravity body data in the body frame.
outputMsgName	N/A	string	Required Input. The name of the message containing ephemeris information in display frame
planetEphemName	N/A	string	Required Input. An ephemeris name for the planet (user-named).

REFERENCES

[1] J. Lundberg and B. Schutz. Recursion formulas of legendre functions for use with nonsingular geopotential models. *Journal of Guidance AIAA*, 11(1):31–38, 1988.

[2] Samuel Pines. Uniform representation of the gravitational potential and its derivatives. *AIAA Journal*, 11(11):1508–1511, 1973.

[3] Hanspeter Schaub and John L. Junkins. *Analytical Mechanics of Space Systems*. AIAA Education Series, 3 edition, 2014.

[4] Daniel Scheeres. *Orbital Motion in Strongly Perturbed Environments*. Springer, 1 edition, 2012.

[5] David Vallado. *Fundamentals of Astrodynamics and Applications*. Microcosm press, 4 edition, 2013.