



Global Lunar Gravity Field Using Local Mascon Models

Sean McArdle¹ · Ryan P. Russell¹

Accepted: 25 July 2024

© The Author(s), under exclusive licence to American Astronautical Society 2024

Abstract

This work is the third in a series on point mascon lunar gravity models. Point mascon models are computationally-efficient replacements for the standard spherical harmonics gravity models used in astrodynamics applications. Weighted cubed-sphere mascon gravity models are introduced as runtime-efficient alternatives to the spherical harmonics representation that do not impose the extreme memory costs imposed by other interpolated gravity modeling schemes. Localized models for the lunar gravity field are generated using sets of point-mass potentials and referenced to a cubed-sphere grid. Adjacent localized point mascon gravity models are combined using Junkins weighting functions to form a smooth global model. Three demonstration models are generated that reproduce the GRGM1200A lunar gravity model with equivalent fidelity to degree 70, 300, and 550 spherical harmonics truncation levels. The weighted cubed-sphere models are benchmarked for runtime and memory cost against the standard spherical harmonics models and two recently-introduced types of globally-defined mascon models. The benchmarking results show that the weighted cubed-sphere mascon models enable significant runtime improvements with reasonable memory costs, especially when using OpenMP parallelization. Comparing acceleration runtime with spherical harmonics, the degree 300 equivalent weighted cubed-sphere model shows up to a 30-fold speedup with an 89 megabyte memory footprint and the degree 550 equivalent model shows up to a 90-fold speedup with a 170 megabyte memory footprint. Order of magnitude speedups are demonstrated without parallelization. The runtime models and driver code are available online.

Keywords Gravity modeling · Interpolation · Lunar gravity · Trajectory prediction

Sean McArdle and Ryan P. Russell have contributed equally to this work.

✉ Sean McArdle
seanmcardle14@gmail.com

Ryan P. Russell
ryan.russell@utexas.edu

¹ Aerospace Engineering, The University of Texas at Austin, 2617 Wichita Street, Austin, TX 78712, USA

1 Introduction

The goal of this research is to develop high-fidelity, computationally-efficient gravity field representations for large celestial bodies, with a particular focus on the Moon.³¹ Nonspherical gravity is especially important at the Moon for two main reasons: first, the lunar gravity field is uneven compared to the gravity fields of other large celestial bodies. Second, because the Moon has a negligible atmospheric density, objects can orbit extremely close to the lunar surface, where localized anomalies in the gravity field are especially impactful on motion. Nonspherical gravity is a critical effect to include in cislunar dynamics models, and is therefore useful for applications where such dynamics models are required, including mission design, orbit determination, and guidance navigation and control.

Spherical harmonics (SH) models are the most common representations for gravity fields in astrodynamics. Kaula provides a foundational text on the application of SH models to gravity modeling [11], and Fantino and Casotto survey a variety of modern SH model evaluation techniques [2]. SH models can be used in analytic orbit theories for trajectory prediction, such as the recent technique presented by Mahajan and Alfriend [13]. SH models are also commonly used to compute accelerations due to nonspherical gravity for numerically propagated trajectories. SH gravity models have a few notable drawbacks. The fidelity of a SH model is quantified by its truncation degree L , and the quantity of terms in the SH series increases quadratically with increasing degree. In the context of this work, SH models are referred to by their degree only, with the understanding that the full order is used in all cases. A study on appropriate SH truncation degree at the Moon presented in McArdle, et. al. [17] indicates that the appropriate SH truncation degree can be substantially greater than $L = 100$ at low lunar altitudes. This large recommended SH truncation degree agrees with analysis from a recent mission design study for the Sustained Low-Altitude Lunar Orbiter Mission (SLALOM), where a degree 400 SH truncation is found to be necessary for sufficient trajectory prediction accuracy in a 10 km lunar altitude orbit [19]. A degree $L = 400$ SH series contains about 160,000 terms. SH models require many recursive associated Legendre function calculations, and mitigating this large processing burden is non-trivial.

Interpolated gravity representations have been developed as more runtime-efficient alternatives to SH models. Junkins applied weighting functions to smoothly combine adjacent local gravity models [8–10]. Jones, Born, and Belkyn applied the cubed-sphere discretization and local B-splines to develop interpolated gravity models [7]. Arora and Russell adapted the Junkins weighting function approach with adaptive local polynomials for their Fetch interpolated gravity models [1]. One drawback of these interpolated gravity model approaches is that the memory footprints for high-resolution interpolated models are quite large, with the Fetch models (for the Earth) requiring more than two gigabytes to represent a gravity fidelity equivalent to a degree $L = 360$ SH truncation.

¹ Presented as Paper 22-827 at the AAS/AIAA Astrodynamics Specialist Conference, Charlotte, North Carolina, August 2022 [18]. Excerpts of this work are included in the dissertation by the author [14].

As another alternative to SH models, point mascon, or mass concentration, models reproduce the gravity field using an ensemble of point mass potentials buried under the surface of the celestial body. Earlier investigations, such as the study by Woodburn, et. al. successfully applied geometrically simple mass objects to model Earth's gravity [22]. One advantage of point mascon models is that they are easy to parallelize, since the contribution from each mascon can be computed independently at the evaluation point. Preceding articles in this series present *fixed-mass* models, where all the masses are equal and the gravity field is reproduced by selecting mascon locations [15], and *free-mass* models, where the gravity field is reproduced by selecting mascon masses [16]. Weighted cubed-sphere (WCS) mascon models are introduced in this current work as runtime-efficient gravity representations that incur a smaller memory cost than previous interpolated gravity models for the same fidelity. WCS models divide the sphere into local patches using the cubed-sphere discretization, and use Junkins weighting functions to smoothly join locally-accurate point mascon models. By concentrating mascons in the local region and distributing mascons more sparsely in distant locations, the local gravity field can be accurately modeled with significantly fewer mascons than would be required to model the global gravity field to the same fidelity.

Using mascons as the basis function for each local patch also has a physical significance: the inverse-squared law of gravity naturally dissipates in the radial direction, avoiding the need to discretize the space in the radial direction. The same set of mascons can achieve a desired level of accuracy over a range of altitudes, allowing the WCS mascon models to use a two-dimensional, rather than three-dimensional, discretization. The gravitational potential and its gradients can be recovered from a mascon model up to an arbitrary derivative order, so there is no need to represent the gradients with separate sets of parameters: only the potential function is modeled. For each evaluation of the WCS mascon model, four local mascon models are combined using Junkins weighting functions to ensure global smoothness [9]. As a serendipitous statistical result, combining four local models reduces error in the overall result, which allows fewer mascons to be used than would be necessary to represent the local gravity field using a single model.

During the development of the WCS gravity modeling technique, early design iterations were inspired by large-scale N -body gravity modeling methods for simulating large cosmological structures [5]. Fast multipole method (FMM) techniques were applied to consolidate mascons distant from a local region-of-interest. For the particular gravity field application presented in this current work, consolidating distant mascons into their center of mass caused unacceptably high errors in the region-of-interest, due mainly to a different scale of parallax when compared to the galaxy modelling application. Using higher-order multipole expansions of distant mascons did reduce the errors in the region-of-interest, but required a complicated two-step process: First the global gravity field would be modeled using mascons and then this global mascon model would be consolidated at each local patch using multipole expansions. The final configuration chosen for the local mascon modeling technique uses a simpler approach, where the local mascon models are fit directly to a reference SH model.

The organization of this paper is as follows: First the structure of the WCS point mascon models is explained in detail, including a description of the cubed-sphere discretization, the Junkins weighting functions, and the WCS model generation method. Then a set of three WCS demonstration models is presented, with a comparison of the accuracy of accelerations derived from the WCS models and SH models, an evaluation runtime analysis, and a trajectory propagation error study using the new WCS models. Finally concluding remarks are made with suggestions for future work and recommendations for use cases of all three types of lunar mascon models: global free-mass, global fixed-mass, and local WCS models. All the mascon models developed in this series, including demonstration WCS mascon models are publicly disseminated with Fortran driver code in an online database.²

2 Methods

In this section, the cubed-sphere discretization and the Junkins weighting function technique are explained, and the structure and generation technique for the local point mascon models are presented.

2.1 Cubed-Sphere Discretization

Locally-valid models are defined in regions of the sphere segmented by a cubed-sphere discretization [20]. The cubed-sphere discretization splits the sphere into patches with nearly equal areas. This discretization does not exhibit the severe distortions that arise from a naive spherical coordinate grid due to the singularities at the poles. The cubed-sphere patches are aligned across face boundaries, so it is possible to obtain smooth transitions between patches on adjacent faces.

The cubed-sphere discretization defines a grid on each of the six faces of a cube and projects these grids onto the surface of the sphere. The areas between grid lines are denoted patches, and the intersections of grid lines are denoted nodes. A location on the sphere can be fully defined in terms of cubed-sphere coordinates, where f is the face index, h is the horizontal patch index, v is the vertical patch index, $x_p \in [0, 1]$ is the local horizontal coordinate within the patch, and $y_p \in [0, 1]$ is the local vertical coordinate within the patch.

Figure 1 illustrates the patches defined by the cubed-sphere discretization. Figure 1a shows coarse patches to accentuate the shape of the cubed-sphere discretization, and Fig. 1b shows a much finer discretization representative of the gravity models presented in the current work. The parameter n_{cs} defines the discretization level. The number of patches spanning a face horizontally or vertically is

² The weighted cubed sphere mascon model database and accompanying driver software are available at <https://doi.org/10.5281/zenodo.5725120>.

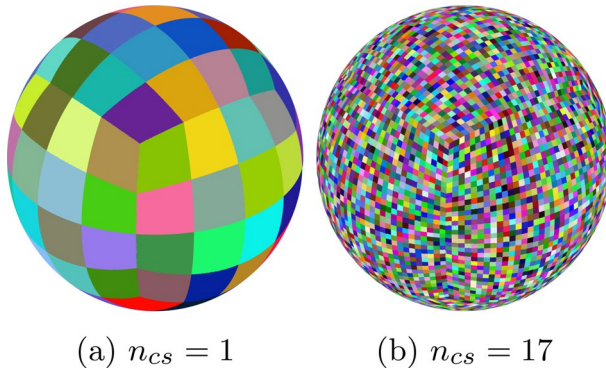


Fig. 1 Cube-sphere discretization

$$m_{cs} = 2(n_{cs} + 1). \tag{1}$$

The number of patches on a single face is m_{cs}^2 , and the total number of patches on the sphere is $6m_{cs}^2$. One node is assigned to each corner of a patch. Adjacent patches share nodes. The full sphere has $6m_{cs}^2 + 2$ unique nodes.

Algorithm 1 Mapping from Cartesian to cubed-sphere coordinates

Input
 n_{cs} : discretization level, $\mathbf{r} = [x, y, z]^T$: Cartesian position

Output
 f : face index, h : horizontal index, v : vertical index,
 x_p : local horizontal coordinate, y_p : local vertical coordinate

- 1: $f = \arg \min_i \|\mathbf{r} - M(i, :)\|$, $M = \begin{bmatrix} 0.5 & 0.0 & -0.5 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.5 & 0.0 & -0.5 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.5 & -0.5 \end{bmatrix}^T$
- 2: $x_B, y_B, z_B = N(f, :)$, $N = \begin{bmatrix} x & y & -x & -y & z & -z \\ y & -x & -y & x & y & y \\ z & z & z & z & -x & x \end{bmatrix}^T$
- 3: $x_C = x_B$, $y_C = z_B$
- 4: $\beta_h = \text{atan2}(y_B, x_B)$, $\beta_v = \text{atan2}(y_C, x_C)$
- 5: $x_t = (n_{cs} + 1)(\frac{4\beta_h}{\pi} + 1)$, $y_t = (n_{cs} + 1)(\frac{4\beta_v}{\pi} + 1)$
- 6: $h = \text{floor}(x_t)$, $v = \text{floor}(y_t)$
- 7: $x_p = x_t - h$, $y_p = y_t - v$

Algorithm 1 describes the procedure to map a point in Cartesian coordinates to a point in cubed-sphere coordinates. Starting with a position vector \mathbf{r} in Cartesian space, the face index is located by finding the nearest neighbor to six points defined in the positive and negative directions along each of the three axes. Once the face index is determined, the position vector is rotated into a coordinate frame

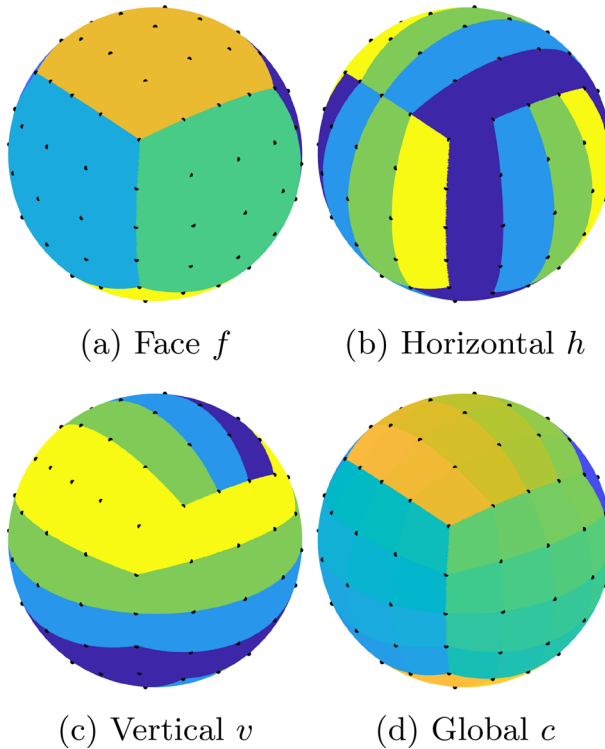


Fig. 2 Cube-sphere indices in Cartesian space ($n_{cs} = 1$)

aligned with the current face. Then the horizontal β_h and vertical β_v angles are calculated between the rotated position vector and the x_B axis. The horizontal and vertical angles are scaled such that the grid lines are located at integer distances from the edge of the face.

The specific implementation outlined in Algorithm 1 does include relatively expensive operations, including inverse trigonometric functions, and could likely be replaced with a faster alternative implementation in applications where the Cartesian to cubed-sphere mapping becomes a bottleneck. In the context of the WCS mascon models presented here, the implementation shown in Algorithm 1 has a negligible runtime compared to the local mascon model summation runtime.

Figures 2 and 3 illustrate the cubed-sphere discretization indices. The cubed-sphere indices are shown on a sphere in Cartesian space and on a longitude/latitude map for comparison. The colors on the Cartesian and longitude/latitude representations are consistent. The global index

$$c = m_{cs}^2 f + m_{cs} v + h \quad (2)$$

is calculated for visualization purposes. The node locations are indicated by black dots. Figures 2 and 3 demonstrate the alignment of the cube-sphere patch edges

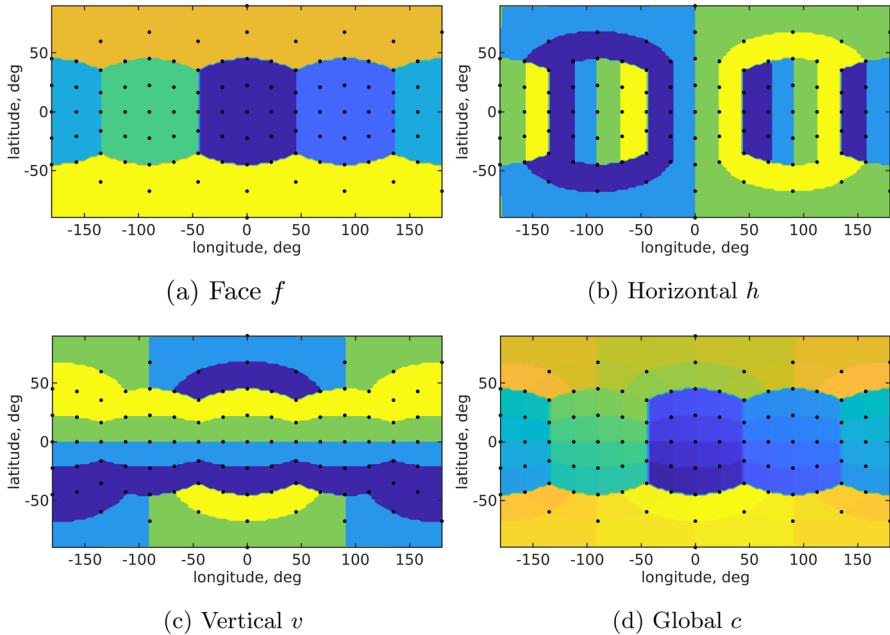


Fig. 3 Cube-sphere indices in spherical coordinates ($n_{cs} = 1$)

across each of the six faces. Each patch has eight adjacent patches, other than the patches abutting the cube corners, which have seven adjacent patches.

2.2 Local Model Evaluation Frame

A different local point mascon gravity model is assigned to each node. The gravitational parameters for the local mascon models are defined uniquely for each node, but each node shares a single set of local mascon locations in the *evaluation frame*. The evaluation frame is defined by rotation matrices from each node. The rotation matrix to rotate node center location $\hat{n} = [n_x, n_y, n_z]^T$ to the evaluation frame is provided by Frisvald [3]:

$$R_{cs} = \begin{bmatrix} 1 - \frac{n_x^2}{1+n_z} & -\frac{n_x n_y}{1+n_z} & -n_x \\ -\frac{n_x n_y}{1+n_z} & 1 - \frac{n_y^2}{1+n_z} & -n_y \\ n_x & n_y & n_z \end{bmatrix}. \tag{3}$$

The single set of mascon locations shared by all nodes is aligned with a notional North Pole in the evaluation frame.

The rotation matrix for each node is precomputed and stored when generating the model. This rotation matrix is valid for all node locations other than the node location at the South Pole, where the diagonal rotation matrix $R_{cs} = \text{diag}(1, 1, -1)$ is used. For an arbitrarily-fine discretization (very large n_{cs}) the evaluation of the

points closest to the South Pole may experience numerical instability due to the singularity in the $1/(1+n_z)$ factors of R_{cs} . This issue is not observed with the WCS models generated in this work.

Earlier iterations of the WCS mascon model technique stored unique positions for the mascons at each node. However, this approach resulted in an extreme memory cost, with finely-discretized models occupying memory footprints on the order of gigabytes. The mascon locations are consolidated to a single point distribution in the evaluation frame in order to dramatically reduce memory consumption.

2.3 Junkins Weighting Functions

The Junkins weighting functions are defined such that the modeled global function and its derivatives up to a specified order are continuous across patch and face boundaries [6, 9]. The Junkins weighting function method allows for an arbitrary number of dimensions, but the WCS mascon model approach only requires two dimensions.

Each weighting function is defined using the same base weight function, with transformed inputs based on which corner node the weighting function corresponds,

$$w_0^{(d)}(x_p, y_p) = w^{(d)}(x_p, y_p), \quad (4)$$

$$w_1^{(d)}(x_p, y_p) = w^{(d)}(1 - x_p, y_p), \quad (5)$$

$$w_2^{(d)}(x_p, y_p) = w^{(d)}(1 - x_p, 1 - y_p), \quad (6)$$

$$w_3^{(d)}(x_p, y_p) = w^{(d)}(x_p, 1 - y_p). \quad (7)$$

Figure 4 shows the placement of the node locations on a single patch and illustrates the local horizontal and vertical coordinates x_p and y_p . The local horizontal and vertical coordinates x_p and y_p span the valid patch area. The local horizontal and vertical coordinates are $x_p = 0$ and $y_p = 0$ at the origin and $x_p = 1$ and $y_p = 1$ at the top right corner of the patch area, as oriented in Fig. 4. The four weighting functions are defined such that the sum of the weighting functions is unity over the entire area of the patch and that the weighting functions and their derivatives up to the specified continuity order d are zero at the two edges opposite to the weighting functions' corners.

The weighting function for continuity order d in one dimension is [6],

$$w^{(d)}(\tilde{x}) = \frac{(2d+1)!(-1)^d}{(d!)^2} \sum_{r=0}^d \frac{(-1)^r \binom{d}{r}}{2d-r+1} \tilde{x}^{2d-r+1}, \quad (8)$$

where the relation between \tilde{x} and x_p is node-dependent according to the input transformations specified in Eqs. 4-7, or:

$$\text{Node 0: } \tilde{x} = x_p, \quad (9)$$

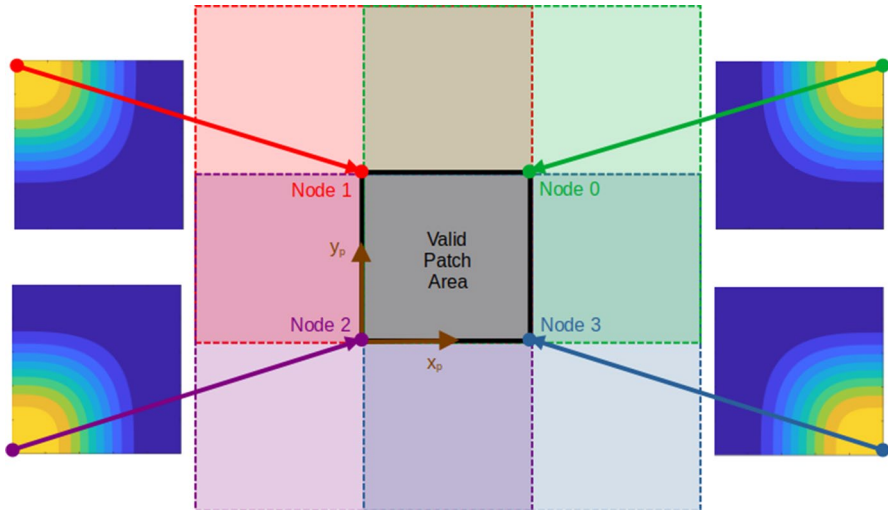


Fig. 4 Diagram of locally-valid region for Junkins weighting functions

$$\text{Node 1: } \tilde{x} = 1 - x_p, \tag{10}$$

$$\text{Node 2: } \tilde{x} = 1 - x_p, \tag{11}$$

$$\text{Node 3: } \tilde{x} = x_p. \tag{12}$$

Eq. 8 has a corresponding equation for \tilde{y} , where the relation between \tilde{y} and y_p is similarly node-dependent according to the input transformations specified in Eqs. 4–7, or:

$$\text{Node 0: } \tilde{y} = y_p, \tag{13}$$

$$\text{Node 1: } \tilde{y} = y_p, \tag{14}$$

$$\text{Node 2: } \tilde{y} = 1 - y_p, \tag{15}$$

$$\text{Node 3: } \tilde{y} = 1 - y_p. \tag{16}$$

The weighting function for continuity order d in two dimensions is simply the product of the one-dimensional weighting functions,

$$w^{(d)}(\tilde{x}, \tilde{y}) = w^{(d)}(\tilde{x})w^{(d)}(\tilde{y}). \tag{17}$$

The four contour plots in Fig. 4 illustrate the Junkins weighting functions for the four nodes of a patch. Note how the value of the weighting function is unity at each weighting function’s corner, and the value of the weighting function is zero at the

Table 1 Weighting functions for various continuity orders d

d	$w^{(d)}(\tilde{x}, \tilde{y})$
0	$\tilde{x}\tilde{y}$
1	$\tilde{x}^2\tilde{y}^2(-2\tilde{x} + 3)(-2\tilde{y} + 3)$
2	$\tilde{x}^3\tilde{y}^3(6\tilde{x}^2 - 15\tilde{x} + 10)(6\tilde{y}^2 - 15\tilde{y} + 10)$
3	$\tilde{x}^4\tilde{y}^4(-20\tilde{x}^3 + 70\tilde{x}^2 - 84\tilde{x} + 35)(-20\tilde{y}^3 + 70\tilde{y}^2 - 84\tilde{y} + 35)$

opposite edges from each weighting functions’ corner. The consequence of this weighting approach is that the global function is exactly equivalent to each local function when the evaluation point is coincident with the local function node. Also, the modeled function and its gradients up to order d smoothly transition across patch and face boundaries. The smooth transitions across patch and face boundaries also occur at the cubed-sphere face corners, as demonstrated in Sect. 3.2.

Table 1 shows explicit formulas for two-dimensional Junkins weighting functions with various continuity orders d . In the context of WCS mascon gravity models, the runtime overhead from evaluating the Junkins weighting functions is negligible compared to the runtime required to evaluate the local mascon model summations. The maximum derivative order desired for applications of the weighted cubed sphere model is three, so the $d = 3$ continuity level is used unless otherwise specified.

2.4 Evaluating the Global Gravity Field

The global gravitational potential within a patch is expressed as a weighted sum of four local potential functions $V_i(x, y, z)$, with each local potential function assigned to one of the four corners of the patch,

$$V(x, y, z) = \sum_{i=0}^3 w_i^{(d)}(x_p, y_p) V_i(x, y, z). \tag{18}$$

For the WCS mascon models, the local potential function is computed for each node using the summation for individual mascon models presented in References [14] and [16]. This local potential computation is performed after rotating into the evaluation frame using R_{cs} :

$$V_i(\mathbf{r}) = V_i^e(R_{cs}\mathbf{r}), \tag{19}$$

where $V_i^e(\cdot)$ is the local potential computed in the evaluation frame.

The gradient of the global potential is found by applying the chain rule to Eq. (18),

$$\frac{dV}{dx} = \sum_{i=0}^3 \left[\frac{dw_i^{(d)}}{dx} V_i + w_i^{(d)} \frac{\partial V_i}{\partial x} \right], \tag{20}$$

Table 2 First-order \tilde{x} derivatives of weighting functions

d	$\frac{\partial w_i^{(d)}(\tilde{x}, \tilde{y})}{\partial \tilde{x}}$
0	\tilde{y}
1	$-6\tilde{x}(\tilde{x} - 1)\tilde{y}^2(-2\tilde{y} + 3)$
2	$6\tilde{x}^2(\tilde{x} - 1)(2\tilde{x}^2 + 5\tilde{x} - 5)\tilde{y}^3(6\tilde{y}^2 - 15\tilde{y} + 10)$
3	$-140\tilde{x}^3(\tilde{x} - 1)^3\tilde{y}^4(-20\tilde{y}^3 + 70\tilde{y}^2 - 84\tilde{y} + 35)$

$$\frac{dV}{dy} = \sum_{i=0}^3 \left[\frac{dw_i^{(d)}}{dy} V_i + w_i^{(d)} \frac{\partial V_i}{\partial y} \right], \tag{21}$$

$$\frac{dV}{dz} = \sum_{i=0}^3 \left[\frac{dw_i^{(d)}}{dz} V_i + w_i^{(d)} \frac{\partial V_i}{\partial z} \right]. \tag{22}$$

Expressions for higher-order gradients of WCS models are found in Ref. [14] and included in the online code (see footnote in the introduction). The higher order gradients are useful for applications requiring first and second order partials of the acceleration. As with the computation of the local potential function, the computation of the local gradient function must be preceded by rotating the position into the node evaluation frame using R_{cs} . Additionally, the local gradient must be rotated from the node evaluation frame back to the original frame using R_{cs}^T :

$$\left[\frac{\partial V_i}{\partial x} \quad \frac{\partial V_i}{\partial y} \quad \frac{\partial V_i}{\partial z} \right]^T = R_{cs}^T \nabla V_i^e(R_{cs} \mathbf{r}). \tag{23}$$

The first-order total derivatives of the weighting functions in terms of the global coordinates are

$$\frac{dw_i^{(d)}}{dx} = \frac{\partial w_i^{(d)}}{\partial x_p} \frac{\partial x_p}{\partial x} + \frac{\partial w_i^{(d)}}{\partial y_p} \frac{\partial y_p}{\partial x}, \tag{24}$$

$$\frac{dw_i^{(d)}}{dy} = \frac{\partial w_i^{(d)}}{\partial x_p} \frac{\partial x_p}{\partial y} + \frac{\partial w_i^{(d)}}{\partial y_p} \frac{\partial y_p}{\partial y}, \tag{25}$$

$$\frac{dw_i^{(d)}}{dz} = \frac{\partial w_i^{(d)}}{\partial x_p} \frac{\partial x_p}{\partial z} + \frac{\partial w_i^{(d)}}{\partial y_p} \frac{\partial y_p}{\partial z}. \tag{26}$$

Table 2 shows the values of the first-order partial derivatives for the weighting functions in terms of the local coordinates.

The derivatives in Table 2 are shown in terms of the local horizontal coordinates. The corresponding expressions for the local vertical coordinates are similar, switching \tilde{x} and \tilde{y} . In order to obtain the full derivatives of the weighting function, the chain rule must be applied to each node:

Table 3 First-order derivatives of local coordinates, f is face index

f	0	1	2	3	4	5
$\frac{\pi}{4(n_{cs}+1)} \frac{\partial x_p}{\partial x}$	$-\frac{y}{x^2+y^2}$	$-\frac{y}{x^2+y^2}$	$-\frac{y}{x^2+y^2}$	$-\frac{y}{x^2+y^2}$	0	0
$\frac{\pi}{4(n_{cs}+1)} \frac{\partial x_p}{\partial y}$	$\frac{x}{x^2+y^2}$	$\frac{x}{x^2+y^2}$	$\frac{x}{x^2+y^2}$	$\frac{x}{x^2+y^2}$	$\frac{z}{y^2+z^2}$	$-\frac{z}{y^2+z^2}$
$\frac{\pi}{4(n_{cs}+1)} \frac{\partial x_p}{\partial z}$	0	0	0	0	$-\frac{y}{y^2+z^2}$	$\frac{y}{y^2+z^2}$
$\frac{\pi}{4(n_{cs}+1)} \frac{\partial y_p}{\partial x}$	$-\frac{z}{x^2+z^2}$	0	$\frac{z}{x^2+z^2}$	0	$-\frac{z}{x^2+z^2}$	$-\frac{z}{x^2+z^2}$
$\frac{\pi}{4(n_{cs}+1)} \frac{\partial y_p}{\partial y}$	0	$-\frac{z}{y^2+z^2}$	0	$\frac{z}{y^2+z^2}$	0	0
$\frac{\pi}{4(n_{cs}+1)} \frac{\partial y_p}{\partial z}$	$\frac{x}{x^2+z^2}$	$\frac{y}{y^2+z^2}$	$-\frac{x}{x^2+z^2}$	$-\frac{y}{y^2+z^2}$	$\frac{x}{x^2+z^2}$	$\frac{x}{x^2+z^2}$

$$\frac{\partial w_i^{(d)}}{\partial x_p} = \frac{\partial w^{(d)}(\tilde{x}, \tilde{y})}{\partial \tilde{x}} \frac{\partial \tilde{x}}{\partial x_p}, \tag{27}$$

where the relationship between \tilde{x} , \tilde{y} and x_p , y_p depends on node index i , specified in Eqs. (4–7).

Table 3 shows the partial derivatives of the local coordinates with respect to the global Cartesian coordinates. Since each face is oriented differently, the derivatives of the local coordinates are different depending on the current face index f . All the derivative expressions in this section are verified using complex step differentiation.

2.5 Local Point Mascon Model Generation

The local mascon model generation technique presented here leads to models that perform well in terms of evaluation runtime compared to SH models with equivalent fidelity. However, this particular approach to mascon placement and weighting requires a moderate amount of manual tuning and is not guaranteed to be the most efficient allocation of mascons.

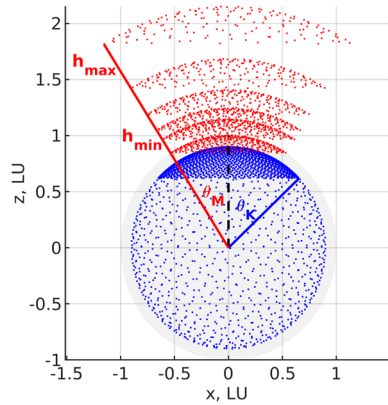
The positions of the mascons are defined such that there is a dense concentration of mascons on a spherical cap centered on the node location and a sparse concentration of mascons outside this spherical cap. Figure 5 illustrates the layout of the local mascon distribution. Note that the hypothetical model shown in Fig. 5 uses exaggerated spherical cap sizes to better illustrate the setup. Many variations of the final placement strategy were considered, including multiple levels of decreasing density. In the end, the selected two-level approach performed well and was the simplest to implement.

The size of the mascon spherical cap is defined by angle

$$\theta_K = \cos^{-1} \left(1 - \frac{\alpha_K}{3m_{cs}^2} \right). \tag{28}$$

This heuristic method to choose the spherical cap size adjusts the area of the spherical cap based on the cubed-sphere patch size. The quantity α_K is a tuning parameter that determines how much area to occupy with the spherical cap.

Fig. 5 Local mascon (blue) and measurement locations (red) (Color figure online)



Choosing $\alpha_K = 4.0$ sizes the spherical cap such that it will cover roughly the same area as four patches, which is the locally-valid area for each node (other than the corner nodes). The mascon positions within the spherical cap are chosen according to a spiral distribution with K_{near} total points, and the mascon positions outside the spherical cap are chosen according to a spiral distribution with K_{far} points. The spiral distribution is a nearly-uniform distribution of points on a sphere [21].

The local mascon masses are defined by solving a minimum-norm linear system consisting of simulated gravitational potential measurements and direct-fit Stokes coefficients as outlined in Ref. [14]. The gravitational potential measurements are evaluated at specified locations using the GRGM1200A [4, 12] SH model with truth degree $L_{t,gen}$ and a selection of Stokes coefficients up to degree L_f . The truth degree L_t is used to perform the altitude vs. error and trajectory error studies.

The first set of gravitational potential measurement locations are distributed on the surface according to a spiral distribution with M_{near} total points. Similar to the local mascon placement, the simulated measurements are placed on a dense spherical cap defined by an angle θ_M defined by Eq. (28), substituting α_M for α_K as a tuning parameter, and θ_M for θ_K as the resulting angle. Additional sets of gravitational potential measurement locations are placed on n_h spherical caps with altitudes from h_{min} to h_{max} using equal log-spacing. The measurement locations above the surface are distributed using spiral distributions with

$$M_h = \text{floor}\left(\frac{M_{near}}{(h/1738.0 + 1)^{v_h}}\right) \tag{29}$$

total points where h is the altitude measured in kilometers, and v_h is a tuning parameter that controls how quickly the number of measurements is reduced with increasing altitude.

The mascon and measurement locations for each node are determined by rotating the base mascon and measurement locations to center on the node

location using Eq. (3). The minimum-norm systems at each node are solved using the Intel Math Kernel Library (MKL) implementation of LAPACK's `dgelsd`, which solves the system using singular value decomposition. The effective rank of the system is defined using a minimum singular value of $\text{rcond}=1.0 \times 10^{-16}$. The solution process is parallelized over each node using Message Passing Interface (MPI) parallelization, allowing even the largest of the three demonstration models to be completed within one hour of wall-clock time.

2.6 Model Naming Convention

The WCS model naming convention is given by,

$$\{\text{Reference Model}\}_{\text{WCS}}\{L_e\}, \quad (30)$$

where L_e refers to the degree of the equivalent SH model. For example, a WCS model corresponding to a degree 70 truncation of the GRGM1200A reference model would be denoted GRGM1200A_WCS_70.

The equivalent size of SH model is determined using the error analysis described in Ref. [17] and Sect. 3.3 of Ref. [14]. This error analysis includes the acceleration error from the published uncertainties of the included Stokes coefficients, denoted commission error, and the acceleration error from the Stokes coefficients that have been truncated from the full model, denoted omission error. The WCS model's acceleration error averaged over the reference sphere is less than the averaged commission and omission error of its equivalent SH model truncation.

3 Results

Three example WCS gravity models for the Moon are presented and compared to equivalent SH models in sequential and CPU-parallelized computing environments.

3.1 Demonstration Models

Table 4 shows the tuning parameters selected for each of the three WCS models presented in this work. The discretization level $n_{cs} = 17$ is selected for all three models.

The three demonstration models are configured to have equivalent fidelity to $L_e = 70, 300,$ and 550 degree SH models. For the three demonstration models, Table 5 reports the number of mascons and gravitational potential measurements used by the local mascon representations. The number of equivalent SH coefficients divided by four times the number of local mascons is a rough proxy for the WCS runtime improvement over SH if no overhead existed for the WCS models. Figure 6 shows the actual distributions of the mascons and gravitational potential measurement locations for each demonstration model in the evaluation frame.

Table 4 Tuning parameters for weighted cubed-sphere models

Equivalent degree	L_e	70	300	550
Truth degree for error analysis	L_t	140	600	1100
Truth degree during model generation	$L_{t,gen}$	84	360	660
Direct-fit degree	L_f	7	18	22
Mascon cap density	K_{near}	14112	194400	544500
Mascon shell density	K_{far}	196	600	1210
Mascon shell radius (LU)	r_k	0.92	0.975	0.985
Mascon cap area	α_K	52.0	36.0	24.0
Measurement cap density	M_{near}	14112	77760	174240
Measurement cap area	α_M	11.0	9.0	8.0
Quantity of measurement caps	n_h	11	10	10
Min. measurement altitude (km)	h_{min}	200.0	10.0	10.0
Max. measurement altitude (km)	h_{max}	8107.0	1710.0	1279.2
Measurement quantity altitude modifier	v_h	0.8	2.0	7.0

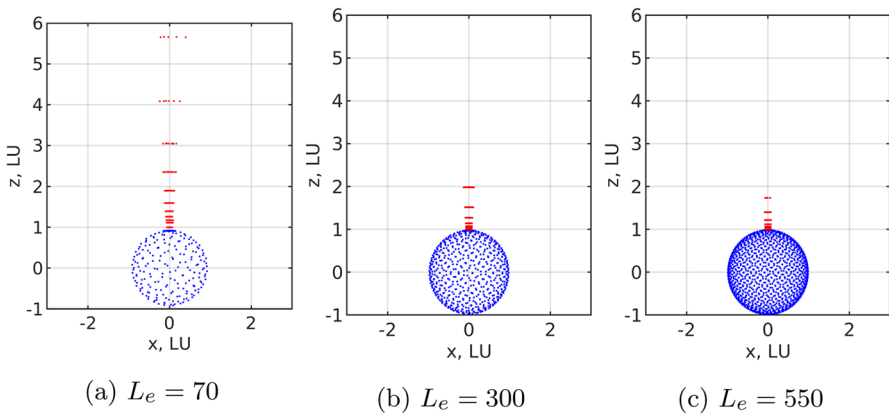


Fig. 6 Weighted cubed-sphere mascons (blue) and measurements (red) (Color figure online)

Table 5 Weighted cubed-sphere model summary

L_e	70	300	550
Local Mascons K	289	1497	2887
Local Measurements M	143	701	1077
Memory (MB)	17.6	88.6	170.4
Equivalent SH Coefficients	5041	90601	303601

3.2 Smoothness Demonstration

Figures 7, 8, 9, 10 demonstrate the continuity across patch boundaries enabled using the Junkins weighting function approach by showing the value of the gravitational

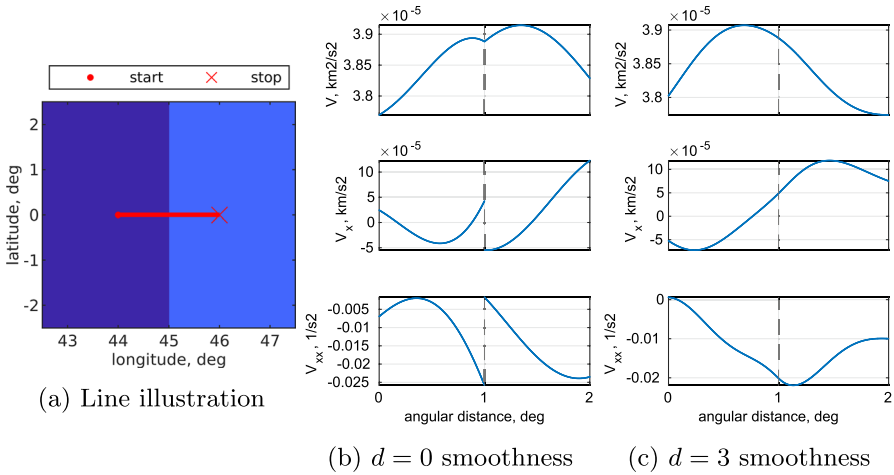


Fig. 7 Smoothness across vertical face boundary

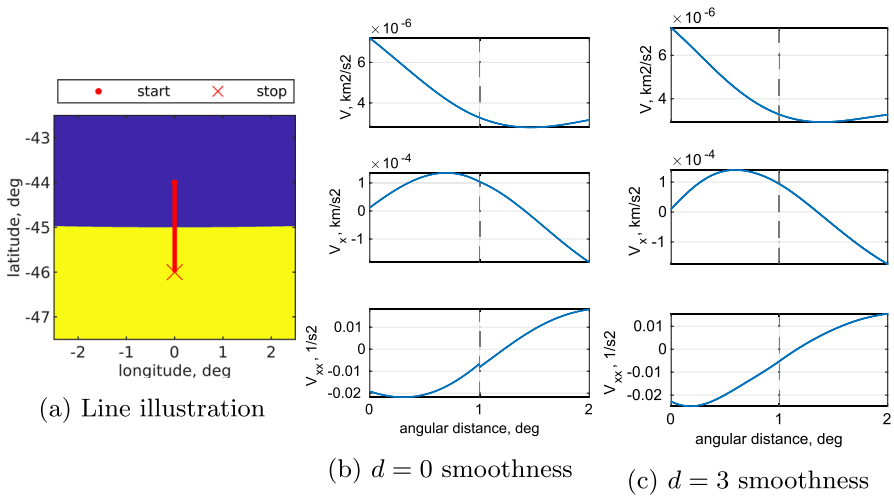


Fig. 8 Smoothness across horizontal face boundary

potential and its derivatives evaluated along the surface of the reference sphere. Figure 7 shows the transition across a vertical face boundary, Fig. 8 shows the transition across a horizontal face boundary, Fig. 9 shows the transition across a corner boundary between three faces, and Fig. 10 shows the transition across a patch boundary at the South Pole.

In the line illustrations, the line of evaluation locations is red and the faces are colored according to their face indices consistent with Fig. 3. The zoomed line illustration in Fig. 10b includes black lines of constant latitude (-85 deg, -80 deg, and -75 deg) and longitude (45 deg, 135 deg, 225 deg, and 315 deg) to show that

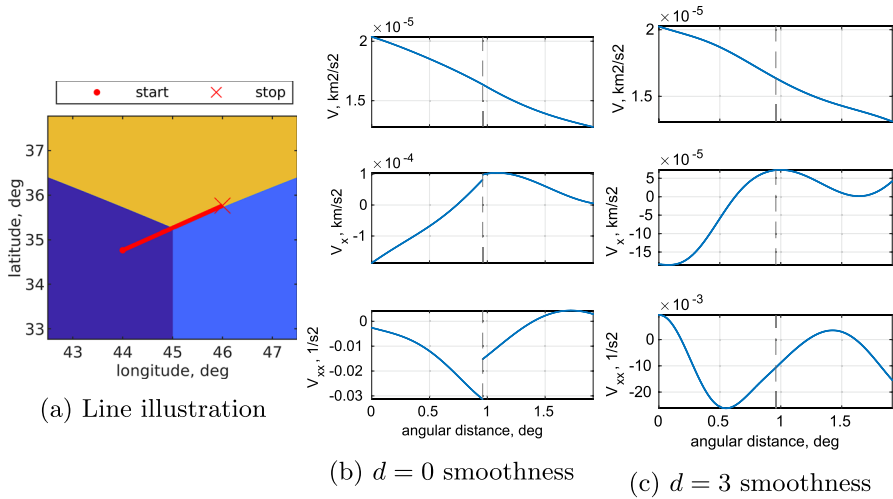


Fig. 9 Smoothness across corner

the smoothness test line passes directly through the South Pole. On the plots of the gravitational potential and its gradients, the horizontal axes represent the angular distance from the first point on each line and vertical dashed lines are shown at patch boundaries. The visualized model is GRGM1200A_WCS_70. The other two models demonstrate similar smoothness behavior.

The results from Figs. 7, 8, 9, 10 show that continuity is achieved up to order d . Above order d , continuity is not guaranteed by the Junkins weighting functions. Fig. 7 is a particularly striking example of a case where the $d = 0$ Junkins weighting functions do not enforce continuity in the derivatives, with a noticeable kink in the curve for the gravitational potential. The $d = 3$ Junkins weighting functions yield smooth curves, because the curves shown are only up to the second-order gradients.

3.3 Gravity Potential and Acceleration

Figures 11, 12, and 13 show the RMS acceleration errors for the degree $L_e = 70$, $L_e = 300$, and $L_e = 550$ WCS models over a range of altitudes, as compared to the RMS acceleration errors of their equivalent SH models. The RMS acceleration errors for Figs. 11, 12, 13 are computed using evaluation points spanning the full sphere in spiral distributions. The number of RMS evaluation points in the spiral distributions are 78,400, 1,440,000, and 4,840,000 for the $L_e = 70$, $L_e = 300$, and $L_e = 550$ models, respectively. For all three demonstration models, the RMS acceleration error curves for the WCS models are close to the corresponding curves for their equivalent SH models. Achieving such closely tracking curves across all altitude ranges in Figs. 11, 12, 13 required significant tuning efforts, resulting in the parameters provided in Table 4.

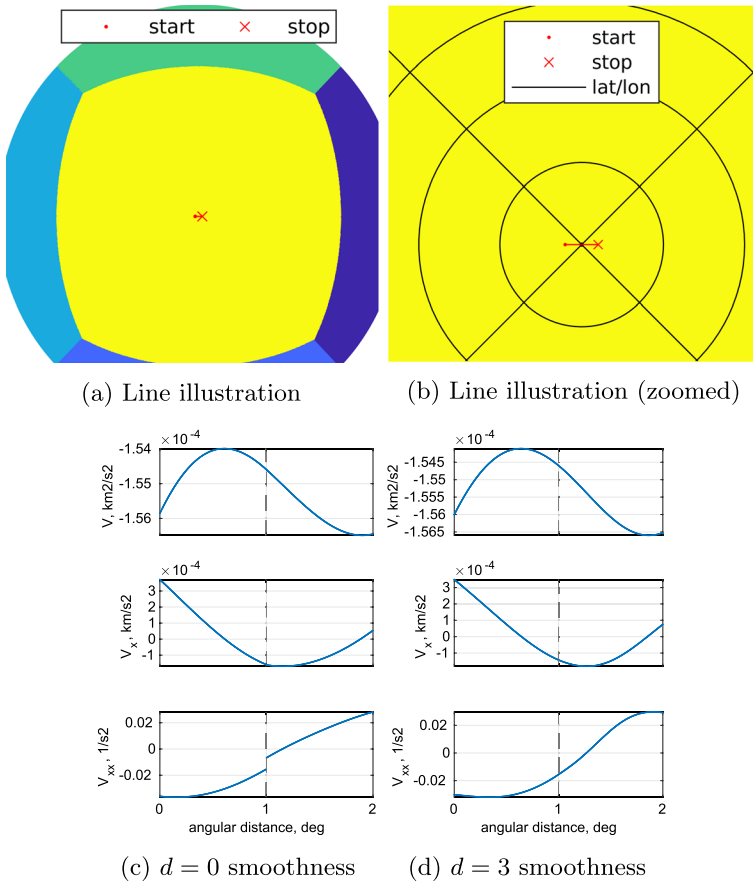


Fig. 10 Smoothness across South Pole

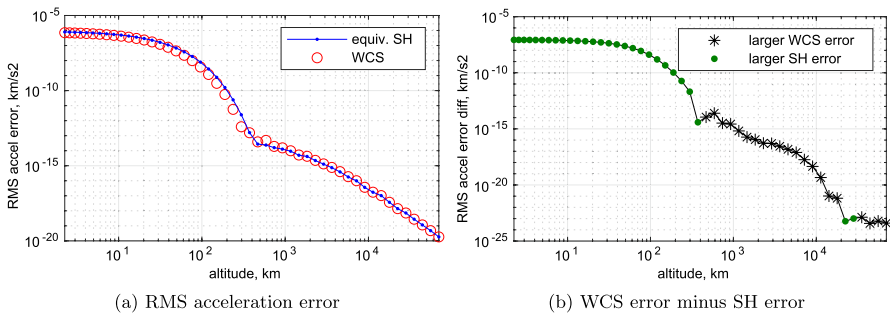


Fig. 11 Weighted cubed-sphere altitude vs. error study, $L_e = 70$

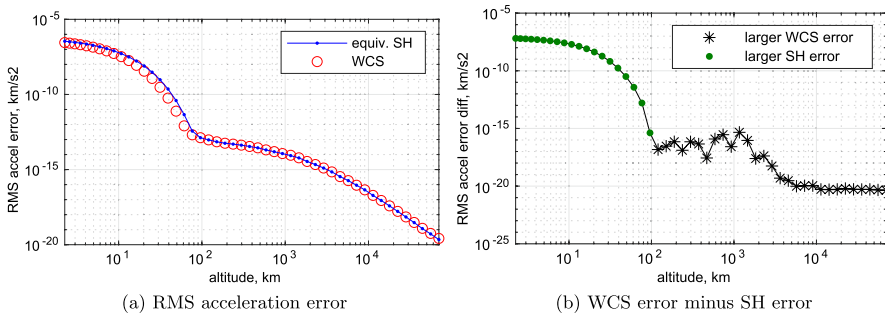


Fig. 12 Weighted cubed-sphere altitude vs. error study, $L_e = 300$

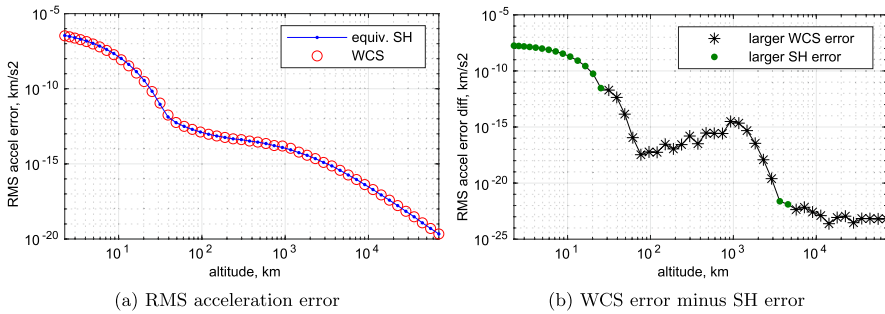


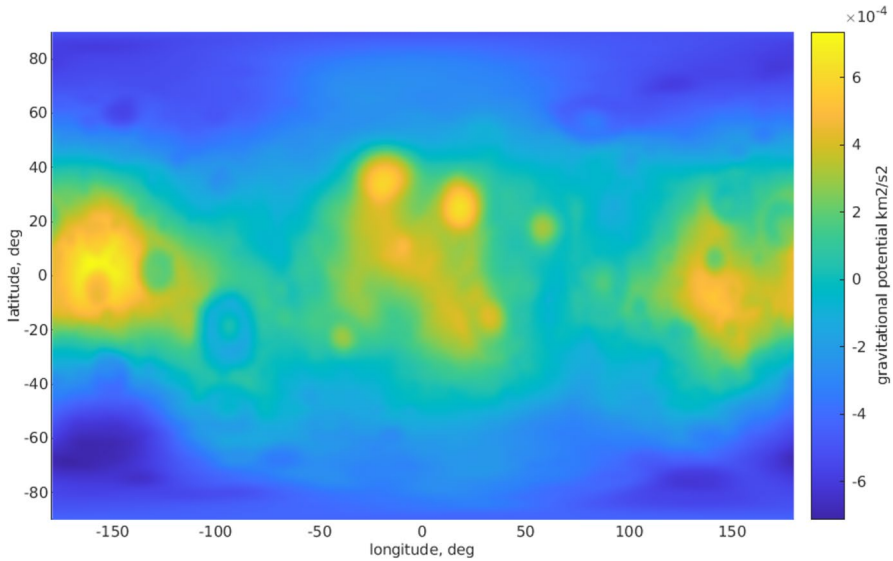
Fig. 13 Weighted cubed-sphere altitude vs. error study, $L_e = 550$

Figures 14 and 15 shows the gravitational potential and acceleration magnitude for GRGM1200A_WCS_300 over all longitudes/latitudes at 50 km altitude. The figures also show the residuals between the WCS model and its $L_e = 300$ equivalent SH model. The WCS model reproduces the potential and acceleration values smoothly.

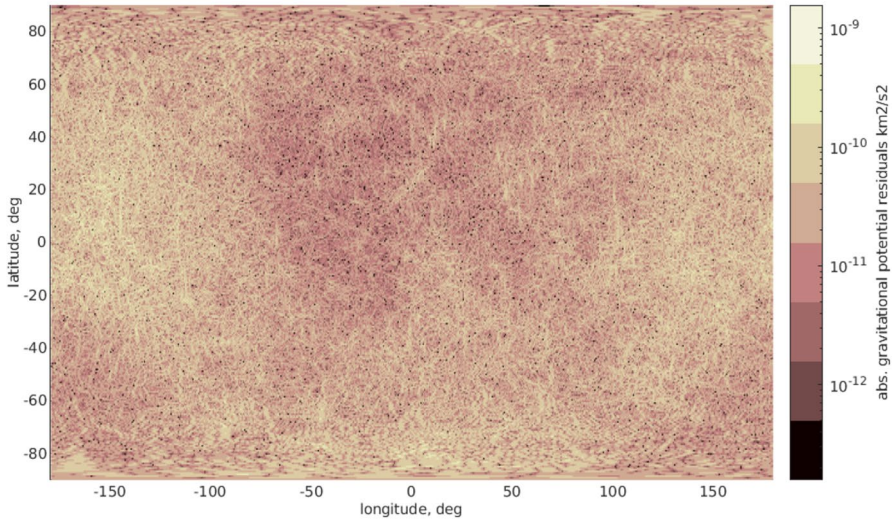
3.4 Trajectory Error Study

A trajectory propagation error study is performed by integrating several example trajectories with the WCS model, with the equivalent-fidelity SH model truncation, and with the truth SH model truncated to the specified L_t value from Table 4. The Stokes coefficients for the truth SH model are perturbed using their published uncertainties to simulate the gravity modeling error in the Stokes coefficients. The differences between the final positions of the trajectory computed using the WCS model and using the truth SH model are considered the WCS position error. Corresponding differences between trajectories computed using the equivalent-fidelity SH model and using the truth SH model are considered the equivalent-fidelity SH position error.

Six classes of example trajectories are used, including four sets of polar orbits with initial altitudes ranging from 25 to 300 km, a Near Rectilinear Halo Orbit



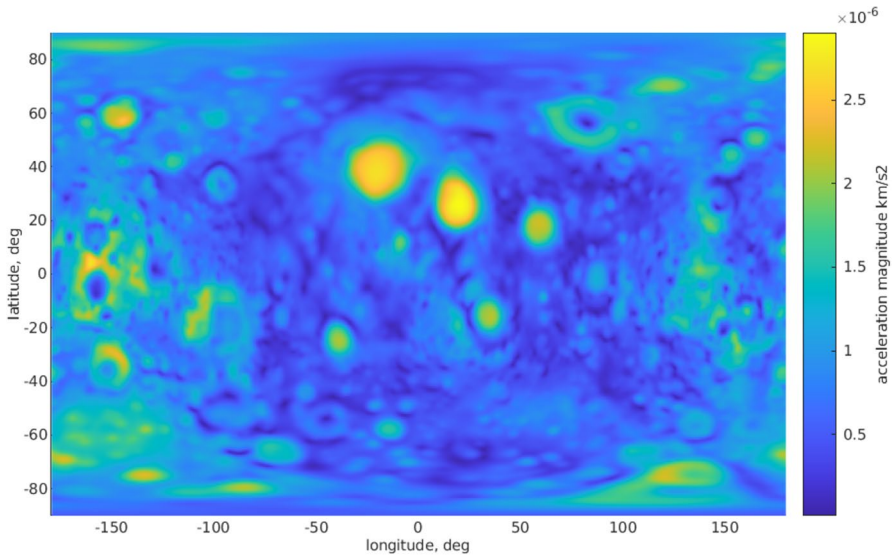
(a) Gravitational potential



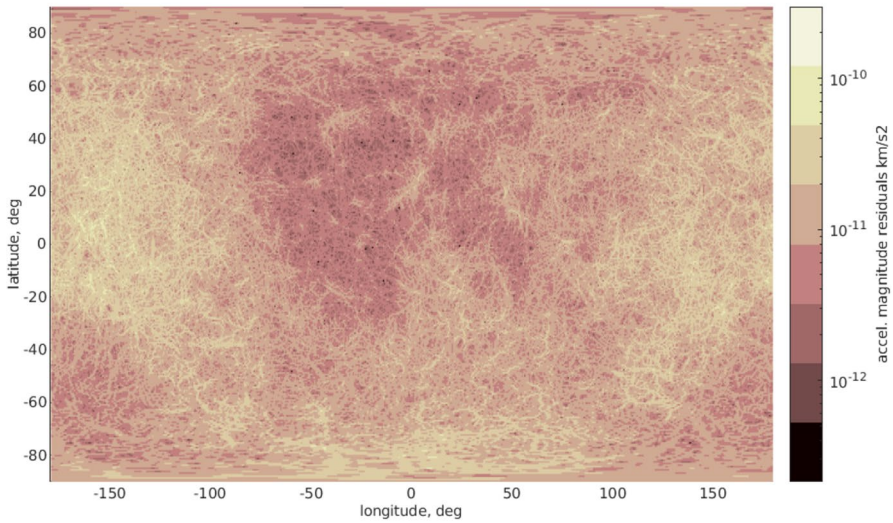
(b) Gravitational potential difference with equivalent SH

Fig. 14 GRGM1200A_WCS_300 potential, 50 km altitude

(NRHO), and a Distant Retrograde Orbit (DRO). The sets of polar orbits include orbits with varying initial longitude of the ascending node Ω_0 values to ensure that the polar orbits cover all latitudes and longitudes. Table 6 summarizes the six sets of orbits used in the trajectory error study. See Chapter 4 in Ref. [14] for more details



(a) Gravitational acceleration



(b) Gravitational acceleration difference with equivalent SH

Fig. 15 GRGM1200A_WCS_300 acceleration, 50 km altitude

on the trajectory error study setup. Some minor typos in the time-of-flight (TOF) values from Table 6 have been corrected since the corresponding table appeared in Ref. [14].

Figures 16 and 17 show the results of the trajectory error study performed on the WCS mascon models. The horizontal lines in Fig. 16 span the range of values

Table 6 Summary of trajectories used for error study

Orbit Type	h_{\min} , km	Revs	TOF, days	# Ω_0	Trials per. IC	Total Runs
25 km polar	23.7	≈ 6	0.458	60	100	6000
50 km polar	48.0	≈ 6	0.464	60	100	6000
100 km polar	99.2	≈ 6	0.477	60	100	6000
300 km polar	298.7	≈ 6	0.529	60	100	6000
NRHO	1506.7	1	6.581	N/A	6000	6000
DRO	68491.8	1	13.982	N/A	6000	6000

for the final position error, the box spans the 25th to 75th percentiles, and the vertical lines indicate the median values, which are also reported in the vertical axis labels. The red boxes in Fig. 16 represent the WCS models and the blue boxes represent the equivalent SH models.

The same box plot format is used for Fig. 17, except this figure shows the differences between the WCS models and their equivalent SH models instead of the errors against the truth model.

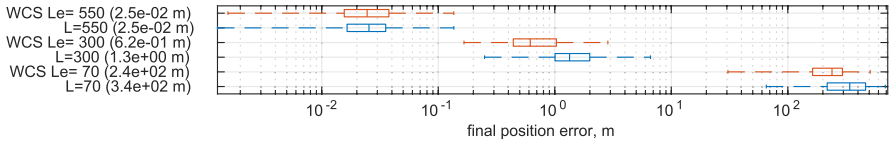
The errors for the WCS models are comparable or lower than their equivalent-fidelity SH models in all orbital regimes, in agreement with the acceleration error vs. altitude curves shown in Figs. 11, 12, 13. The highest-fidelity $L_e = 550$ WCS model is extremely accurate, with centimeter-level final position errors in the lowest 25 km altitude polar orbits.

3.5 Evaluation Runtime Benchmarking

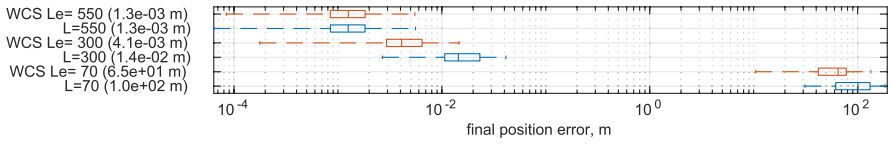
Tables 7 and 8 show the sequential evaluation runtime speedup for the WCS models compared to their equivalent SH models. The runtime analysis is performed using Fortran driver routines on the same Linux workstation and Stampede2 Skylake compute node as the runtime analysis presented in Ref. [14].

The evaluations for the runtime analysis are performed at random locations on the sphere to avoid a possible unfair advantage for the WCS models that would arise from repeatedly accessing the same local mascon model in the CPU cache. Note that for many applications, such as trajectory propagation, the same local model would in fact be accessed repeatedly, potentially granting better runtime improvements than presented here.

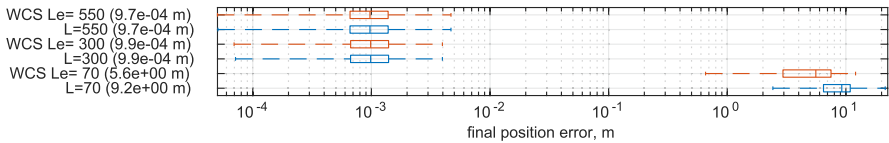
Tables 9 and 10 show the parallel evaluation runtime speedups for the WCS models compared to their equivalent SH models when using OpenMP parallelization. The speedup from the best choice of `OMP_NUM_THREADS` is shown and the selected value for the number of threads is reported in parentheses. As indicated by the entries where zero threads are found to be optimal, the 70 degree equivalent WCS model does not benefit from OMP parallelization on Stampede2. The larger models only benefit from a modest number of processors, since the local models are not large enough to justify the overhead from managing a large amount of threads.



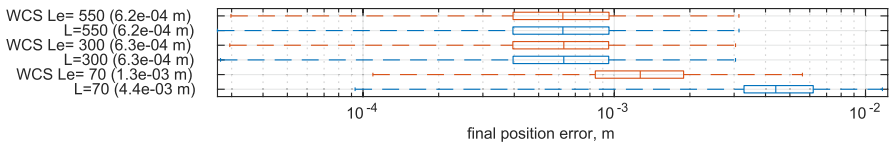
(a) 25 km altitude polar



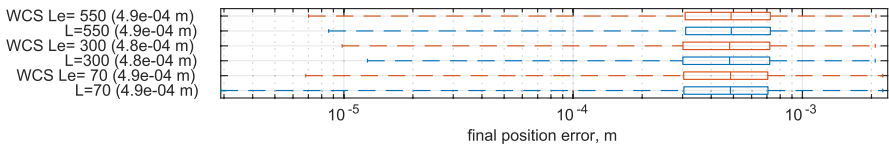
(b) 50 km altitude polar



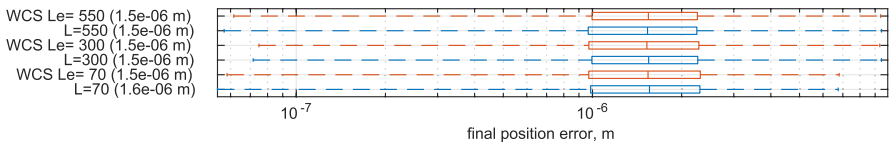
(c) 100 km altitude polar



(d) 300 km altitude polar



(e) NRHO



(f) DRO

Fig. 16 WCS trajectory error study final position errors compared to the truth (see L_i in Table 4 for size of SH truth field)

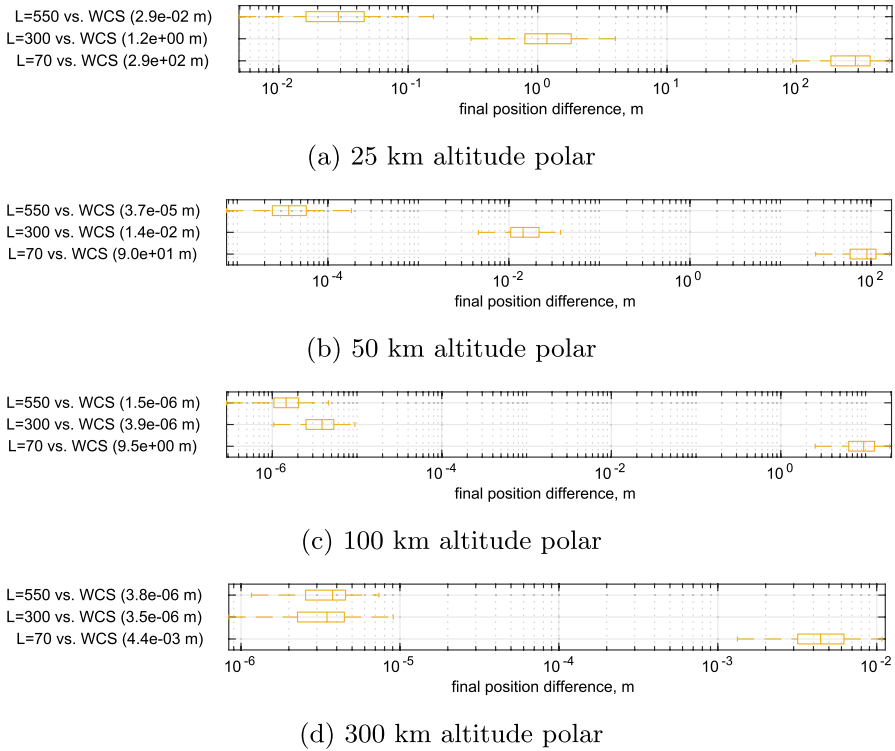


Fig. 17 WCS trajectory error study final position differences

Table 7 WCS sequential runtime analysis on Linux workstation

Equiv. SH Deg	WCS vs. SH Speedup			
	V	∇V	∇∇V	∇∇∇V
70	2.11	3.07	4.26	5.19
300	6.35	10.48	22.44	27.40
550	12.00	28.02	50.46	105.55

Table 8 WCS sequential runtime analysis on Stampede2

Equiv. SH Deg	WCS vs. SH Speedup			
	V	∇V	∇∇V	∇∇∇V
70	4.05	4.10	4.87	7.71
300	15.82	21.64	28.70	40.77
550	24.73	52.51	56.28	91.18

Table 9 WCS OpenMP parallel runtime analysis on Linux workstation

Equiv. SH Deg	WCS vs. SH speedup (# Threads)			
	V	∇V	∇∇V	∇∇∇V
70	3.57 (4)	5.06 (4)	6.91 (4)	10.21 (4)
300	18.00 (4)	29.74 (4)	63.58 (4)	84.73 (4)
550	38.80 (4)	90.09 (4)	162.33 (4)	356.88 (4)

Table 10 WCS OpenMP parallel runtime analysis on Stampede2

Equiv. SH Deg	WCS vs. SH speedup (# Threads)			
	V	∇V	∇∇V	∇∇∇V
70	4.05 (0)	4.10 (0)	4.87 (0)	7.71 (0)
300	17.99 (3)	36.83 (3)	43.25 (24)	64.72 (4)
550	57.63 (3)	86.42 (20)	134.02 (7)	238.12 (11)

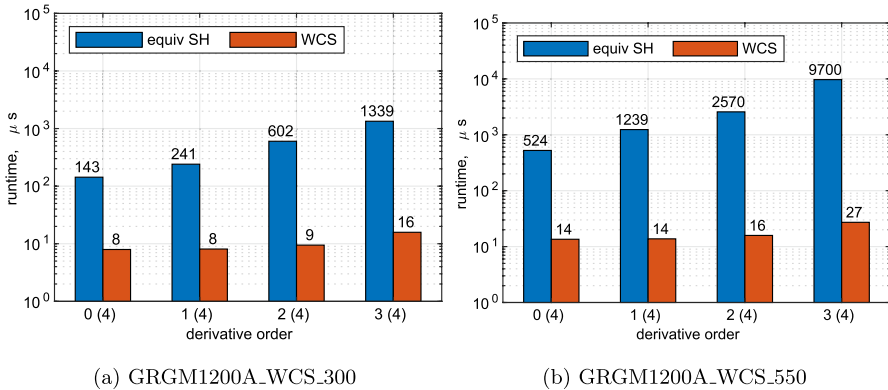


Fig. 18 WCS OpenMP parallel runtimes, Linux workstation

Figures 18, 19, 20, 21 report the evaluation runtimes and parallel runtime analyses for the degree 300 and degree 550 equivalent WCS models. The WCS models demonstrate favorable runtime performance compared to their equivalent SH models when evaluated using both sequential and CPU-parallelized code (using a modest number of processors). With parallelization, the WCS mascon models achieve a two order of magnitude speedup for the second- and third-order gradients.

Notably, the speedup performance is comparable on the Linux personal workstation and Stampede2 for the WCS models. The WCS models reduce the large global model down into relatively small local models, so the high-performance AVX512 registers on Stampede2 have a smaller influence on the runtime performance. The effect of OpenMP parallelization is also somewhat diminished by the small local model sizes, which is especially apparent in the relatively low parallel speedup values reported in Fig. 21.

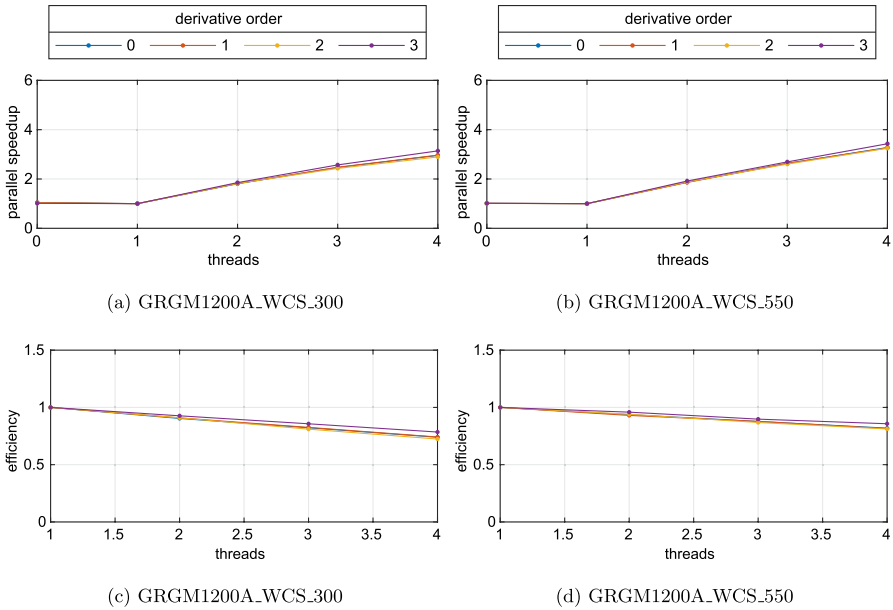


Fig. 19 WCS OpenMP parallelization analysis, Linux workstation

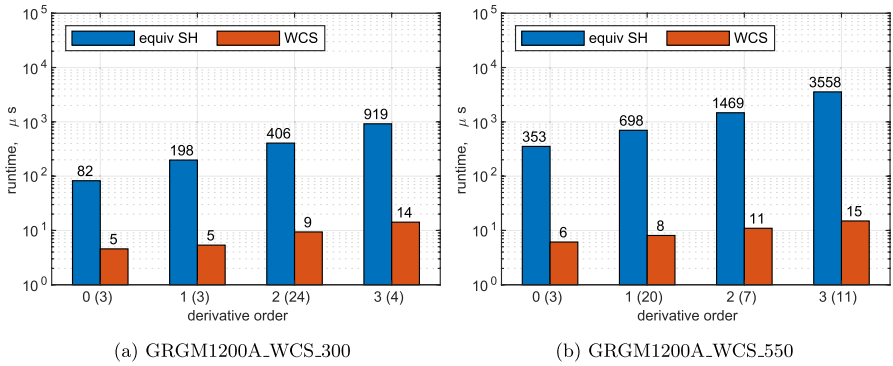


Fig. 20 WCS OpenMP parallel runtimes, Stampede2

3.6 Comparison with Previous Mascon Models

Figure 22 illustrates a runtime and memory benchmarking analysis carried out for highlighted fixed-mass models from Ref. [15] (square: GRGM1200A_PMC_EQM), free-mass models from Ref. [16] (circle: GRGM1200A_PMC), and WCS models from this work (diamond: GRGM1200A_WCS). All three

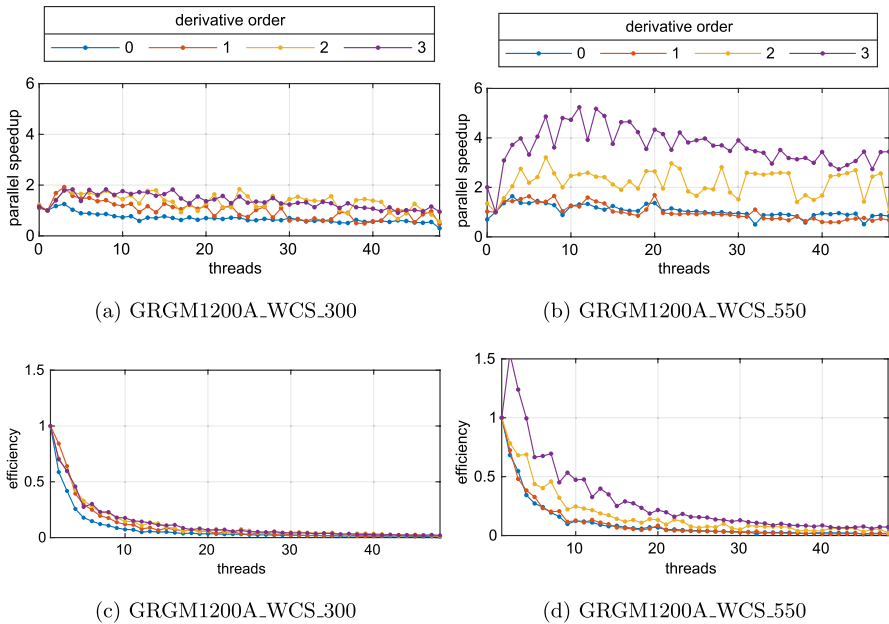
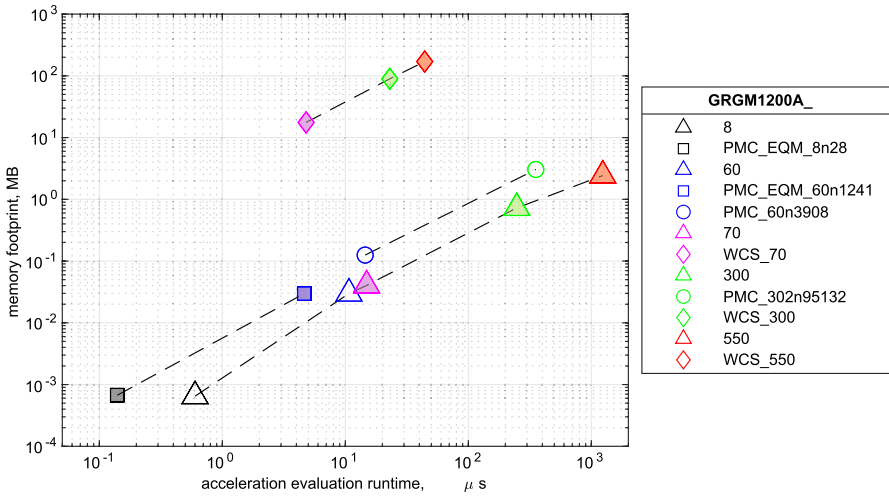


Fig. 21 WCS OpenMP parallelization analysis, Stampede2

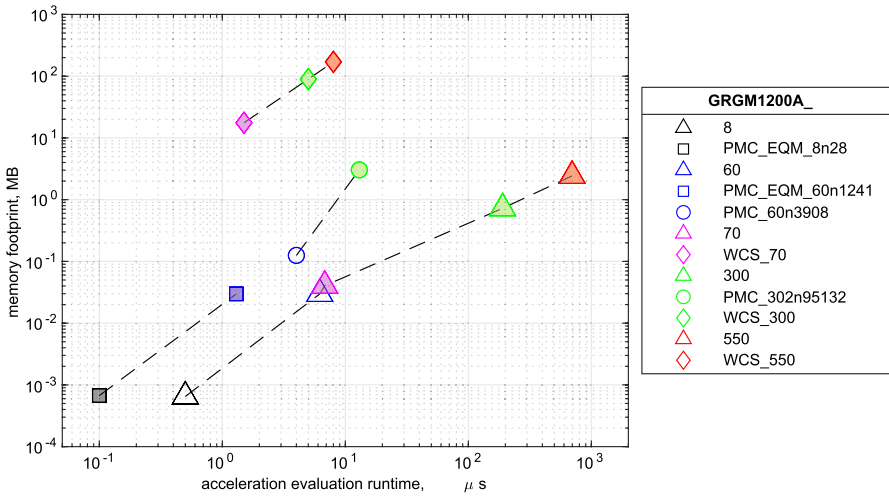
types of mascon models are compared to their equivalent SH models (triangle: GRGM1200A). Table 11 shows the recommended model types used for different levels of fidelity. *Memory* indicates the ratio of the memory occupied by the suggested model divided by the memory occupied by its equivalent-fidelity SH model. *Speedup* indicates the ratio of the acceleration evaluation runtime speed for the equivalent-fidelity SH model divided by the runtime speed for the suggested model.

4 Conclusion

This paper introduces the WCS mascon modeling technique, enabling a more widespread use of higher-fidelity gravity models for the Moon. The WCS demonstration models presented in this work yield similar runtime speedups to previous interpolated gravity representations with lower memory footprints (on the order of hundreds of megabytes, rather than gigabytes, for high-resolutions). The presented WCS models are useful for astrodynamics practitioners who require high-fidelity gravity models with low evaluation runtime costs. Using CPU multithreading parallelization yields additional runtime advantages, typically not available to SH models. The $L_e = 550$ WCS model achieves a nearly two order of magnitude faster acceleration evaluation runtime than SH. As the final of three papers in a series on mascon models at the moon, a comprehensive comparison is also performed, leading to various recommendations for models depending on the desired resolution



(a) Linux workstation without parallelization



(b) Stampede2 with OpenMP parallelization

Fig. 22 Memory footprint and evaluation runtime, filled markers are Pareto-optimal, dashed lines connect models of the same type, colors discriminate resolution

of gravity field and computer architecture. As a rule of thumb, the global fixed-mass models are ideal for low fidelity in non-parallel computing environments, the global free-mass models are ideal for medium fidelity when parallelization is available, and the WCS mascon models are ideal for the highest fidelity as well as medium fidelity when parallelization is not available.

One potential disadvantage of mascon gravity formulations over the standard SH formulation is that it is more straightforward to adjust the fidelity of a SH model than a mascon model. Since the terms in a SH model are orthogonal to each other,

Table 11 Recommended point mascon models from database

Equivalent Degree/Order	Linux Workstation			Stampede2		
	Sequential			OpenMP Parallelization		
	Type	Memory	Speedup	Type	Memory	Speedup
8	Fixed	1.0	4.3	Fixed	1.0	6.2
60	Fixed	1.0	2.3	Fixed	1.0	4.8
300	WCS	123.6	10.5	Free	4.2	14.6
550	WCS	70.2	28.0	WCS	70.2	86.4

the fidelity of a SH model can be reduced by simply truncating Stokes coefficients. Adding and removing arbitrary mascons from an existing mascon model does not result in an accurate representation of the gravity field. As the required gravity model fidelity varies with changing altitudes, the user of a mascon model would need to switch between models to use the most runtime-efficient model for the current accuracy needs.

There are a number of opportunities for future work. The generation process for the local mascon models could be improved, with a more systematic selection of mascon positions and masses. The demonstration models presented here are not necessarily the most efficient models possible in terms of the quantity of mascons used to represent gravity with a given level of fidelity.

The WCS models generated in this work are based on the existing GRGM1200A lunar gravity model, however it may be preferable to generate WCS models using direct measurements of the gravity field, rather than using a SH reference model. Generating WCS mascon models using direct measurements of the gravity field would resolve two disadvantages of the current approach: first, an additional source of error from the necessarily imperfect fit to the SH model would be avoided. Second, the uncertainty of the WCS model parameters could be computed during model generation, given the known errors in the gravity field measurements. This quantification of WCS model parameter uncertainty could be then mapped to the resulting orbital state uncertainty when using the WCS model to propagate trajectories.

The WCS representation could be applied to other gravity models besides the Moon, with the Earth, Mars, and Venus being the prime candidates. The WCS interpolation method could be immediately applied with basis functions other than point mascons to smoothly combine locally-valid models into a global function for gravity or other functions of interest. The models and runtime drivers presented in this paper are available online.

Acknowledgements This material is based upon work supported by NASA under award No. 80NSSC18M0121. The authors would like to thank Shane Robinson and David Dannemiller for their technical guidance discussions and insightful suggestions. The authors acknowledge the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing high performance computing resources that have contributed to the research results reported within this paper. The authors would also like to thank three anonymous reviewers for their insightful feedback.

Funding This material is based upon work supported by NASA under award No. 80NSSC18M0121.

Code Availability The weighted cubed sphere mascon model database and accompanying driver software are available at <https://doi.org/10.5281/zenodo.5725120>.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no Conflict of interest.

References

1. Arora, N., Russell, R.P.: Efficient interpolation of high-fidelity geopotentials. *J. Guid. Control. Dyn.* **39**(1), 128–143 (2016). <https://doi.org/10.2514/1.G001291>
2. Fantino, E., Casotto, S.: Methods of harmonic synthesis for global geopotential models and their first-, second- and third-order gradients. *J. Geodesy* **83**(7), 595–619 (2009). <https://doi.org/10.1007/s00190-008-0275-0>
3. Frisvad, J.R.: Building an orthonormal basis from a 3D unit vector without normalization. *J. Graph. Tools* **16**(3), 151–159 (2012). <https://doi.org/10.1080/2165347X.2012.689606>
4. Goossens, S., Lemoine, F., Sabaka, T. et al.: A global degree and order 1200 model of the lunar gravity field using GRAIL mission data. In: 47th Lunar and Planetary Science Conference, Abstract #1484 (2016)
5. Greengard, L., Rokhlin, V.: A fast algorithm for particle simulations. *J. Comput. Phys.* **73**(2), 325–348 (1987). [https://doi.org/10.1016/0021-9991\(87\)90140-9](https://doi.org/10.1016/0021-9991(87)90140-9)
6. Jancaitis, J.R., Junkins, J.L.: Modeling in N dimensions using a weighting function approach. *J. Geophys. Res.* **79**(23), 3361–3366 (1974). <https://doi.org/10.1029/JB079i023p03361>
7. Jones, B.A., Born, G.H., Beylkin, G.: Comparisons of the cubed-sphere gravity model with the spherical harmonics. *J. Guid. Control. Dyn.* **33**(2), 415–425 (2010). <https://doi.org/10.2514/1.45336>
8. Junkins, J.L.: Investigation of finite-element representations of the geopotential. *AIAA J.* **14**(6), 803–808 (1976). <https://doi.org/10.2514/3.61420>
9. Junkins, J.L., Miller, G.W., Jancaitis, J.R.: A weighting function approach to modeling of irregular surfaces. *J. Geophys. Res.* **78**(11), 1794–1803 (1973). <https://doi.org/10.1029/JB078i011p01794>
10. Junkins, J.L., Younes, A.B., Woollands, R.M., et al.: Efficient and adaptive orthogonal finite element representation of the geopotential. *J. Astronaut. Sci.* **64**(2), 118–155 (2017). <https://doi.org/10.1007/s40295-016-0111-3>
11. Kaula, W.M.: *Theory of Satellite Geodesy*. Dover Publications (2000)
12. Lemoine, F.G., Goossens, S., Sabaka, T.J., et al.: GRGM900C: a degree 900 lunar gravity model from GRAIL primary and extended mission data. *Geophys. Res. Lett.* **41**(10), 3382–3389 (2014). <https://doi.org/10.1002/2014GL060027>
13. Mahajan, B., Alfriend, K.T.: Analytic orbit theory with any arbitrary spherical harmonic as the dominant perturbation. *Celest. Mech. Dyn. Astron.* **131**(10), 45 (2019). <https://doi.org/10.1007/s10569-019-9923-3>
14. McArdle, S.: Gravity modeling for lunar orbits. PhD thesis, The University of Texas at Austin, (2022). <https://doi.org/10.26153/tsw/43345>
15. McArdle, S., Russell, R.P.: Point mascon global lunar gravity models. *J. Guid. Control. Dyn.* **45**(5), 815–829 (2022). <https://doi.org/10.2514/1.G006361>
16. McArdle, S., Russell, R.P.: High-resolution global point mascon models for the moon. *J. Guid. Control. Dyn.* **46**(7), 1390–1396 (2023). <https://doi.org/10.2514/1.G006921>
17. McArdle, S., Russell, R.P., Bettadpur, S.: A practical method for truncating spherical harmonic gravity fields, application at the Moon. In: 43rd American Astronautical Society Annual Guidance, Navigation and Control Conference, Paper AAS 20-048 (2020)
18. McArdle, S.M., Russell, R.P.: Global lunar gravity field using local point mascon models. In: AAS/AIAA Astrodynamics Specialist Conference, Paper AAS 22-827 (2022)
19. Parker, J.S., Chikine, S., Kayser, E., et al.: Sustained low-altitude lunar orbital mission (SLALOM) navigation concept. In: 31st AAS/AIAA Spaceflight Mechanics Conference, Paper AAS 21-337 (2021)

20. Ronchi, C., Iacono, R., Paolucci, P.S.: The “cubed sphere”: a new method for the solution of partial differential equations in spherical geometry. *J. Comput. Phys.* **124**(1), 93–114 (1996). <https://doi.org/10.1006/jcph.1996.0047>
21. Saff, E.B., Kuijlaars, A.B.: Distributing many points on a sphere. *Math. Intell.* **19**(1), 5–11 (1997). <https://doi.org/10.1007/BF03024331>
22. Woodburn, J., Szebehely, V., Zare, K.: An alternative representation of the geopotential. In: AAS/AIAA Spaceflight Mechanics Meeting, Paper AAS 95-191 (1995)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.