AAS 11 - 501

FAST, EFFICIENT AND ADAPTIVE INTERPOLATION OF THE GEOPOTENTIAL

Nitin Arora[*] and Ryan P. Russell[†]

Conventional high-fidelity geopotential computations rely on expensive spherical harmonics (SH) series. In this study an interpolation scheme is proposed that classically improves compute speed at the expense of memory. The approach is exact in the sense that accelerations are calculated naturally as the gradient of the fitted potential, and continuity and smoothness to arbitrary order are ensured across local cells using the Junkins weight functions. Millions of local interpolating functions are chosen with a new adaptive method that minimizes coefficient storage subject to a maximum error threshold. Analytic inversions of the normal equations associated with each candidate interpolant allow for rapid solutions to the least squares process without resorting to the conventional numerical linear system solvers. Accordingly, time is afforded to cycle through hundreds of candidate interpolants for each of the millions of nodes, resulting in a global model with a highly optimized memory requirement and uniform error distribution. Speed is ensured by choosing simple polynomials as candidate interpolants. For example, the interpolation approach (deemed FETCH) fitting the full GRACE02C 200×200 spherical harmonics (SH) field requires 1.8 Gigabytes of memory and achieves over 300x speedups compared to a Pines SH implementation. The error profile of the interpolation model is adaptively selected throughout the global domain to conservatively mirror the published expected errors of the SH fitting function.

INTRODUCTION

High-fidelity trajectory computation using conventional spherical harmonics gravity field model is computationally slow and non-intuitive to implement efficiently.[1] Various representations (singular and non singular) of the spherical harmonics model can be found in the literature[1–3] all of them being computationally slow for high fidelities. With the current computational resources it is simply not feasible to account for high fidelity geopotential models in near real-time trajectory applications such as monitoring the space catalog.[4] Furthermore, the rapidly increasing number of objects in orbit is driving the need to develop faster techniques for computing high fidelity geopotentials.

Various alternatives to SH for computing gravity potentials have been proposed over the years and can be broadly divided into two classes 1) Mascon models and 2) 3D interpolation models. Mascon models (deemed as any models using point masses, or surface or volume mass distributions) were attractive in the early days of satellite geodesy due to the limited amount of global data.[5–9] Both terrestrial gravity anomalies and spacecraft tracking arcs were localized and combinations of the global SH with the localized mascon models provided higher resolution in regions with better data. Originally therefore the mascon models provided the main benefit of adaptive local resolution. More recently, with the growing interest in robotic and potential human exploration of small irregular shaped bodies, the volume-based and surface-based mascon models are of great interest because they are valid in the entire domain above the surface while SH representations fail to converge inside the reference radius.[10, 11] In particular the polyhedral method is a robust and elegant solution for irregular small bodies although the extra computational requirements are cumbersome. Mascon models have historically not been designed or used based on speed considerations. However, a modern high fidelity geopotential mascon model was recently demonstrated to provide order of magnitude speed improvements

[*]PhD Candidate, Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, 270 Ferst Drive, Atlanta, GA, 30332-845-541-7861,N.arora9@gatech.edu
[†]Assistant Professor, Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, 270 Ferst Drive, Atlanta, GA, 30332-0150, 404-385-3342 (voice), 404-894-2760 (fax),ryan.russell@gatech.edu

over SH when implemented on common Graphics Processing Unit (GPU).[12]

The second and more popular class of alternative potential formulations is the 3D interpolation class. Methods in this class are applicable for both irregular and near-spherical shaped bodies, and expedite computations by effectively trading computer memory for run-time speed. Essentially first proposed by Junkins in 1976,[13] the interpolation methods have been bolstered recently by the extraordinary memory resources of common computers. Depending on the interpolation method, a variety of techniques and basis functions are employed including weighting functions,[13, 14] wavelets,[15] splines,[15, 16] octrees[17] and psuedocenters.[18] Each interpolation method balances accuracy with efforts to maximize runtime and minimize memory footprint while achieving exactness, continuity and smoothness as appropriate.

The new 3D interpolation model proposed in this study is motivated primarily by the works of Junkins,[13, 14] Hujsak,[18] Colombi et. al,[17] and Beylkin and Cramer[15] among others. Over thirty years ago, Junkins demonstrated that a one order of magnitude speedup was possible at the expense of a modest investment in memory. This was a remarkable feat considering the quality of computers at the time. His model encompassed the region around Earth out to 1.2 radii and achieved roughly 6 digits of accuracy at the expense of storing 30,000 coefficients. Hujsak revisited the problem some 20 years later introducing the concept of interpolating the pseudocenter coordinates instead of the potential or acceleration. With a simpler interpolating scheme and improved hardware he fit a 70×70 field (for the northern hemisphere at altitudes between 400 and 1500 km) with only 5 megabytes of storage. The algorithm requires the calculation overhead equivalent to a 5×5 field, leading to approximately a 100 fold speedup compared to SH for the 70×70 resolution. Columbi, Hirani, and Villac recently applied similar concepts using modern tools for calculating gravity fields for highly non-spherical bodies such as comets and asteroids. Their methods are demonstrated to provide approximately 100 fold speedups except their gains are compared to the expensive polyhedral methods not SH. Beylkin and Cramer show recent progress in the efficient storage of multi-resolution interpolating functions, and subsequently published the first global, modern 3D interpolation geopotential model called the Cubed-Sphere model.[19] Using multiple concentric shells and chebyshev basis functions, their method demonstrates 30 fold speedups compared to SH for their highest resolution 150×150 model. Their break even model in terms of time is said to be approximately at the resolution of 20×20. The memory footprint of their 150×150 model is 856 MB.

Presumably, the main reason for not widely adopting the 3D interpolation geopotential models is the memory requirements. As memory and processor technology has exponentially grown over the years, a renewed interest in memory-intense numerical methods is increasingly justified. Given the tremendous potential benefits (trading abundantly available memory for tremendous speed improvements), it is surprising that only a handful of authors have worked on this approach for the Earth application in the last thirty years, and only one other group[15, 19] in the last fourteen years. The time is ripe for innovation, and in this study a new model is proposed in an attempt to glean the best properties from the previous and competing models, while finding innovative solutions to correct their respective problems. The goal then of the current study is to build a modern 3D interpolation geopotential model that is fast and memory efficient with a priority placed on a solution method that is 1) continuous and smooth across the global domain to an arbitrary order of derivatives, 2) adaptive in terms of local vs. global resolution, and 3) has a residual error profile that is conservatively in the noise of the accuracy of the SH function, and 4) is singularity free. No previous or existing models can claim a complete handle on each of these desirable traits.

The new interpolation method proposed in this study deemed as 'Fetch' is unique in the sense that it is locally adaptive, and globally continuous, leading to an optimal trade off between memory and runtime performance. It relies primarily on a weighted interpolation scheme (developed by Junkins[13]) and uses locally adaptive polynomials. The specific application for modeling the geopotential utilizes a regular spherical coordinate grid. The classic singularity and associated numerical problems near the the poles (when converting to Cartesian) is avoided using a two level global grid structure with an additional weighting function to ensure continuity. Adaptive local target residual levels are used which are conservatively two times smaller than the published expected errors of the GGM02C gravity field data.[20] The error tolerances are mapped in a physics based manner to all altitudes such that the residuals tend to zero as altitude tends to infinity. The proposed model extend to the moon and the coefficients are generated on a 200 node CPU cluster. The interpolated geopotential is exact in the sense that accelerations and higher order derivatives (Jacobians and Hessians of the accelerations for example) are calculated as exact derivatives of the interpolant function. Only the potential function is interpolated thereby leading to significant memory savings especially when high orders derivatives are needed. Therefore the interpolated field remains conservative and preserves many attractive dynamical features therein. Furthermore, the higher order derivatives are also available at a modest cost and remain smooth and continuous for use in trajectory optimization and estimation applications. Note that the selection of weight function order influences how many derivative orders are continuous and smooth. In this study, 7th order weight functions are used to allow for smooth and continuous derivatives up to 6th and 7th order respectively.

The paper starts by giving details on the Fetch interpolation approach and the specifics when applied to the geopotential application, deemed as the GDF model. Next is a comprehensive performance profiling by direct comparison with a state-of-the-art, singularity-free SH implementation. Multiple orders of magnitude in speedups are reported for the high fidelity model and the break even SH model size in terms of compute speed is 10×10, (making the GF model approach approximately 4 times faster than the Cubed-Sphere approach). The next section gives an overview of the general computational model.

GENERAL COMPUTATIONAL MODEL

Localized representation of the gravity field follows naturally from the fact that there are uneven gravity undulations over the Earth surface. In order to separate the dominant global effects from the smaller local undulations we propose a geopotential model approximation as given by Eq. 1:

$$U \simeq U_{J_2} + U_{fetch} \tag{1}$$

where U represents the total potential from a specified degree and order of the GGM02C spherical harmonics field, U_{J_2} represents the potential only due to the J_2 term (J_2 for the reference field is 0.0010826356666), and U_{fetch} is the local potential obtained from the proposed interpolation method. The J_2 term in Eq. 1 is orders of magnitude more significant than all the higher order terms combined that follow later in the SH series. Removing of U_{J_2} (given by Eq. 2) from higher order terms provides an extra 3-4 digits of accuracy in the potential interpolation at an almost negligible computational cost.

$$U_{J_2} = \frac{\mu}{r}[J_2 * \frac{R_e}{r}^2 * P_2[sin(\phi)]] \tag{2}$$

here mu and R_e are the reference gravitational parameter and radius of the Earth respectively, r and ϕ are the magnitude and geocentric latitude of the position vector respectively, and P_2 is the second degree Legendre polynomial. Most all previous gravity field interpolation efforts exploited the benefit of removing the low frequency terms.[10,12,13,15,17–19]

The U_{fetch} term in Eq. 1 represents the final composite polynomial term computed using a weighted average of the eight node polynomials that are centered at each of the eight corners of the cell containing the current location (see Fig. 1 and Eq. 3). The coefficients for each of the eight local node polynomials are computed via least squares fits of the local geopotential (with J_2 removed). More details are provided later on the least squares solutions and candidate forms for the local node polynomials.

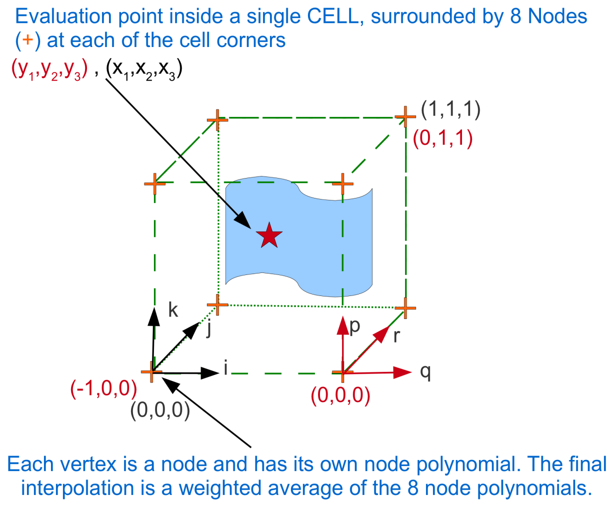
Evaluation point inside a single CELL, surrounded by 8 Nodes (+) at each of the cell corners
(y1,y2,y3) , (x1,x2,x3)
(1,1,1)
(0,1,1)
k
j
i
p
r
q
(-1,0,0)
(0,0,0)
(0,0,0)
Each vertex is a node and has its own node polynomial. The final interpolation is a weighted average of the 8 node polynomials.

Figure 1. Cell and node geometry for the weighed interpolation (see text for explanation)

$$U_{fetch} = \sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1} w_{ijk} U_{qrp}^{node} \tag{3}$$

In Fig. 1 and Eq. 3, the ijk coordinate system defines the cell and is normalized from 0 to 1 in each direction for use by the weight functions (w_{ijk}). The qrp coordinate system is the origin for the indicated local node polynomial and is valid from -1 to 1 in each direction (occupying 8 neighboring cells). Each cell has a single weight function origin, and 8 node origins (one at each of its corners). In Eq. 3, i,j and k (similarly q,r and p) correspond to latitude, longitude and radial directions respectively. The w_{ijk} are Hermite weight functions[13] normalized in (0, 1) dimensional space. The order of the Hermite functions can be chosen arbitrarily high to ensure any degree of user desired continuity or smoothness. The term U_{ijk}^{node} is the local node polynomial and is normalized in (-1, 1) dimensional space along each direction. The details of the weighted interpolation scheme are discussed later. The accelerations are obtained by taking the gradient of the resulting polynomial, and higher order derivatives are also available at a modest cost. It is worth re-emphasizing that the Junkins weighting function scheme allows for arbitrary functions to be used for the local node interpolants. In the current study, simple polynomials are chosen to minimize runtime. The degrees of each node polynomial is allowed to adaptively vary in each direction in order to maximize efficiency (defined as high accuracy and low memory) for the local conditions.

The process of generating and using the Fetch model is implemented in two phases. Phase 1 is performed once off-line and consists of parallel computation of the localized node polynomial coefficients on a regular and global 3D grid. In order to deal with the singularity at the poles when working in the spherical coordinates, a second rotated global grid is also required. Details on the singularity problem and the rotated grid solution are given in a later section. Phase 2 is the runtime interpolation and involves a weighted evaluation of the local node polynomials which are selected and stored for runtime use in Phase 1. The order of the weight functions (and accordingly the degree of continuity of the final composite function across the boundaries) is independent from the coefficient generation and can be selected at runtime. The next section gives a brief overview of the coefficient generation process.

PHASE 1: COEFFICIENT GENERATION

The Fetch interpolation model relies on a global discretization scheme for achieving adaptivity and continuity. The whole solution space is divided into 3D cells in the traditional spherical coordinate system consisting of longitude (lon), latitude (lat) and radial (r) directions. The geopotential within each cell is fit (in a least square sense) via node polynomials that are locally valid within each of the 8 cells touching that node. Hence, the problem of fitting the global geopotential is reduced to finding localized polynomial coefficients for each node.

Surface Discretization

Spherical coordinates are used for discretization of the whole solution domain. The weighted interpolation scheme (used to achieve continuity across the whole solution space) requires a uniform discretization for each dimension. In other words, the cutting planes that partition the global domain into local cells must all be mutually orthogonal, although the spacing between the planes can vary. Hence the spacing of the cutting planes is a degree of freedom and can be utilized to improve efficiency. The longitude-latitude space is ultimately chosen to have equal spacing for simplicity. Furthermore, the adaptive degree selection of the local polynomials affords the rationale that the smaller cells (in the Cartesian sense) near the poles will require polynomials of lower order to achieve the same accuracy as the larger cells near the equator. The cutting plane spacing in the radial direction is highly adaptive as the rapid undulations near the surface require shallow spacings while the high altitudes can be modeled with relatively distant spacings.

Tackling the Singularity A spherical coordinates based grid suffers from singularity at the poles when converting back and forth from the Cartesian coordinate space. Furthermore numerical degradation of the SH geopotential and its derivatives are known to be problematic at high latitudes nodes. To overcome these numerical issues we propose a two level global grid structure. The global grid is divided into two sub-grids, the primary grid and the rotated gird. The primary grid suffers with singularities at the poles but its domain of validity is constrained to lie within a certain latitude (+-75 deg for the current study). Starting from the poles a rotated grid is designed which has singularities at two points on the equator of the original grid (due to a 90 degrees rotation about the x-axis). The main idea is to evaluate the interpolant either on the primary or the rotated grid depending on the latitude of the point of interest. Hence, the rotated grid serves to remove the singularity at the poles in the spherical coordinate system. Precise continuity between the two grid sub-models is maintained via a Hermite weighting function applied in a narrow overlapping latitude band. Details on the continuity between the two grids are discussed later.

Figures 2 and 3 show the final surface discretization for the primary and the rotated grids. The active region for both the grids is shown in cyan and the inactive region is shown in white. The overlapping region is indicated by the solid shaded rings (and also bounded by the solid lines in Fig. 4). Evaluation inside the overlap region requires that both sub-models are computed and a 1D weighting function dependent only on latitude is used to ensure continuity across the sub-model boundaries.

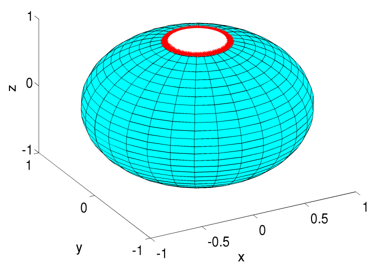

Figure 2. Primary grid

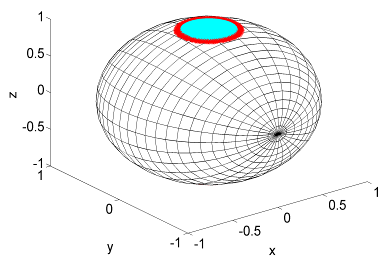

Figure 3. Rotated grid

Note that for the coefficient generation phase, each sub-model only requires coefficients for its respective domain including the overlapped areas. Therefore the two sub-model strategy creates little overhead during coefficient generation. To determine the valid domain of the rotated grid, the “z” distance of each node on the rotated grid and is compared with the "z" distance of each node of the primary grid for the overlapped region. Furthermore, the overlapping region is identified by padding the "z" distance of the primary grid with the latitude width of the overlapped region. Figure 4 shows the primary grid (dots) overlaid rotated grid (+) along with the overlapped region (bounded by the solid lines).

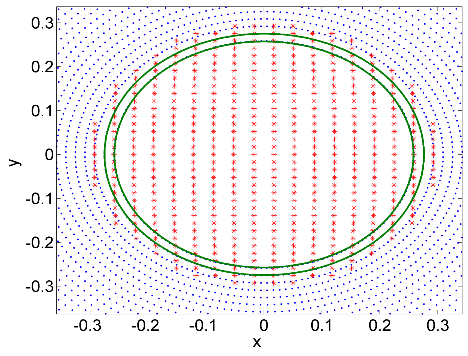

Figure 4. Overlapped grid

Radial Discretization

For the current study a total of 56 unevenly spaced rings are used to space the cutting planes in the radial direction extending all the way to the Moon. The rings are densely packed near the surface of the Earth where the field changes rapidly and they spread out rapidly as altitude increases. Having closely packed shells decreases the number of coefficients per cell but increases the number of cells required for the global model. On the other hand, choosing a very large radial step size increases the number of coefficients per cell making runtime function evaluation slower. Hence, there is a trade off between runtime memory and speed requirements. Figure 5 shows the radial shells placement and width selected for this study for each of the 56 shells. Starting from a log based distribution manual tuning is performed to obtain the final shells spacing.

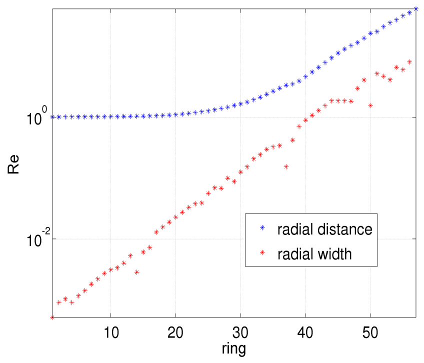

Figure 5. Radial discretization

Localized least square approximation

The local approximation for each node polynomial is obtained via a least squares fit. Before explaining the coefficient generation process, we define a variable called *polytype* ijk which represents the class of fitting polynomial being considered. For example, *polytype* 324 means the fitting polynomial is of degree 3, 2 and 4 in the lat, lon, and r directions respectively. Each cell in total has $i * j * k$ coefficients include all the cross terms of the node polynomial. As each node has its own localized polynomial there is hardly any inter-node communication required during coefficient generation. This strong decoupling leads to a simplified and efficient parallel implementation of the coefficient generation algorithm. The conventional least squares method for generating the coefficients is summarized by Eq. 4.

$$\vec{c} = \mathbf{B} * \vec{u} \tag{4}$$

Here, $\vec{c}$ is a column vector of coefficients estimates, $\vec{u}$ represents the measurement vector [the SH potential evaluated at each of the evenly spaced m^3 measurements within the node's region of validity (8 touching cells)] and **B** denotes the least squares inverse (LSI) matrix(Eq. 5):

$$\mathbf{B} = (\mathbf{H}^T * \mathbf{H})^{-1} \tag{5}$$

were the matrix **H** ($m^3 \times N$) represents the function evaluation matrix. Here, m represents the number of measurements evenly spaced in each direction and N represents the total number of coefficients per node polynomial. The **H** matrix contains the first order partials of the interpolant with respect to the estimated coefficients. As an example, consider a *polytype* 232 interpolant with $2 * 3 * 2 = 12$ coefficients given by Eq. 6:

$$\begin{split} U_{cell}[232] = \big(&c_1 + c_2 x_3 + c_3 x_2 + c_4 x_2 x_3 + c_5 {x_2}^2 + c_6 + {x_2}^2 x_3 + c_7 x_1 \\ &+ c_8 x_1 x_3 + c_9 x_1 x_2 + c_{10} x_1 x_2 x_3 + c_{11} x_1 {x_2}^2 + c_{12} x_1 {x_2}^2 x_3\big) \end{split} \tag{6}$$

the functional form for each row of H is:

$$H_{(i,:)} = \left(\frac{\partial U_{cell}[ijk]}{\partial \vec{c}}\right)_{evaluated\ \ at\ \ \rho_i}$$

where ρ_i is the location in node superscript, ijk subscript normalized space from [-1,1] of the i^{th} measurement. Computing the analytic partials and evaluating them at the specific measurement locations, the full H matrix is computed and stored in a symbolic manipulator program as a $m^3 \times N$ matrix of rational fraction entries. For the *polytype* 232 considering $m = 4$ or $4^3 = 64$ evenly spaced measurements, the $H^T * H$ matrix is easily computed and its inverse is analytically computed via Maple and has the form:

$$(H^T * H)^{-1} = \begin{bmatrix}
\frac{41}{1024} & 0 & 0 & 0 & -\frac{45}{1024} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{369}{5120} & 0 & 0 & 0 & -\frac{81}{1024} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{9}{320} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \frac{81}{1600} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\frac{45}{1024} & 0 & 0 & 0 & \frac{81}{1024} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -\frac{81}{1024} & 0 & 0 & 0 & \frac{729}{5120} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \frac{369}{5120} & 0 & 0 & 0 & -\frac{81}{1024} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{3321}{25600} & 0 & 0 & 0 & -\frac{729}{5120} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{81}{1600} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{729}{8000} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -\frac{81}{1024} & 0 & 0 & 0 & \frac{729}{5120} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -\frac{729}{5120} & 0 & 0 & 0 & \frac{6561}{25600}
\end{bmatrix}$$

Each node polynomial domain (the eight touching cells) is normalized between -1 to 1 along each of its dimensions. If the number of measurements (m^3) and their respective positions within each cell do not change, then the fully evaluated numerical **H** matrix is invariant across all nodes for a given candidate interpolant. This static property of the **H** matrix is only true if the number of measurements and their relative positions

(ρ_i for all measurements) are the same across all node domains. Given that the **H** matrix is static, the inversion matrix that solves the least squares problem in Eq. 5 can be computed analytically just once. In other words, once the LSI matrix is solved analytically, there is no need to numerically solve a linear system in order to obtain the least squares coefficient fit for a given candidate function. Therefore, in order to test a single candidate function across the global domain, each cell simply requires one matrix multiplication (Eq. 4) (also avoiding the expensive quadratures necessary in the Junkins approach[14]). It is further emphasized that the **H** matrix when evaluated with a symbolic manipulator is exact, where each entry is composed of rational fractions. Accordingly, such matrices can be analytically inverted using symbolic manipulators and the results are also exact and in the form of rational polynomials. Therefore, the typical numerical problems associated with solving large linear systems is completely avoided and the resulting inverted matrix is exact.[16]

For this study a Maple worksheet capable of creating analytically inverted LSI matrices corresponding to all possible combinations of polynomial degree ($polytype\ ijk$) within each node domain is implemented. The worksheet takes the maximum polynomial degree ($PNmax$) and m as inputs and writes out the LSI matrices using the automated Maple worksheet. Analytic inversion of a $6^3 \times 6^3$ matrix is computationally intensive and was not possible on a typical desktop computer until recently. However it is emphasized that these inversion matrices only need to be computed just once. During the coefficient generation process these matrices then are simply read once at the startup and reused later for all the node's domains. The number of observations in each direction is kept at a minimum of $max(i, j, k) + 1$ in order to preserve a well posed (over constrained) least squares problem. At runtime only a matrix multiply is needed to obtain the least squares coefficients. Exact and precomputed LSI matrices ensure a fast, accurate, and numerically well-conditioned coefficient generation process. For the current study, the value of m has been fixed at 7 for all the node domains and the maximum polynomial degree in each direction is fixed at 6.

Scalable spherical harmonics degree selection

Once all the LSI matrices have been generated, the measurement $\vec{u}$ for each node domain must be computed in order to obtain the coefficients for all the candidate functions using Eq. 5. The measurement vector requires evaluation of the SH function at each of the m^3 locations within each node domain. Noting there are on the order of ten million node domains (with 8 fold overlapping) for the global 200×200 resolution case, obtaining the measurement vector for m=7 requires evaluation of the SH function on the order of a billion times. Fortunately the full precision SH field is not necessary at all altitudes as the higher order terms become undetectable to machine precision as the altitude increases. Note that if m is odd, then the measurements in overlapping node domains are identical and only need to be calculated once.

Accordingly, a scalable SH degree selection method is adopted. Figure 6 shows the altitudes where the contributions of higher order terms of the SH expansion become less than a normalized 1e-15 (near machine precision for double). For example at a radial distance of $2Re$ (equivalently at an altitude of $1Re$), only terms up to degree and order 28 are necessary for the computation of the measurement vector. Using 1e-15 as a cutoff tolerance is overly conservative considering that the max normalized residuals in the least squares fits (to be discussed later) are always bigger than 1e-14.

The equation used to generate the curve in Fig. 6 is given in Eq. 7 and is found via a least squares curve fit of a large sampling of measurements taken across the global lat, lon and r domain. Equation 7 takes in normalized radial distance (r) as input and returns the degree and order value(F_{siz}) which is the max degree and order SH gravity field necessary for that radial distance.

$$F_{siz} = int(\frac{20.105042266558979}{log(r) + 0.007876575109652}) \tag{7}$$

It is evident from Fig. 6 that for most of the domain the high order spherical harmonics terms are not significant even in double precision arithmetic. As the evaluations move radially outwards, the measurement vector

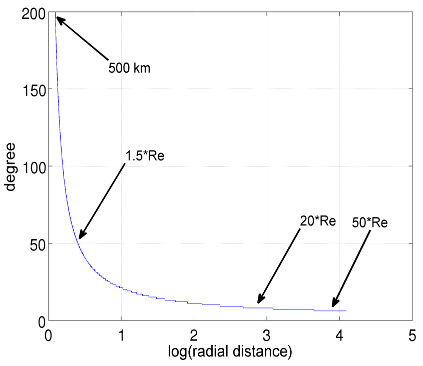

Figure 6. Degree and Order selection curve

computation time decreases rapidly thereby significantly speeding up the coefficient generation process. Note that Eq. 7 can be used to speed up the computation of SH for general applications as well.

Adaptive polynomial selection

The weighted interpolation approach affords the freedom to choose any degree polynomial for the local approximation function within each node domain. This benefit is utilized in the least square polynomial fitting process by adaptively choosing a different degree polynomial for each direction within each node domain (see Fig 1). Each of the candidate functions are ordered in terms of number of total coefficients. Starting with candidate functions with the fewest coefficients, the least squares problem is evaluated from the stored LWI matrices and Eq. 5. If a candidate function solution is found with its max residual less than the prescribed error tolerance, then the process is stopped and candidate functions with a greater number of coefficients need not be evaluated. In this manner, the max residuals of the local node solutions will always hover just below the user prescribed error, providing a uniform global error distribution. In addition, because the candidate functions are evaluated in order of increasing memory footprint, each node will have a minimized memory requirement. Therefore the coefficients for each local node are chosen so as to minimize its memory footprint but subject to a maximum residual constraint.

The runtime residual error tolerance is a function of two subtolerances, the potential residual error tolerance and the acceleration residual error tolerance. The potential error tolerance is adaptively determined at runtime based on the radial distance of the current node and the degree and order of the SH function being fit. The target value is chosen to conservatively mirror the expected errors of the SH function. Estimated accuracies of the GGM02C solution are given in[20] and the release notes. The accumulated error as a function of SH degree is replicated in Fig. 7. For example, up to degree and order 70, the accumulated error for the geoid height is 6 mm or 1e-9 in normalized units. Up to degree and order 200, the accumulated normalized error is approx. 3e-8. Therefore the confidence of the potential evaluation at the surface of a 200×200 and 70×70 field is approximately 8 and 9 digits of accuracy respectively. The accumulated error curve in Fig 7 serves as a target for the residual error level for geopotential evaluation at the surface. To be conservative the target potential residual error at the surface is chosen to be some factor less that that shown in Fig 7 (in this study 3 is the chosen reduction factor). In order to map the target residual errors to different altitudes a physics based approach is used where the errors are scaled from the surface to any altitude. For a given size of the SH field the error coefficients (Ce and Se) from the GGM02C data are obtained. For example for a 11×11 field there are 11 different pairs or Ce and Se terms. Norm of each Ce and Se pair is taken and multiplied by the radial scaling factor (Re/r). The procedure is repeated for the 11 Ce and Se pair and the final values are summed up to obtain the final cutoff tolerance for a 11×11 field . Hence, an error scaling graph is obtained specifying the cutoff tolerances as a function of radial distance.

Figure 8 shows such a graph for various SH fields. Cutoff errors beyond 4*Re are almost the same for all the various SH fields shown. This further validates the scalable SH field selection method described in the previous section. Finally, in order to preserve consistency in the accelerations with the SH codes near the surface, the error tolerance in the potential is never chosen to be smaller than 5e-10 for fields with order and degree less than or equal to 150 and 8e-10 for higher order fields.

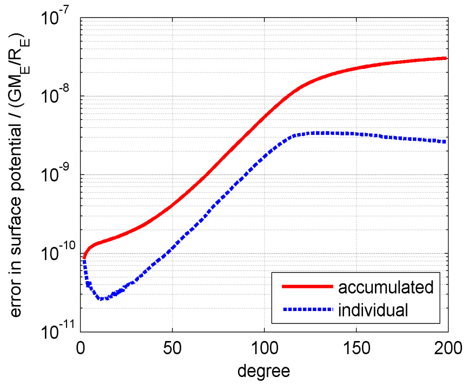

Figure 7. GGM02C accumulated error

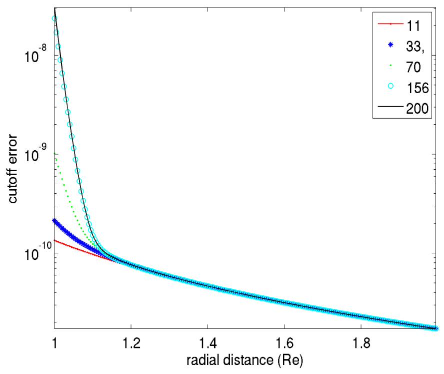

Figure 8. Error scaling graph

This final value obtained from the appropriate order function as depicted in Fig. 8 is the potential error tolerance used during the fitting process. The acceleration error tolerance is directly obtained by dividing the potential error tolerance by a user defined constant (100 for the current study) and is compared with the norm of acceleration error within each node domain. It can be argued, because the residuals of the interpolation model are is within the published accuracy of the SH model, that neither the fitting SH model nor the interpolation model is more likely to represent the true geopotential. Despite such an argument, the additional constraint to limit errors in the accelerations is included in order to maintain consistency in a uniform manner for the acceleration results from both GGM02C SH and the Fetch model. No constraints are placed on higher order derivatives because there is no justification for exact consistency amongst the higher order derivatives. To the contrary, applications such as orbit determination and trajectory optimization require exact derivatives of the function being used not the function being approximated.

Equations 8 and 9 give the full set of candidate polynomials considered during the fitting process. The polynomials given in Eq. 8 include coefficients for all possible cross terms, including user specified maximum degrees for each of the three directions. In addition, a reduced set of polynomials are given in Eq. 9 that includes limited cross terms. Deemed for this paper as Junkins polynomials, they only consider terms such that the degree of the terms of each direction add up to a specified total degree. The reduced set is found from[14] and provide an additional 6 candidate polynomials. Note that the Junkins polynomials can be represented by a subset of the polynomials in Eq. 8 except many of the cross term coefficients are fixed to zero. For the current study the maximum degree polynomial and the minimum degree polynomial for each direction is chosen as 6 and 2 respectively. A minimum degree limit on the polynomial ensures accuracy in the partials of the potential interpolant, which are further need to compute the accelerations at runtime.

$$P_f = \sum_{i=0}^{imax} \sum_{j=0}^{jmax} \sum_{k=0}^{kmax} c * {x_1}^{i-1} * {x_2}^{jj-1} * {x_3}^{kk-1} \tag{8}$$

$$P_r = \sum_{nn=0}^{imax} \sum_{ii=0}^{nn} \sum_{jj=0}^{nn-ii} c * {x_1}^{ii} * {x_2}^{jj} * {x_3}^{nn-ii-jj} \tag{9}$$

In Eqs. 8 and 9 x_1, x_2, and x_3 represent the normalized (-1 to 1) lat, lon and r value of the point of interest within a node domain. The variables $imax$, $jmax$ and $kmax$ represent the degree of the polynomial in lat, lon and r direction respectively. Finally, P_f and P_r corresponds to the full and the reduced (Junkins) polynomial functions respectively.

Figure 9 shows the surface residuals (in normalized units) in potential and acceleration at 300 km altitude for a Fetch model interpolating a 156×156 SH model. As expected maximum residual errors in both potential and acceleration are found to be less than the computed runtime error tolerances. Near the surface the acceleration errors drive the selection of the coefficients hence we observe regions in the geopotential error plot where its residuals are an order of magnitude less then those in the surrounding region. Note that the max error in the potential from Fig. 9 is found to be around 3e-10 normalized, or after multiplying by Re and given units is approximately 3e-7 mGals. From[19] the Cubed-Sphere 156×156 model evaluated at 300 km has an order of magnitude lower max error. Looking at Fig. 7 the estimated RMS accuracies of the GGM02C 156×156 model are approximately 2e-8 at the surface (and get mapped to slightly lower when evaluated at 300 km), therefore the Fetch model, with a max geopotential error of 3e-10, is conservatively (noting the RMS error is even lower) two orders of magnitude below the noise level of the SH function being fit. The Cubed-Sphere model is then three-orders of magnitude below the noise of the SH model. Therefore, both the Cubed-Sphere and Fetch models can be considered sufficiently accurate (if not overly accurate) with respect to the true geopotential.

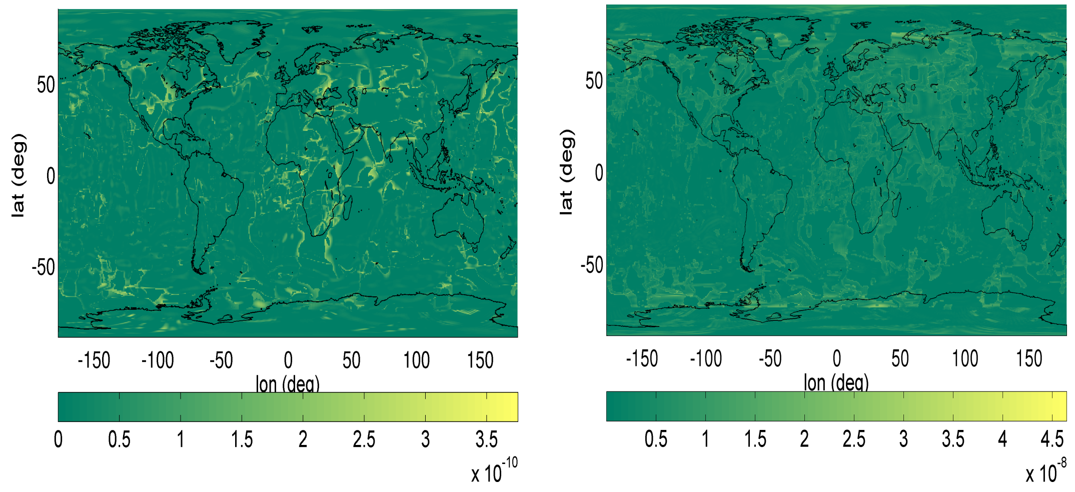

Figure 9. Potential(left) and Acceleration(right) error at 300 km altitude : 156×156 **field**

Figure 10 shows the residual statistics for a randomized sweep of the global domain. Each point contains the max or RMS of 1000 random evaluations at different lat-lon locations at the specified radius after the coefficient generation process for a 156×156 SH field. Near the surface the acceleration error tolerance is the limiting one of the two residual error criteria and therefore the potential residuals are significantly smaller in some of these regions. Moving further away from the surface, the residuals drop according to the target levels. The dip occurs when the evaluation points fall close to the shell boundaries where the accuracy of the interpolant is better.

Coefficient storage and Parallel coefficient generation

The computed coefficients along with the required meta-data are stored in a single dense binary file. The coefficients are stored using an array data structure with allocatable subarrays to eliminate any memory wastage. This structure leads to a 20% increase in the file read time but the memory savings justify the extra time to load the file for use with the runtime evaluations.

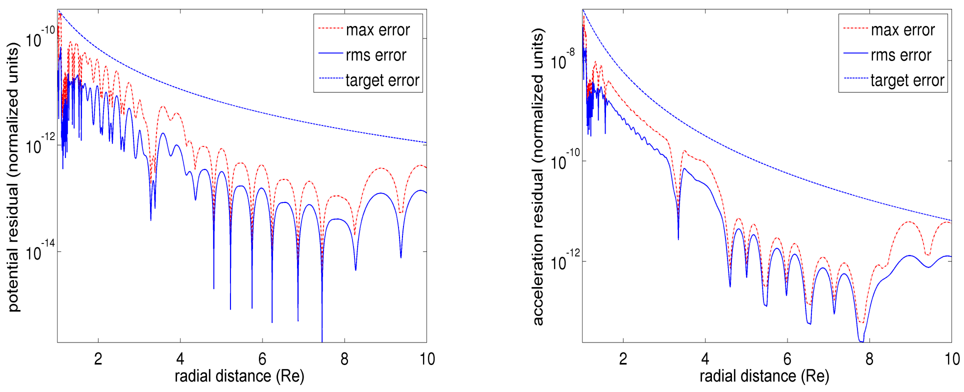

Figure 10. Absolute Potential(left)/Acceleration(right) normalized residual error: 156×156 **field**

In spite of various algorithmic optimizations as stated in the previous sections, the coefficient generation process for a complete 200×200 global model would require approx 20 to 30 days on a single modern CPU. Note that the 200×200 degree and order model contains approximately 9 million nodes and up to 222 candidate functions are evaluated for each node where each candidate function contains up to 216 coefficients. As the coefficients need to be generated just once, the process is parallelized using the Message Passing Interface (MPI) programming model. The global grid is divided longitudinally amongst all the processors and each processor is responsible for locally fitting the node polynomials which fall in its domain. Each processor then writes separate coefficient files and the master processor is finally responsible for joining all the files into one coefficient binary file. This domain decomposition is ideally suited for explicit parallelism as there is little required communication between the processors. The parallel version of the code is implemented in FORTRAN 2003 and can be compiled using either the Intel MPI compiler of the OPEN MPI compiler. The complete process for fitting a 200 degree and order field across the full domain takes 16 hrs on a cluster of 200 processors. Table 1 lists the various Fetch models that are generated for this release and are classified based on the degree and order of the SH field that is fit. Detailed performance statistics are given in later sections.

Table 1. Generated Models

Model Name	Element size (Degree)	Memory needed
GF11a	1.8	332 Mb
GF33a	1.5	588 Mb
GF70a	0.9	1.15 Gb
GF156a	0.6	1.67 Gb
GF200a	0.58	1.85 Gb

Figure 12 shows the distribution for number of coefficients in each cell for the 156×156 model.The distribution shows a strong and relatively even preference for between approximately 10 and 55 coefficients. Some of the nodes required as many as 84, although none of them used the higher degree $polytypes$ noting that up to 216 coefficients were available for consideration.Figure 11 shows the contour of the geopotential (with two body and J_2 terms removed) evaluated at surface for 156×156 field. The next section discusses the runtime and usage aspects of the model.

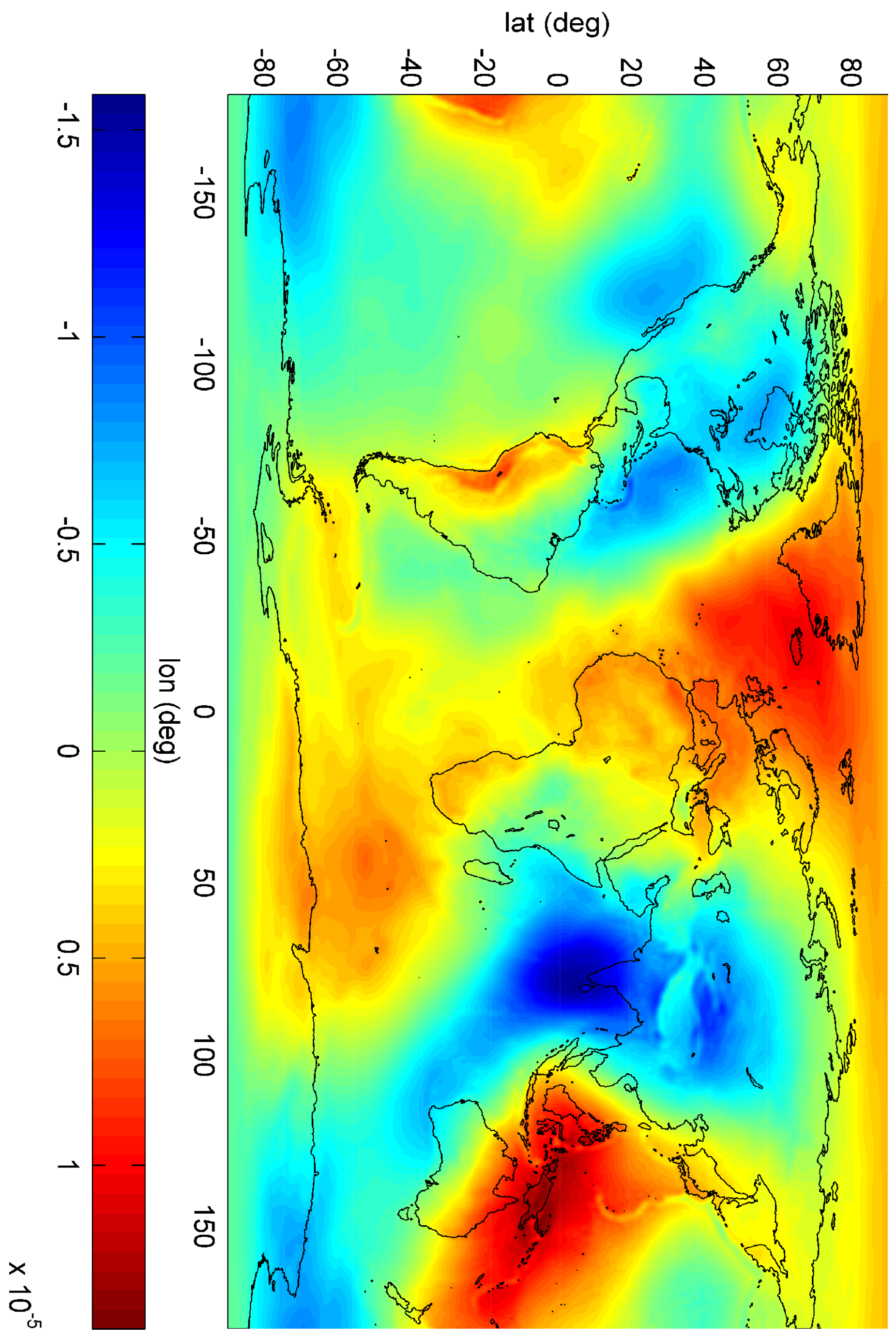

Figure 11. Contour of the geopotential for 156×156 field (with two body and J_2 terms removed)

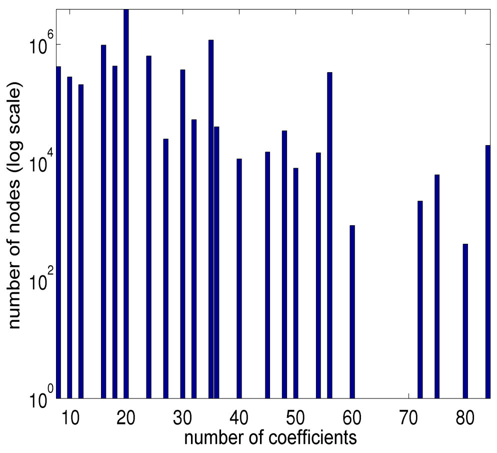

Figure 12. Coefficient distribution

PHASE 2: RUNTIME EVALUATION

The runtime implementation of a Fetch model is designed to be as simple as possible. The steps required at runtime are as follows:

1. Initialize the global grids
2. Load model binary file
3. Call Fetch runtime routines to retrieve the potential and any desired derivatives.

The first step initializes the primary and rotated grids to be used by the global model. The necessary parameters can be passed or changed by the user via a namelist. The second step loads the binary file requested by the user to the computer's Random Access Memory (RAM). Both these steps are executed only once during the model initialization phase. Once the model is initialized various calls to Fetch routines can proceed. Inside the core runtime routines the coefficients are efficiently tracked and the correct 8 neighboring node polynomials are identified and evaluated followed by a final weighted evaluation of the composite Fetch runtime function.

Coefficient lookup

All routines take the spacecraft position vector as part of their input and subsequently identify the 3D cell housing that position vector. Normally, such a task would require a big lookup table ; however the uniform surface grid spacing along with the precomputed radial shell distances can be exploited to identify the correct cell by only using a double to integer conversion and other simple equations. Hence, the complexity of a lookup table is bypassed and thereby the runtime performance is significantly improved.. Furthermore, the coefficients are stored in manner such that only one node position lookup is required in order to gather all of the neighboring eight node positions.

Continuity

Continuity in zeroth and higher order derivatives is a desirable feature for any force model. Global continuity in the first derivative is necessary to accurately represent a conservative field, and continuity in higher order derivatives may be important, depending on the specific application. Most 3D interpolation based methods are not globally continuous or require complex algorithms to achieve continuity. The tricubic interpolation method by Lekien and Marsden[16] is an example that achieves first order continuity but at the expense of reduced accuracy performance. Furthermore, in their method the interpolants must be of the same degree globally, and accurate high order derivatives of the fitting data are required for the coefficient generation process. The Cubed-Sphere model is discontinuous even to the zeroth order across the shell boundaries, although the discontinuity is small in the published models. Also, like the tricubic method, the degree of the interpolants for the Cube-Sphere model are fixed globally. In the case of the Fetch model, continuity in any order derivative is achieved by utilizing the weighted interpolation scheme (developed by Junkins[13]) which leads to eight interpolant function evaluations instead of one for each composite function call. The eight interpolants applicable to each cell are evaluated in their own -1 to 1 normalized space and the weighting is done in the overlapping space of the individual eight node domains , normalized from 0 to 1. The 0 to 1 normalization allows the use of Hermite weighting functions which enable continuous higher order derivatives. Figure 13 summarizes the weighting function technique for two dimensions.

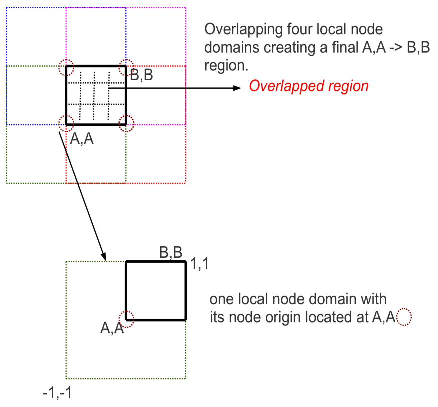
Overlapping four local node domains creating a final A,A -> B,B region.
Overlapped region
B,B
A,A
B,B
1,1
one local node domain with its node origin located at A,A
A,A
-1,-1

Figure 13. Continuity from weighted evaluation

Because the primary and rotated grids are used to overcome the singularity problem (as discussed before), there exists a latitude band above and below the equator where both the primary and rotated grids overlap. To achieve continuity across this overlapped region a simple 1D weighting function is used and is based on evaluations from both grids. Equation 10 summarizes the final weighted evaluation of the function in the small region of overlap (a single latitude band, see Fig. 4).

$$U_{wt} = U_{pri}(\phi) * wt(\phi) + [1 - wt(\phi)] * U_{rot}(\phi) \tag{10}$$

where ϕ refers to normalized latitude which varies between 0 and 1 across the overlapped region. U_{pri} and U_{rot} are the interpolated geopotentials evaluated on the primary and the rotated grid respectively. U_{wt} is the final weighted geopotential in the overlapping region, and wt represents a 7th order 1D Hermite weight function given by Eq. 11.

$$wt = x^4 * (35 - 84 * x + 70 * x^2 - 20 * x^3) \tag{11}$$

Equation 10 can be further differentiated to obtain continuous derivatives up to the desired order. Hence, due to the weighting evaluation and the overlapping techniques, continuity and exact higher order derivatives of the interpolated geopotential are maintained globally. Exact higher order derivatives possess attractive dynamical properties especially for applications such as orbit determination and trajectory optimization. Figure 14 illustrate the continuity by showing the potential and its first three derivatives with respect to x for the overlapping region. Figure 14 illustrates continuity and smoothness in all the four cases and is achieved by choosing a 7th order Hermite weighting function. Kinks and discontinuities in the higher order derivatives are illustrated in Fig. 14 and they appear due to using only a 3rd order weighting function at runtime.

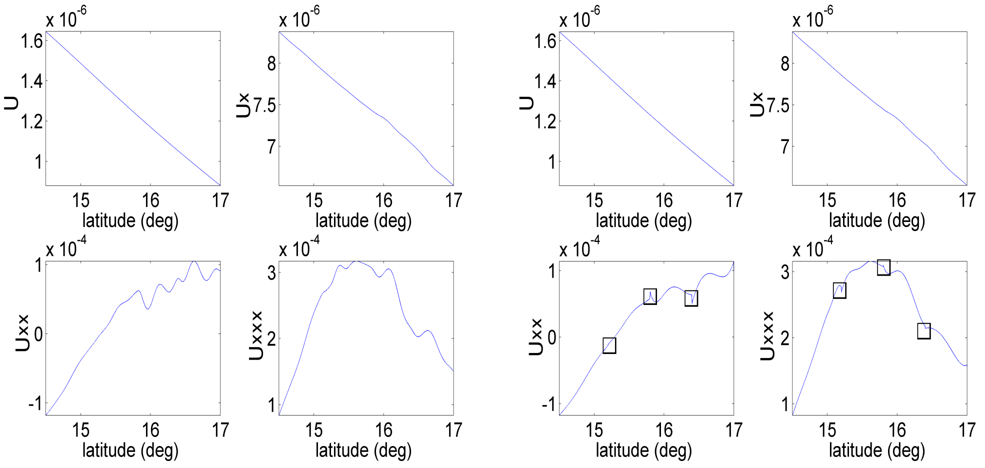

Figure 14. 7th Order(left)/3rd order(right) weighting functions: 156×156 **field**

Runtime routines

Table 2 lists the runtime routines that have been implemented for the current release. All of these routines take the normalized spacecraft position vector as input and give the normalized interpolated potential plus derivatives as output. The normalization length and time units are derived by setting the reference gravitational parameter and radius associated with the GGM02C SH model to unity (1 LU=6378.1363 km, 1 TU=806.810991306733 s).

Table 2. Available FETCH routines

Routine Name	Output
get_local_u	potential only
get_local_ua	potential + acceleration
get_local_uad	potential + acceleration + higher order derivatives

The accelerations are computed by taking the gradient of the interpolated geopotential function with respect to spherical coordinates, then performing the necessary transformations to the Cartesian coordinates.[21] Similar coordinate transformations and chain rules are required for any necessary higher order derivatives. In this implementation, runtime routines are provided to include up to third order derivatives (with respect to Cartesian coordinates) of the potential. Again, it is emphasized that the singularity issues with the spherical coordinates conversions known to be problematic near the poles is handled through the use of the two overlapping grids.

PERFORMANCE COMPARISON

In this section the performance of the Fetch models are evaluated against an optimized CPU implementation of the Pines SH model.[2] It is noted that both the Pines SH algorithm and the Fetch model are singularity free. Even though the Fetch model extends all the way to the moon the performance comparisons are limited to an altitude of $2{*}Re$. All five of the Fetch models are considered for performance comparison from table 1. Table 3 gives the specifications of the runtime test hardware and compiler.

The performance evaluation region is divided into 5 altitude bands listed in table 4. For each band sample direct calls to Pines SH and Fetch models are made. Both speed and error comparison are performed. The number of sample points for each band is selected heuristically depending on the degree and order of the field.

Table 3. Test hardware/software specifications

Component type	Component
CPU	Intel Xeon E5520 @ 2.27 Ghz
RAM (Memory)	12.0 GB
Compiler	Intel FORTRAN 12.0

Table 4. Performance evaluation regions

Band id	Start altitude (km)	End altitude (km)
Band 1	150	500
Band 2	500	1500
Band 3	1500	3000
Band 4	3000	6378
Band 5	6378	12756

Figures 15 to 19 give the speedup over SH implementation for the fitted geopotential only and fitted geopotential plus acceleration cases. An exponential increase in speedup with increasing order of the SH field is observed as the computation time for Fetch is not heavily dependent on degree and order of the SH field. It also observed that the evaluation time of Fetch model is similar to that of an 11×11 field evaluated using the Pines SH algorithm. The break even size of field is approximately 10×10 indicating approximately a four fold speed increase over the Cubed-Sphere model that has a published break even size of 20×20.[19] The underlying adaptivity in polynomial selection and radial step direction ensures that the speedup remains relatively constant across bands. The Fetch model achieves an order of magnitude or more speedup for medium order fields (70×70) and 2 orders of magnitude or more speedup for higher order fields (200×200).

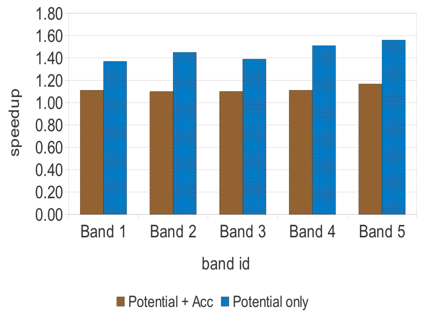

Figure 15. Case: 11 degree and order field

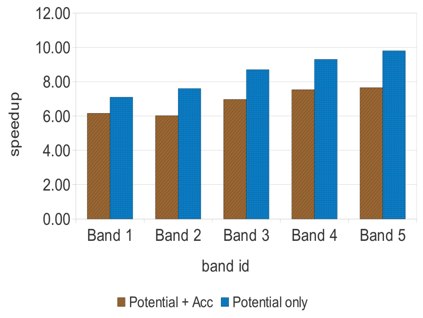

Figure 16. Case: 33 degree and order field

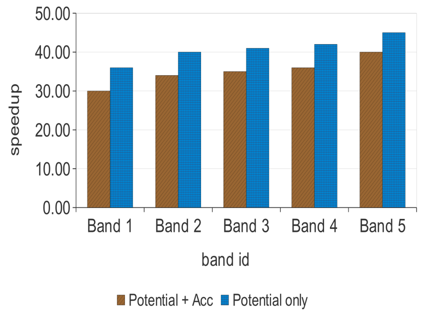

Figure 17. Case: 70 degree and order field

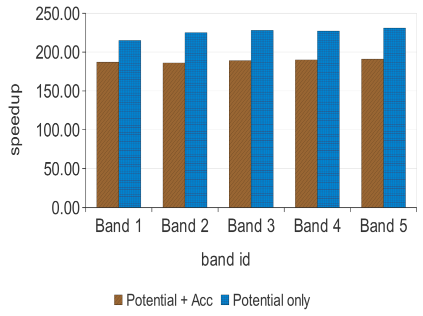

Figure 18. Case: 156 degree and order field

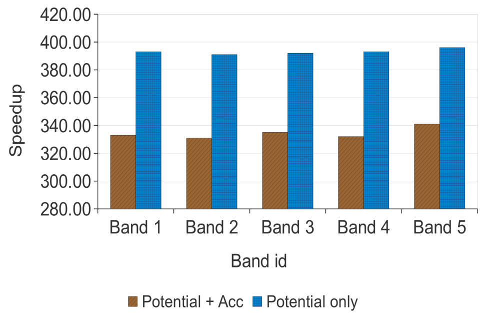

Figure 19. Case: 200 degree and order field

Tables 5 to 8 list the max and RMS residual errors in potential and acceleration corresponding to each of the bands and for the various SH models. As expected, the error in potential is always found to be less than the target error and the error in acceleration is generally one to two orders of magnitude worse than that in the potential. The errors in lower fidelity fields are much smaller due to the lower target errors associated with the lower fields.

Table 5. Max normalized error in potential

	11×11	33×33	70×70	156×156	200×200
Band 1	8.0E-012	6.7E-012	3.1E-011	3.0E-010	6.0E-010
Band 2	7.0E-012	6.2E-012	3.1E-011	2.5E-010	5.7E-010
Band 3	6.0E-012	2.8E-012	2.9E-011	1.1E-010	1.2E-010
Band 4	6.0E-012	3.7E-012	2.5E-011	9.0E-011	1.0E-010
Band 5	4.0E-012	1.8E-012	1.9E-011	7.0E-011	8.0E-011

Table 6. RMS normalized error in potential

	11×11	33×33	70×70	156×156	200×200
Band 1	4.7E-013	7.4E-013	5.1E-012	4.9E-011	8.0E-011
Band 2	2.0E-013	8.7E-013	5.3E-012	2.3E-011	4.3E-011
Band 3	3.0E-013	1.5E-013	2.6E-012	2.1E-011	5.5E-011
Band 4	1.5E-013	4.6E-013	1.2E-012	1.1E-011	9.0E-011
Band 5	8.7E-013	3.8E-013	9.6E-013	9.2E-012	1.0E-011

Table 7. Max normalized error in acceleration

	11×11	33×33	70×70	156×156	200×200
Band 1	2.3E-09	3.0E-09	1.1E-08	4.0E-08	7.0E-08
Band 2	2.0E-09	2.5E-09	9.0E-09	3.0E-08	6.5E-08
Band 3	1.1E-09	1.4E-09	8.5E-09	1.2E-08	2.0E-08
Band 4	9.0E-10	1.1E-09	6.0E-09	9.0E-09	8.4E-09
Band 5	7.0E-10	9.0E-10	3.0E-09	6.5E-09	7.0E-09

Table 8. RMS normalized error in acceleration

	11×11	33×33	70×70	156×156	200×200
Band 1	1.8E-10	2.0E-10	1.1E-09	8.5E-09	3.4E-09
Band 2	1.8E-10	1.8E-10	1.0E-09	5.5E-09	3.5E-09
Band 3	1.8E-10	2.6E-10	7.9E-10	3.1E-09	4.0E-10
Band 4	2.9E-10	1.7E-10	7.1E-10	2.1E-09	1.1E-10
Band 5	1.2E-10	9.2E-11	4.8E-10	1.0E-09	8.0E-11

The higher order derivative routine is also tested and it is found that including the Jacobian or the Jacobian and the Hessian of the acceleration cost an additional 20% and 40% compute time respectively when compared to the potential and acceleration only case. The speedup factors demonstrate that the Fetch model is capable of providing multiple orders of magnitude speedups over at least one implementation of a tuned Pines SH code while still preserving accuracy and continuity over the whole solution domain extending all the way up to the Moon. Furthermore, the model derivatives (in this study up to third order are demonstrated) are exact and both easy and fast to compute.

CONCLUSION

The main objectives of the Fetch model is to provide a fast and accurate way for calculating the geopotential while preserving the mathematical niceties associated with the SH formulation. The Fetch method trades an affordable memory investment for significant speed gains. It uses a weighting function technique to achieve continuity over the whole solution domain and allow for localized resolution. The geopotential is the only interpolated function and the acceleration and any higher order derivatives are calculated by explicitly differentiating the interpolant. The singularity present at the poles when conventionally dealing in spherical coordinates is tackled by implementing a two level grid structure with an overlapping latitude band. Therefore, the four priorities laid out in the introduction are achieved. When compared to the Cubed-Sphere model, the Fetch model is approximately four times faster in computing the geopotential, while the Cubed-

Sphere model has a more favorable memory footprint (requiring approximately half as much memory when comparing the 150×150 model to the 156×156 Fetch model). The Cubed-Sphere model interpolates each term of every derivative and therefore its memory footprint scales with number of required derivative terms (above its nominal potential plus acceleration). Furthermore, its explicit derivatives are not exact, nor are they smooth or continuous across certain boundaries. Alternatively, the Fetch model provides an arbitrary level of smoothness and continuity for any desired level of derivative, and the memory footprint is fixed at the cost for storing the potential only.

Adaptivity for the Fetch model is achieved both through the radial direction and through the optimization of choosing the best out of 222 candidate interpolants for each node. Precomputed analytic solutions to the least squares normal equations afford the extreme flexibility of evaluating all candidate interpolants for up to 10 million nodes. Furthermore, target error tolerances for each node are obtained from a physics based logic originating with the published accuracy of the GMM02C gravity model. These error tolerances are used to guide the fitting process, so as to minimize the number of coefficients needed for each node while maintaining a global residual error profile that is consistent with the accuracy of the SH function being fit. A MPI based parallel version of the algorithm is implemented to efficiently generate the coefficients for the high order spherical harmonic fields.

Orders of magnitude of speedups are observed for small to medium order fields while the higher degree fields demonstrate multiple of orders of magnitude in speedup. The highest resolution investigated (and native size of the GGM02C model) is 200×200 where the Fetch model achieves over 300x speedups. The residual errors in potential and acceleration are always within the order of the tolerance specified. The ease of implementation combined with extreme speed and favorable continuity properties makes the Fetch model attractive for any high fidelity orbit determination or trajectory optimization task. Problems previously thought intractable due to their spherical harmonics based computational burden can now be tackled and solved. The authors intend to immediately make the runtime routines and coefficient files public. Please contact them for details.

ACKNOWLEDGEMENT

The authors thanks Haijun Shen and Analytical Mechanics Associates, Inc.; Jack Van Wieren and colleagues from the Air Force Space Command; and Kent Miller from the Air Force Office of Scientific Research (AFOSR) for collaborations and support. This material is in part based upon work supported by AFOSR under Contract No. FA9550-10-C-0061.

NOTATION

x, y, z	Cartesian space vectors
Lat, Lon	Latitude and Longitude
i, j, k	Latitude and Longitude and radial direction index
x_1, x_2, x_3	-1 to 1 normalized latitude, longitude and radial distance
r	Normalized radial distance
N	Total number of coefficients per cell
m	Total number of measurements in each direction
c	Array of coefficient vector
B	LSI matrix
Re	Radius of Earth
SH	Spherical Harmonics
U_{pri}	Potential evaluated on primary grid
U_{sec}	Potential evaluated on rotated grid
U_{wt}	Potential evaluated on the overlapped grid
U_{node}	Fitted polynomial associated with each node point

U_{fetch}	Final composite polynomial computed using a weighted average
wt	Hermite weighting function
LSI	Least square inverse
$imax$	Polynomial degree in lat direction
$jmax$	Polynomial degree in lon direction
$kmax$	Polynomial degree in r direction
$PNmax$	Maximum degree of polynomial allowed in each direction

REFERENCES

[1] S. Casotto and E. Fantino, "Evaluation of methods for spherical harmonic synthesis of the gravitational potential and its gradients," *Advances in Space Research*, Vol. 40, 2007, pp. 69–75.

[2] S. Pines, "Uniform Representation of the Gravitational Potential and its Derivatives," *AIAA Journal*, Vol. 11, 1973, pp. 1508–1511.

[3] J. B. Lundberg and B. Schutz, "Recursion Formulas of Legendre Functions for Use with Nonsingular Geopotential Models," *Journal of Guidance, Control, and Dynamics*, Vol. 11, 1, pp. 32–38.

[4] F. R. Hoots, P. W. Schumacher, and R. A. Glover, "History of Analytical Orbit Modeling in the U.S. Space Surveillance System," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 2, 2004, pp. 174–185.

[5] K. R. Koch, "Simple Layer Model of the Geopotential in Satellite Geodesy," *The Use of Artificial Satellites for Geodesy, Geophysics Monograph Series*, Vol. 15, 1972, pp. 107–109.

[6] L. Wong, G. Buechler, W. Downs, W. Sjogren, P. Muller, and P. Gottlieb, "A SurfaceLayer Representation of the Lunar Gravitational Field," *Journal of Geophysical Research*, Vol. 76, No. 26, pp. 6220–6236.

[7] K. R.Koch and B. U. Witte, "Earth's Gravity Field Represented by a Simple Layer Potential from Doppler tracking of Satellites," *Journal of Geophysical Research*, Vol. 76, No. 35, 1971, pp. 8471–8479.

[8] K. R.Koch and F. Morrison, "A Simple Layer Model of the Geopotential from a combination of Satellite and Gravity Data," *Journal of Geophysical Research*, Vol. 75, No. 8, 1970, pp. 1483–1492.

[9] F. Morrison, "Algorithms for Computing the Geopotential using a Simple Density Layer," *Journal of Geophysical Research*, Vol. 81, No. 26, 1976, pp. 4933–4936.

[10] R. Werner and D. Scheeres, "Exterior Gravitation of a Polyhedron Derived and Compared with Harmonic and Mascon Gravitation Representations of Asteroid 4769 Castalia," *Journal of Geophysical Research*, Vol. 81, No. 26, 1976, pp. 4933–4936.

[11] R. S. Park and D. J. Scheeres, "Nonlinear Semi-Analytic Methods for Trajectory Estimation," *Journal of Guidance, Control and Dynamics*, Vol. 30, 2007, pp. 1668–1676.

[12] R. P. Russell and N. Arora, "Global Point Mascon Models for Simple, Accurate, and Parallel Geopotential Computation," *AAS/AIAA Space Flight Mechanics Meeting*, Vol. AAS 11-158, 2011.

[13] J. L. Junkins, "Investigation of Finite-Element Representations of the Geopotential," *AIAA Journal*, Vol. 14, No. 6, 1976, pp. 803–808.

[14] J. L. Junkins and R. C. Engels, "Local Representation of the Geopotential by Weighted Orthonormal Polynomials," *Journal of Guidance, Control, and Dynamics*, Vol. 3, No. 1, 1980, pp. 55–61.

[15] G. Beylkin and R. Cramer, "Toward multiresolution estimation and efficient representation of gravitational fields," *Celestial Mechanics and Dynamical Astronomy*, Vol. 84, No. 1, 2002, pp. 87–104.

[16] F. Lekien and J. Marsden, "Tricubic Interpolation in Three Dimensions," *International Journal for Numerical Methods in Engineering*, Vol. 63, 2005.

[17] A. Colombi, A. H. Hirani, and B. F. Villac, "Adaptive Gravitational Force Representation for Fast Trajectory Propagation Near Small Bodies," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 4, 2008, pp. 1041–1051.

[18] R. S. Hujsak, "Gravity Acceleration Approximation Functions," *AAS-96-123*.

[19] B. A. Jones, G. H. Born, and G. Beylkin, "Comparisons of the Cubed-Sphere Gravity Model with the Spherical Harmonics," *Journal of Guidance, Control, and Dynamics*, Vol. 33, 2, pp. 415–425.

[20] B. Tapley, J. Ries, S. Bettadpur, a. M. C. D. Chambers, F. Condi, B. Gunter, Z. Kang, P.Nagel, R. Pastor, T. Pekker, S.Poole, and F. Wang, "GGM02-An improved Earth gravity field model from GRACE," *Journal of Geodesy*, Vol. 79, No. 8, 2005, pp. 467–578.

[21] G. Spier, "Design and Implementation of Models for the Double Precision Trajectory Program (DPTRAJ)," *Technical Memorandum*, Vol. 33-451, 1971.