Sequential Orbit Determination with the Cubed-Sphere Gravity Model

Brandon A. Jones\textsuperscript{1}, George H. Born\textsuperscript{2}, and Gregory Beylkin\textsuperscript{3}

\textit{University of Colorado, Boulder, CO, 80309, USA}

The cubed-sphere model provides rapid evaluation of the gravity field for more efficient orbit propagation. This paper characterizes the improved computational efficiency of sequential orbit determination, specifically the extended and unscented Kalman filters, when using this new model instead of the common spherical harmonic model. To use the new gravity model with the extended Kalman filter, capabilities to represent the Jacobian of the gravity acceleration are added to numerically integrate the state transition matrix. Filter tests consider improvements for several simulated satellite scenarios with several combinations of measurements provided for estimation. Since cubed-sphere models of higher degree require only a slight change in computation time, orbit propagation and determination systems may now use this model to improve fidelity without any significant change in cost. Using the cubed-sphere model reduces the computational burden of the orbit determination process, with larger benefits found for high-degree filter models. Differences in the estimated trajectories when using the disparate gravity models remain several orders of magnitude less than the absolute filter error for the cases examined.

\begin{flushleft}
\textsuperscript{1} Graduate Research Associate, Colorado Center for Astrodynamics Research, UCB 431, AIAA Member.
\textsuperscript{2} Professor, Colorado Center for Astrodynamics Research, UCB 431, AIAA Fellow.
\textsuperscript{3} Professor, Department of Applied Mathematics, 526 UCB.
\end{flushleft}
Nomenclature

\( \alpha \) = unscented transformation scale factor
\( \beta \) = generic parameter
\( \lambda \) = offset scale factor used in the unscented transformation
\( \rho \) = atmospheric density (kg/m\(^3\))
\( \sigma \) = standard deviation
\( \Phi(t_i, t_{i-1}) \) = state transition matrix mapping from \( t_{i-1} \) to \( t_i \)
\( A(t_i) \) = matrix of force model partial derivatives at time \( t_i \)
\( a \) = acceleration vector (km/s\(^2\))
\( b \) = degree of B-spline interpolation on the cube face
\( C_D \) = coefficient of drag
\( C \) = vector of gravity model coefficients
\( CD \) = number of common digits
\( e \) = random vector with zero mean and unit \( \sigma \)
\( F(X, t) \) = force function/model
\( L \) = lower-triangular matrix generated by the Cholesky decomposition
\( l \) = degree of Chebyshev interpolation between shells
\( M \) = number of cubed-sphere model primary shells
\( N \) = Cubed-sphere model grid size
\( n \) = number of filter estimated parameters
\( P \) = filter state-error variance-covariance matrix
\( \bar{P} \) = filter state-error variance-covariance matrix after a time update
\( p \) = number of values represented by the cubed-sphere model
\( Q \) = filter process noise matrix
\( r \) = position vector (km)
\( S_F \) = filter speedup factor
\( S_I \) = integrator speedup factor
\( T \) = time of execution (s)
\( \tilde{T} \) = normalized time of execution
\( t_i \) = \( i \)-th value of time
\( U \) = Gravity potential (km\(^2\)/s\(^2\))
\( \mathbf{v} \) = velocity vector (km/s)
\( W^c_j \) = weights used to compute the time updated covariance matrix in the unscented transformation
\( W^m_j \) = weights used to compute the time updated mean state vector in the unscented transformation
\( \mathbf{X} \) = filter state vector
\( \hat{\mathbf{X}} \) = estimated filter state vector
\( \bar{\mathbf{X}} \) = time updated, or a priori, filter state vector
\( \mathbf{X}^* \) = reference state vector
\( \mathbf{X}_{j,i-1} \) = \( j \)-th sigma point at time \( t_{i-1} \)

I. Introduction

Demands for improved satellite force model accuracy and computational efficiency are increasing. Missions like COSMIC [1] require reduced orbit determination latency to provide near real-time observations of space and terrestrial weather [2]. The NASA Jet Propulsion Laboratory generates near real-time orbit determination solutions for the Jason-2 satellite [3] and the GPS constellation [4]. In the case of Jason-2, the orbit determination system uses a 200×200 spherical harmonic model [5]. Some systems observe and generate orbit determination solutions of multiple satellites. The US Air Force maintains a catalog of all known space objects, and propagates the state and state-error covariance matrix of each forward in time for future tracking and collision prediction. In 2004, the Air Force tracked approximately 10,000 space objects [6]. In just five years, that estimate almost doubled to 19,000 objects with a predicted 100,000 objects tracked with improved monitoring capabilities [7]. As the catalog continues to expand, so does the computational burden of these surveillance operations. For this reason, such systems require rapid orbit propagation for both catalog estimation and propagation. Force model computational efficiency also limits simulation capabilities, especially for real-time and Monte Carlo studies needed for requirements
definition, trade studies, and flight software verification and validation. All of these systems require computationally efficient models of the satellite dynamics.

In the early days of gravity model development, memory greatly limited computing capabilities. This led to the development of gravity estimation and modeling techniques that minimize memory requirements, i.e. the current forms of the spherical harmonic model. However, this model now fails to adapt to modern demands for computational efficiency. High-degree spherical harmonic models account for a majority of force model computation. Since modern computers have memory limitations on the order of gigabytes, this is no longer a concern. Additionally, modern computers allow for parallelization. New methods for improving the efficiency of gravity field evaluation (such as [8–15]) leverage off of these changes in modern computers. Specifically, they swap computation time for larger memory requirements.

This paper presents results that demonstrate the computation improvements in sequential orbit determination when using the cubed-sphere model introduced in [16]. To verify that reductions in computation time yield only small deviations in the filter solution, this paper also compares orbit determination solutions generated when using the cubed-sphere model to those produced via the spherical harmonic model. This study expands on the applications of the cubed-sphere model, both in terms of the fidelity of available models and an approximation of the variational equations required to generate a state transition matrix.

II. Cubed-Sphere Gravity Model

A. Model Description

Originally proposed in [16], the cubed-sphere model defines a new method to compute geopotential and acceleration. The sphere is mapped to a cube with a new coordinate system defined on each face. As seen in Fig. 1, the mapping to a cube yields no singularity at the poles and creates a more uniform distribution of points on the surface. Each face is segmented by an uniform grid with interpolation performed between grid points to find the acceleration. Multiple spheres, each mapped to a cube, are nested within each other with interpolation between adjacent shells accounting for the acceleration variation in the radial direction. A grid spacing scheme is established with repre-
sentations of acceleration precomputed at intersections of the grid lines. Basis splines, or B-splines, are used to represent functions on each face of the cube. Similar to the degree and order of the spherical harmonics model, the fidelity of the cubed sphere is determined by the grid size $N$, the degree of the B-spline interpolation on the cube face $b$, the degree of the Chebyshev interpolation between concentric shells $l$, and the number of primary concentric shells $M$. The cubed-sphere model is generated from another base model and saved for future use in orbit propagation. A more detailed description of the cubed-sphere model, including orbit propagation tests and a discussion of its capabilities and limitations, may be found in [17].

![Fig. 1 Illustration of the mapping of a cube to the sphere. [17]](image)

The size of the cubed-sphere model may be described by the number of B-spline interpolation coefficients

$$\text{Number of coefficients} = 6p(l + 1)(M - 1) \left( \frac{N}{4} + b \right)^2$$

where $p$ is the number of parameters modeled by the cubed sphere. Each value approximated by the cubed-sphere model uses a submodel, e.g. one model for each of the three components of acceleration. The models of [17] represent both the acceleration and the potential, which yields a value of four for $p$.

The results of [17] demonstrate that the cubed-sphere model provides rapid orbit propagation capabilities with only small changes in integrated trajectories. These tests profiled cubed-sphere models equivalent to the $20 \times 20$, $70 \times 70$, and $150 \times 150$ spherical harmonic models. Monte-Carlo-like tests demonstrated that using the cubed-sphere model provided orbit propagation as much as 30 times faster than when using the spherical harmonic model. Resulting trajectories deviate from
B. Model Configurations

The models used for this paper differ slightly from those previously described in the literature. First, characterizing the cubed-sphere model in the orbit determination process requires a larger selection of models, i.e. models based on 20×20 through 200×200 spherical harmonic models in increments of 10×10. The models of [17] fixed the degree of the Chebyshev interpolation for all base model degrees and only changed the grid density on the surface of the cube. This parameter now varies with each model to satisfy precision requirements. Models using Chebyshev polynomials of degree ten or less exhibit improved computational efficiency when compared to those of [17]. This research uses highly-precise models, i.e. cubed-sphere models that agree closely with the base model. To this end, cubed-sphere models are configured so that acceleration vectors match those of the spherical harmonics to 14 digits or more (8.9×10⁻¹⁴ m/s²) at all points at an altitude of 300 km or greater. See [17] for more information on the model generation process. Additionally, the previous cubed-sphere gravity models used the common Legendre form of the spherical harmonic model (see, for example, [18, 19]) for the base model, which contains an artificial singularity at the poles. These models instead use the nonsingular Cartesian representation of [20] as the base model. Table 1 describes the new cubed-sphere model configurations.

The cubed-sphere models described in Table 1 yield a difference from the spherical harmonic model several orders of magnitude below the most accurate gravity models. Fitting closely to the spherical harmonic model aids in its acceptance. The need to justify slight differences (even if they happen to be beneficial) is a higher burden than supplying a model that matches the accepted spherical harmonic model within the user specified precision. Additionally, such a tight precision approaches the limits of floating-point accuracy in the evaluation of the gravity model itself. The papers [21] and [22] characterize the accuracy of model evaluation when using double-precision values by comparing to quadruple-precision-based solutions. On average, those results demonstrate an accuracy or 14-15 digits when evaluating the potential and acceleration.
Table 1 Precise cubed-sphere gravity model configurations for degree 20 through 200.

<table>
<thead>
<tr>
<th>Model</th>
<th>Base Model</th>
<th>l</th>
<th>b</th>
<th>M</th>
<th>N</th>
<th>Size (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS-34</td>
<td>20×20</td>
<td>6</td>
<td>11</td>
<td>14</td>
<td>136</td>
<td>34</td>
</tr>
<tr>
<td>CS-48</td>
<td>30×30</td>
<td>7</td>
<td>11</td>
<td>14</td>
<td>192</td>
<td>67</td>
</tr>
<tr>
<td>CS-62</td>
<td>40×40</td>
<td>8</td>
<td>11</td>
<td>14</td>
<td>248</td>
<td>115</td>
</tr>
<tr>
<td>CS-76</td>
<td>50×50</td>
<td>8</td>
<td>11</td>
<td>14</td>
<td>304</td>
<td>163</td>
</tr>
<tr>
<td>CS-94</td>
<td>60×60</td>
<td>8</td>
<td>11</td>
<td>14</td>
<td>376</td>
<td>237</td>
</tr>
<tr>
<td>CS-100</td>
<td>70×70</td>
<td>9</td>
<td>11</td>
<td>14</td>
<td>400</td>
<td>294</td>
</tr>
<tr>
<td>CS-112</td>
<td>80×80</td>
<td>9</td>
<td>11</td>
<td>14</td>
<td>448</td>
<td>361</td>
</tr>
<tr>
<td>CS-122</td>
<td>90×90</td>
<td>9</td>
<td>11</td>
<td>14</td>
<td>488</td>
<td>422</td>
</tr>
<tr>
<td>CS-136</td>
<td>100×100</td>
<td>9</td>
<td>11</td>
<td>14</td>
<td>544</td>
<td>515</td>
</tr>
<tr>
<td>CS-148</td>
<td>110×110</td>
<td>10</td>
<td>11</td>
<td>14</td>
<td>592</td>
<td>662</td>
</tr>
<tr>
<td>CS-150</td>
<td>120×120</td>
<td>10</td>
<td>11</td>
<td>14</td>
<td>600</td>
<td>679</td>
</tr>
<tr>
<td>CS-156</td>
<td>130×130</td>
<td>10</td>
<td>11</td>
<td>14</td>
<td>624</td>
<td>731</td>
</tr>
<tr>
<td>CS-164</td>
<td>140×140</td>
<td>10</td>
<td>11</td>
<td>14</td>
<td>656</td>
<td>802</td>
</tr>
<tr>
<td>CS-174</td>
<td>150×150</td>
<td>11</td>
<td>11</td>
<td>14</td>
<td>696</td>
<td>978</td>
</tr>
<tr>
<td>CS-184</td>
<td>160×160</td>
<td>11</td>
<td>11</td>
<td>14</td>
<td>736</td>
<td>1087</td>
</tr>
<tr>
<td>CS-192</td>
<td>170×170</td>
<td>11</td>
<td>11</td>
<td>14</td>
<td>768</td>
<td>1178</td>
</tr>
<tr>
<td>CS-198A</td>
<td>180×180</td>
<td>11</td>
<td>11</td>
<td>14</td>
<td>792</td>
<td>1248</td>
</tr>
<tr>
<td>CS-198B</td>
<td>190×190</td>
<td>11</td>
<td>11</td>
<td>14</td>
<td>792</td>
<td>1248</td>
</tr>
<tr>
<td>CS-198C</td>
<td>200×200</td>
<td>11</td>
<td>11</td>
<td>14</td>
<td>792</td>
<td>1248</td>
</tr>
</tbody>
</table>

III. Orbit Determination

This paper characterizes filter efficiency improvements for both the extended and unscented Kalman filters when using the cubed-sphere model for evaluation of the gravity field. The following sections describe these filters, their configurations, and incorporation of the cubed-sphere model in the filter force model.

This study only considers a selection of simulation-based tests, which likely provides a best-case assessment of computational improvements. The amount of computation time reduction possible with the cubed-sphere model depends on the complexity of the OD system. In the case of precise
orbit determination (POD) systems, increased measurement and dynamical model complexity reduces the gravity model calculation’s percentage of the total computation time. This reduces the benefit of the cubed-sphere for those cases. These tests do not include temporal variations in the gravity field, but a brief discussion on accounting for such perturbations in the cubed-sphere model may be found in [17].

A. Filter Descriptions

The extended and unscented Kalman filters consist of two distinct processes: the time and measurement updates. The time update uses knowledge of the satellite dynamics to propagate the $n$ elements of the filter estimated state $\hat{X}$ and the $n \times n$ state-error variance-covariance matrix $P$ forward in time between observations. The measurement update applies corrections to the estimated state and uncertainty based on provided measurements. From a computation time perspective, including the cubed-sphere model solely influences the time update. Hence, this section briefly describes this operation to provide background for future discussion.

When performing a time update in the extended Kalman filter (EKF), the selected integration algorithm propagates the reference trajectory and a state transition matrix (STM) $\Phi(t_i, t_{i-1})$ through time. The force model may be written as

$$\dot{X} = F(X, t),$$

with $X \equiv X_0$ at $t_o$. This allows for numerically integrating the STM using the differential equation

$$\dot{\Phi}(t_i, t_{i-1}) = A(t_i)\Phi(t_i, t_{i-1})$$

where $\Phi(t_{i-1}, t_{i-1}) = I$ and

$$A(t) = \left[\frac{\partial F(t)}{\partial X(t)}\right]^*.$$  

The right side of Eq. 4 is simply the Jacobian of the force model with respect to the estimated state vector, and evaluated using a reference trajectory $X^*$. The STM provides a first-order mapping of the state-error covariance matrix $P_i$ via

$$P_i = \Phi(t_i, t_{i-1})P_{i-1}\Phi^T(t_i, t_{i-1}) + Q$$
where $Q$ is a process noise matrix that accounts for mapping errors due to incorrect force modeling.

To map $P_{i-1}$, the EKF requires both a force model and its Jacobian. A full description of the EKF may be found in [23].

To prevent the need for linearization, the unscented Kalman filter (UKF) propagates a probability density function (PDF) forward in time instead of an estimated state and its covariance matrix. This process, called the unscented transformation, begins with a priori knowledge of the filter state $\hat{X}_{i-1}$ and covariance knowledge $P_{i-1}$. After selecting a value for the parameter $\alpha$ such that $10^{-4} \leq \alpha \leq 1$, the weights are

\begin{align*}
W^m_0 &= \frac{\lambda}{n+\lambda} \quad (6) \\
W^c_0 &= W^m_0 + (3-\alpha^2) \quad (7) \\
W^m_j &= W^c_j = \frac{1}{2(n+\lambda)}, \quad j = 1, \ldots, 2n \quad (8)
\end{align*}

where

\begin{equation}
\lambda = 3\alpha^2 - n. \quad (9)
\end{equation}

The scale factor $\alpha$ determines the distance of the $\sigma$-points from the mean. This research uses $\alpha = 1.0$, which yields $3\sigma$ distances from the estimated state.

The UKF time update uses the unscented transformation and a collection of $\sigma$-points to propagate a PDF. Starting with $\hat{X}_{i-1}$ and $P_{i-1}$ at $t_{i-1}$, the $\sigma$-points are

\begin{align*}
\mathbf{x}_{0,i-1} &= \hat{X}_{i-1}, \quad (10) \\
\mathbf{x}_{j,i-1} &= \hat{X}_{i-1} + \left(\sqrt{n+\lambda}\right) L_j, \quad (11) \\
\mathbf{x}_{j+n,i-1} &= \hat{X}_{i-1} - \left(\sqrt{n+\lambda}\right) L_j, \quad (12)
\end{align*}

where $j = 1, \ldots, n$, and $L_j$ is the $j$-th column of the Cholesky decomposition $L$ of $P_{i-1}$, i.e. $P_{i-1} = LL^T$. Each point is then propagated forward using Eq. 2 and initial condition $\mathbf{x}_{j,i-1}$ for $j = 0, \ldots, 2n$, which generates the $\sigma$-points $\mathbf{x}_{j,i}$ at $t_i$. Finally, the propagated $\sigma$-points combine to
generate a time updated state and covariance matrix

$$\bar{X}_i = \sum_{j=0}^{2n} W^m_j X_{j,i}$$  \hspace{1cm} (13)$$

$$\bar{P}_i = Q + \sum_{j=0}^{2n} W^e_j (X_{j,i} - \bar{X}_i) (X_{j,i} - \bar{X}_i)^T$$  \hspace{1cm} (14)$$

where $Q$ is a process noise matrix. More information on the UKF may be found in [24].

B. Force Model Description

This section defines the filter modeled forces affecting satellite dynamics. This includes a discussion of the steps required to add the cubed-sphere gravity model to both of the filters. This research uses the TurboProp orbit integration package [25] for orbit propagation.

Using the EKF requires the evaluation of the gravity acceleration’s Jacobian with respect to the satellite position; a capability now added to the cubed-sphere model. Recall that the EKF maps the covariance matrix $P_{-1}$ using the STM and Eq. 5. As implied by Eqs. 3 and 4, generation of $\Phi(t_i,t_{i-1})$ requires the Jacobian of the gravity acceleration

$$\frac{\partial F(t)}{\partial X(t)} = \frac{\partial a_{\text{grav}}}{\partial r} = \frac{\partial^2 U}{\partial x^2} = \begin{bmatrix}
\frac{\partial^2 U}{\partial x^2} & \frac{\partial U^2}{\partial x \partial y} & \frac{\partial U^2}{\partial x \partial z} \\
\frac{\partial U^2}{\partial x \partial y} & \frac{\partial^2 U}{\partial y^2} & \frac{\partial U^2}{\partial y \partial z} \\
\frac{\partial U^2}{\partial x \partial z} & \frac{\partial U^2}{\partial y \partial z} & \frac{\partial^2 U}{\partial z^2}
\end{bmatrix}$$  \hspace{1cm} (15)$$

where $\mathbf{r}$ and $\mathbf{a}_{\text{grav}}$ are the instantaneous position and gravity acceleration, respectively. Since this Jacobian matrix is symmetric, Eq. 15 only requires the evaluation of six terms: $\partial^2 U/\partial x^2$, $\partial^2 U/\partial y^2$, $\partial^2 U/\partial z^2$, $\partial^2 U/\partial x \partial y$, $\partial^2 U/\partial x \partial z$, and $\partial^2 U/\partial y \partial z$. The cubed-sphere model represents these terms as submodels, which requires additional memory since $p$ in Eq. 1 increases by 6. One of the three terms along the diagonal may be computed by assuming the terms obey Laplace’s equation. However, using such a method accumulates the sum of errors in two terms and adds it to the third. With this representation of the Jacobian, the cubed-sphere model may be used in the EKF.

Cubed-sphere models representing the terms of the Jacobian are generated using the parameters provided in Table 1. This process uses the analytic solution to Eq. 15 for the Cartesian model as the base model. It is noted that Eq. 5-10 of [20] contains an erroneous minus sign in the equation.
for the value of $S$. Using the same model configuration for representing the gravity acceleration and its Jacobian yields a faster evaluation, i.e. reducing the number of duplicate evaluations of the interpolants. The models must be reconfigured if a Jacobian accurate to 14 digits is required, but such accuracy is not necessary. As demonstrated later in Section IV, this approximation meets the accuracy requirements for generating a STM.

Since the value of $p$ increases from 4 to 10 in Eq. 1, the total size of the cubed-sphere model increases by a factor of 2.5. At first glance, such a large file size may appear to limit its use. However, each shell in the cubed-sphere requires no information on the others, and contains information at a discrete altitude. Thus, the software must only load portions of the cubed-sphere model applicable to the satellite based on orbit altitude and the degree of the Chebyshev interpolation. Such an implementation reduces the runtime memory requirements from gigabytes to hundreds of megabytes.

Using a collection of submodels approximates the variational equations instead of solving for the derivatives of the interpolants. Taking a partial an interpolation model like the cubed sphere does not guarantee behavior representative of the true derivatives. Also, such numerical differentiation inevitably looses accuracy and adds to the computational cost. If the partials do not reasonably model the true variations, then the mapping of a deviation vector using the STM is unpredictable and the filter may diverge. Accuracy of such a method would only be guaranteed if the generation of the B-spline coefficients took the variational equations into account, e.g. a Hermite-like interpolation scheme. That is not the case here, and, for models of this size and precision, model generation would likely be a computationally intense problem. To minimize these issues while still providing a fast evaluation of the Jacobian, the submodel approach is used. Straightforward tabulation of the derivatives does not have these issues and the required additional memory is well within memory of a typical laptop. More sophisticated approaches to evaluating the variational equations are being considered in further development.

The UKF does not require the evaluation of any Jacobians, thus no changes to the cubed-sphere model are required for this filter. The integration software simultaneously propagates all of the $\sigma$-points, but this is simply a choice of implementation. The UKF may propagate each point via separate executions of the integration software.
As mentioned previously, other filters models limit the coverall computation speedup when using the cubed-sphere model. To make this assessment of orbit determination speed improvements more realistic, this research includes the NRLMSISE-00 [26] model of atmospheric density in the filter’s force model. With the atmospheric density computed using this model, the filter calculates drag with the usual formulation of the drag acceleration, e.g. see Eq. 8-28 of [18]. Drag due to interactions with the residual atmosphere comprises the second largest perturbing force for low-altitude satellites [18]. Since the filter models drag effects, the coefficient of drag $C_D$ is added to the estimated state vector. Thus, the $7 \times 1$ filter estimated state vector is

$$X = \begin{bmatrix} \mathbf{r} \\ \mathbf{v} \\ C_d \end{bmatrix}$$  (16)

where $\mathbf{v}$ is the satellite velocity vector.

When calculating the Jacobian of the drag acceleration, $\frac{\partial \rho(t, \mathbf{r})}{\partial \mathbf{r}}$ should be evaluated. Unfortunately, complex density models like the NRLMSISE-00 make such an evaluation burdensome. Numerical differentiation provides an alternative, but such a method must account for discontinuities and other singularities in such evaluations [19]. Variations accounting for spatial changes in atmospheric density tend to be vary small, especially when compared to the contributions of other terms. This research sets this partial equal to zero since the additional computation time outweighs the benefits of including the term. Other terms of the drag acceleration partial derivatives are included in the propagation of the STM.

Including filter process noise in both the EKF and UKF compensates for force modeling errors. This requires a correction term $Q$ (see Eq. 5) to the time update of the state covariance matrix [23]. There are multiple methods for varying this correction term depending on modeling errors and
where $\Delta t$ is the time between measurements and $\sigma_{\hat{X}}$, $\sigma_{\hat{Y}}$, and $\sigma_{\hat{Z}}$ represent the standard deviation of the unknown acceleration in each component direction. Derived from Eqs. 4.9.47 and 4.9.50 of [23], Eq. 17 assumes a constant acceleration in each component direction over a small $\Delta t$. In this implementation, process noise is not added for large $\Delta t$ since this violates the constant acceleration assumption. Filter tuning refers to the selection of the $\sigma$ values, which are largely unknown, to produce a more accurate estimate. Setting $\sigma_Q = \sigma_{\hat{X}} = \sigma_{\hat{Y}} = \sigma_{\hat{Z}}$, a value is selected such that the filter yields a realistic covariance matrix, i.e. approximately 97.1% of all estimated state errors are within the 3-$\sigma$ filter covariance ellipsoid. The value selected for a given scenario remains constant for all measurement times. We provide the values for $\sigma_Q$ for each scenario in Fig. 2.

C. Filter Test Description

Scenarios based on satellites at three different orbit altitudes are considered for this study: the Ocean Surface Topography Mission (OSTM)/Jason-2 [27], the Gravity Recovery and Climate Experiment (GRACE) [28], and a GPS Satellite [29]. The simulated GRACE satellite orbits at an altitude of approximately 485 km, while the Jason-2 satellite has a higher orbit at $\sim$1,336 km. Orbit determination systems for these two satellites require a fairly high degree model [5, 30]. In the case of GRACE, the orbit altitude is low enough to generate the 160×160 GGM02S model, and aid in estimating the gravity field to degree 200 when combined with terrestrial data [28]. GPS
saturates orbit at roughly 20,500 km. Hence, high-degree models are not required for estimation. However, this satellite scenario is included since current efforts seek to generate near real-time orbit determination solutions for the GPS constellation [4], and, as seen in Section IV, it also provides an extreme case with many observations. A combination of ground- and GPS-based observations are used, with further details provided in the next section. To prevent confusion with GPS-based observations of the Jason-2 and GRACE satellites, the satellite in the final scenario is referred to as a semi-synchronous, or SemiSync, satellite. A semi-synchronous satellite has a 12 hour period.

Table 2 Initial conditions for the simulated Jason-2, GRACE, and SemiSync satellites

<table>
<thead>
<tr>
<th>Element</th>
<th>Jason-2</th>
<th>GRACE</th>
<th>SemiSync</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semimajor Axis (km)</td>
<td>7714.1363</td>
<td>6863.1363</td>
<td>26559.8</td>
</tr>
<tr>
<td>Eccentricity</td>
<td>0.01</td>
<td>0.0</td>
<td>0.01</td>
</tr>
<tr>
<td>Inclination (deg)</td>
<td>66.0</td>
<td>89.0</td>
<td>55.0</td>
</tr>
<tr>
<td>Right Ascension of Ascending Node (deg)</td>
<td>61.662</td>
<td>80.690</td>
<td>272.85</td>
</tr>
<tr>
<td>Argument of Perigee (deg)</td>
<td>149.654</td>
<td>357.069</td>
<td>0.0</td>
</tr>
<tr>
<td>True Anomaly (deg)</td>
<td>178.548</td>
<td>15.869</td>
<td>11.68</td>
</tr>
</tbody>
</table>
True orbits for each satellite are generated using accurate gravity and atmospheric density models. Generation of the true trajectory uses the full \(200 \times 200\) GGM02C [31] spherical harmonic gravity model. The Jacchia-Bowman 2008 (JB2008) [32] model provides the most realistic representation of atmospheric density currently available. In terms of area used in the drag calculation, the satellite is modeled as a simple cannonball. The TurboProp RK5(4) software [25] provides the numerical integration capabilities for generation of the true trajectories and within the filters themselves. The simulation epoch time is July 14, 2000 00:00:00 UTC. This date corresponds to a solar maximum, and, hence, a higher drag perturbation. Using these force models and the initial states provided in Table 2, trajectories are propagated forward 24 hours to a final time of July 15, 2000 00:00:00 UTC.

**Table 3 Summary of filter error sources.** Truth + \(\sigma\) means true value plus zero mean, 1-\(\sigma\) Gaussian noise. Sigma values are provided where appropriate.

<table>
<thead>
<tr>
<th>Element</th>
<th>Truth Value</th>
<th>Filter Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial Position</td>
<td>True Position</td>
<td>Truth + 1-(\sigma), (\sigma = 0.01) km</td>
</tr>
<tr>
<td>Initial Velocity</td>
<td>True Velocity</td>
<td>Truth + 1-(\sigma), (\sigma = 0.001) km/s</td>
</tr>
<tr>
<td>Initial (C_D)</td>
<td>2.3</td>
<td>2.0</td>
</tr>
<tr>
<td>Gravity Estimate</td>
<td>GGM02C</td>
<td>1-(\sigma) clone of GGM02C</td>
</tr>
<tr>
<td>Gravity Model Degree</td>
<td>200 (\times) 200</td>
<td>Test dependent</td>
</tr>
<tr>
<td>Atmospheric Density Model</td>
<td>JB2008</td>
<td>NRLMSISE-00</td>
</tr>
<tr>
<td>GPS Satellite</td>
<td>True Position</td>
<td>Truth + periodic error ((\sigma = 2.8) cm)</td>
</tr>
<tr>
<td>Ground Stations</td>
<td>True Position</td>
<td>Truth + 1-(\sigma), (\sigma = 2) cm</td>
</tr>
</tbody>
</table>

Table 3 describes the a priori and modeling errors included in these scenarios. These tests assume an accurate a priori solution, which primarily affects filter settling time and not overall accuracy. The NRLMSISE-00 atmospheric density model does not incorporate the latest advancements in space weather modeling; hence its reduced accuracy when compared to the JB2008 model. Gravity clones model errors in the gravity field, with more details on this technique provided below. Tests assess computational improvement as a function of degree by testing each of the models in Table 1. The filter measurement models also include errors in the ground station locations and GPS ephemeris.
The filter’s knowledge of ground station locations include Gaussian errors with zero mean and a 2 cm standard deviation. The filter modeled GPS satellite position includes a 2.8 cm 1-σ error in each component direction. This corresponds to the estimated 5 cm RSS position error of the IGS ultra-rapid ephemeris [33].

Gravity clones allow for the incorporation of statistical uncertainty, i.e. the gravity model estimate covariance matrix, in Monte-Carlo-like tests. Given

\[
P_{\text{grav}} = LL^T, \tag{18}
\]

where \( L \) is computed via the Cholesky decomposition, one may generate a gravity clone using

\[
c_{\text{clone}} = c_o + L^T e \tag{19}
\]

where \( c_o \) is a vector of the Stokes coefficients describing the spherical harmonics gravity model, \( P_{\text{grav}} \) is the full estimation error covariance matrix of these coefficients, and \( e \) is a vector of random numbers with zero mean and unit variance. See, for example, [34] for more information on generating a vector that includes random deviations from the mean when given a multinormal distribution.

Five 1-σ gravity clones are generated using the publicly available portion of the GGM02C spherical harmonic model covariance matrix. The University of Texas Center for Space Research (CSR) only releases the diagonal terms of the GGM02C covariance matrix, but indicates that correlations are small.\(^1\) Thus, given this truncated \( P_{\text{grav}} \), true clones are not created. Without correlation information, the clones used in this study likely represent a conservative representation of the gravity errors. The corresponding cubed-sphere representations of these clone models are computed using the configurations of Table 1. Each clone is used as the filter-modeled gravity field, and filter-estimated state errors are averaged to generate a value incorporating variations in the gravity field within the uncertainty of the gravity field model.

D. Measurements

For these filter tests, three measurement types are considered: Earth-based range observations, Earth-based range-rate, and satellite-based range. These types approximate observations from the

---

\(^1\) Personal communication with John Ries of CSR.
Satellite Laser Ranging (SLR) [35], Doppler Orbitography and Radiopositioning Integrated by Satellite (DORIS) [36], and Global Positioning System (GPS) services, respectively. Locations of the SLR\textsuperscript{2} and DORIS\textsuperscript{3} ground stations are based on the actual network. Simulation of the GPS satellite constellation propagates 24 satellites forward in time, with initial conditions described by the optimal satellite design in [37]. Using the true satellite trajectory, simulated observations relative to the measurement reference point are generated with Gaussian noise added. For the GRACE and Jason-2 scenarios, available measurements are provided every 10 s. The measurement interval for the SemiSync scenario is increased to every 60 s. More accurate measurement types are available, such as GPS carrier phase, but they often require additional processing. Like other high-accuracy POD methods, this additional processing will reduce the overall computation savings when using the cubed-sphere model.

The measurement simulation process used for these simulations deviates from operational methods in several ways. Operationally, if multiple satellites are visible, a ground station must decide which one to track. This study assumes no such resource conflicts exist, and that a ground station will always track the satellite while visible. Also, civilian tracking of the GPS satellites is performed via pseudorange measurements gathered on the ground by the International GNSS Service (IGS) ground stations [33]. These tests emulate this capability through SLR measurements instead of modeling the complete IGS network.

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Gaussian Measurement Noise</th>
<th>Filter Observation σ</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPS Pseudorange</td>
<td>1 m</td>
<td>1.002 m</td>
</tr>
<tr>
<td>SLR/Range</td>
<td>5 mm</td>
<td>4 cm</td>
</tr>
<tr>
<td>DORIS/Range-Rate</td>
<td>2 mm/s</td>
<td>1 cm/s</td>
</tr>
</tbody>
</table>

Table 4 describes the simulated observation properties. It is noted that the measurement noise
for the GPS pseudorange is over twice that of the Jason-2 orbit determination estimate in [5]. Measurement reference locations are not estimated; thus, the filter must compensate for the station location error as a measurement error. The filter observation $\sigma$ values are selected to yield a normal distribution of prefit-residual errors in filter processing. Thus, the filter observation $\sigma$ value also accounts for errors in the locations of the GPS satellites and ground stations. In real-world processing, there would be a larger distinction between the white noise value and the required filter observation to accounts for all systematic error sources.

E. Software

The software used for this study uses a combination of already developed software and new tools specific to this study. Orbit propagation uses the TurboProp package [25], which is written in C and compiled with the Intel C compiler version 11.0.081. The gravity models and computationally intensive filter operations are written in Fortran 90/95 and compiled with the Intel Fortran compiler version 11.0.081. All software compilation used -O3 optimization. Portions of the EKF and UKF software not computationally intensive use Python 2.6.5. Speed comparisons are performed on a dedicated RedHat Enterprise 5 Linux machine with a dual Intel Xeon 2.33 GHz processor and 8 GB of memory.

Great care must also be taken when implementing the gravity models since differences in their computation time have the largest influence on the timing tests of Section IV. The spherical harmonic model uses precomputation of constants and vectorized programming wherever possible. See, for example, discussions in [20] and [38]. Development of the cubed-sphere model uses interpolation table implementations of the Chebyshev and B-spline interpolants to speedup their evaluation. Both models contain additional optimizations to minimize memory swapping between cache and RAM.

IV. Test Results

A. Propagation of the State Transition Matrix

This section verifies that the cubed-sphere model adequately approximates the Jacobian to generate a STM. Test orbits in this section employ the same propagation methods and conditions as [17]. Using the RK7(8) integrator in TurboProp, a variety of initial conditions are generated and
propagated with position, velocity, and the STM output every 20 seconds over a 24 hour span. The initial orbit altitudes span 200 to 1,000 km at 50 km intervals. The right ascension of the ascending node ranges from 0° to 180° in 5° increments, while the inclination varies from 0° to 90° in 2.5° intervals. All other orbit elements are initially set to zero. Thus, for each altitude, approximately 1,300 orbits are tested. The Greenwich sidereal time is set to 0° at the epoch time, with an Earth rotation rate of 360° per solar day. The planetary radius and gravitation parameter are set to the appropriate values as determined by the GGM02C model. Each set of initial conditions is propagated using the cubed sphere model and the Cartesian representation of the spherical harmonic model. These tests consider the CS-34, CS-100, CS-174, and CS-198C cubed-sphere models from Table 1.

To compare the STM generated using the different gravity models, the number of common digits

\[ CD = \log_{10} \left| \frac{\beta_t}{\beta_e - \beta_t} \right| \]  

(20)
is used where \( \beta_t \) and \( \beta_e \) are the defined true and test values, respectively. For example, \( \beta_t \) and \( \beta_e \) can correspond to the first row, first column of the spherical-harmonic- and cubed-sphere-determined STM matrices, respectively, and Eq. 20 determines the number of common digits for the values. In these tests, \( \Phi(t, t_o) \) is a 6×6 matrix containing values of multiple magnitudes and different units. Given these characteristics, precision is measured by Eq. 20 instead of an absolute value, e.g. m/s². Additionally, instead of providing 36 values of \( CD \) corresponding to each element of the matrix, all 36 values are averaged at all times for a given altitude.

Figure 3 illustrates the minimum and mean precision, with 1-σ error bars, for the STMs generated using the different gravity representations via the integration tests. As mentioned previously, these values are generated based on integrated states at all times for all orbits with the same initial altitude. Results demonstrate 12 to 13 digits of precision for altitudes at or above 300 km, with standard deviations of approximately 1 digit. Minimum values appear small, however values with such a low precision correspond to those with a very small magnitude. Thus, the absolute error is small and has little impact on the mapping of \( P_{t-1} \).

Figure 4 provides the 3D RMS position and velocity errors for these integration tests over all points in time for all orbits over the 24 hour span. Like the results of [17], position errors remain at sub-millimeter magnitudes, and decrease slightly with altitude. This extends conclusions from [17]
Fig. 3 Precision, in \( CD \), of the STM computed using various cubed-sphere gravity models when compared to those computed via the spherical harmonic base model.

to models of degree 200, i.e. the cubed-sphere model is capable of representing a \( 200 \times 200 \) spherical harmonic model with little or no difference in propagated trajectories.

These \( \Phi(t_i, t_0) \) results affect a batch filter more than a sequential filter. In a sequential algorithm like the EKF, one integrates from \( t_{i-1} \) to \( t_i \). In such a process, the initial condition \( \Phi(t_{i-1}, t_{i-1}) = I \) is used with each execution of the integration algorithm. Thus, accumulated differences in \( \Phi(t_i, t_{i-1}) \) when computed via the cubed-sphere and spherical harmonic model are removed with each filter time update.

B. Filter Execution Time Considerations

This section describes the influence of the time and measurement updates on the execution time of the EKF and UKF. These results help to provide a better understanding of the filter speedup results in the next section.

Figure 5 illustrates the integration speedup, both with and without atmospheric drag, when
Fig. 4 Summary of orbit propagation 3D RMS state differences for various cubed-sphere gravity models when compared to trajectories computed via the spherical harmonic model.

using the cubed-sphere model instead of the spherical harmonic model. An initial condition is propagated using each of the cubed-sphere and spherical harmonic models, with the ratio of the execution times computed to generate a speedup factor. This test also uses the TurboProp RK5(4) of [25], and propagate for 24 hours. For each gravity model degree, the provided speedup comprises an average of 30 tests. In the single-state results, the position and velocity of a single satellite is integrated. The UKF-like integration simultaneously propagates 13 satellite state vectors, and reflects the integration speedup for this implementation of the cubed-sphere model in the UKF. The third plot describes the speedup when integrating one satellite state vector and the STM. The matrix operations in this third test increase the fixed cost of evaluating the force model; thus, the speedup factor decreases. Atmospheric drag tests used the NRLMSISE-00 atmospheric density model, which decreases the percentage of the overall computation cost associated with the gravity model. As a result, integration speedup factors decrease. Speedup factors for the single-state and UKF-like test are nearly identical, which is expected. Small deviations occur due to noise in the runtime measurements and slight variations in runtime.

To gain some analytic insight into the effect of reduced time update computation time on orbit determination efficiency, an approximation of the filter speedup $S_F$ based on the measurement
Fig. 5 Integration speedup factors when using the cubed-sphere gravity model. Solid lines only include gravity in the force model, while the dashed lines include atmospheric drag.

The update evaluation time $T_M$ and orbit propagation speedup factors $S_I$ is derived. Upon separating the filter execution time into a sum of the time and measurement updates, the filter speedup factor is

$$S_F = \frac{T_{SH} + T_M}{T_{CS} + T_M},$$

(21)

where $T_{SH}$ and $T_{CS}$ are the evaluation times of the time update using the spherical harmonic model and the cubed-sphere model, respectively. Eq. 21 assumes that the measurement update time does not vary between the cubed-sphere and spherical harmonic model tests. Upon representing time in units of $T_{CS}$,

$$S_F \approx S_I + \frac{\tilde{T}_M}{1 + \tilde{T}_M}$$

(22)

where

$$\frac{T_{SH}}{T_{CS}} \approx S_I,$$

(23)

and $\tilde{T}_M = T_M/T_{CS}$. Although integration dominates the execution of the time update, other operations add some computation cost. Hence, Eq. 23 is an approximation. This now provides a representation of the filter speedup as a function of the measurement update evaluation time,
normalized by $T_{CS}$, and the integration speedup. Assuming $S_I \geq 1$, then $S_F$ is in the range $[1, S_I]$ and varies with $\tilde{T}_M$.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Average Data Gap (s)</th>
<th>Average Number of Observations</th>
<th>Mean EKF $\tilde{T}_M$</th>
<th>Mean UKF $\tilde{T}_M$</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRACE GPS</td>
<td>10</td>
<td>11.3</td>
<td>2.06</td>
<td>1.55</td>
</tr>
<tr>
<td>Jason-2 GPS</td>
<td>10</td>
<td>11.0</td>
<td>2.01</td>
<td>1.55</td>
</tr>
<tr>
<td>Jason-2 SLR</td>
<td>17.5</td>
<td>4.9</td>
<td>0.93</td>
<td>0.49</td>
</tr>
<tr>
<td>Jason-2 SLR+DORIS</td>
<td>10.5</td>
<td>6.4</td>
<td>1.51</td>
<td>0.87</td>
</tr>
<tr>
<td>SemiSync SLR</td>
<td>60</td>
<td>17.5</td>
<td>3.05</td>
<td>2.25</td>
</tr>
</tbody>
</table>

Table 5 includes statistics on the average time between observations, the average number of observations per measurement update, and $\tilde{T}_M$ for both the EKF and UKF for the five scenarios. Simulations using SLR measurements have more observations available at a given epoch than expected because tasking of the network is not modeled in this study, i.e. all visible stations observe the satellite. Also, some geographical areas, such as Japan, Europe, and regions of the United States of America, have a relatively dense concentration of networks that raise this average during select passes. For the Jason-2 SLR scenario, the relatively small $\tilde{T}_M$ values imply that it will yield the largest filter speedup factors. More observations are processed in the SemiSync SLR scenario than all others due to its high altitude, i.e. it is visible to many ground stations for long periods of time. For this reason, the SemiSync scenario has the largest measurement update evaluation time. $\tilde{T}_M$ for the EKF increases, when compared to the UKF, because the time required for integration decreases with fewer states.

### C. Filter Execution Time Results

This section describes changes in filter computation time when using the cubed-sphere model instead of the spherical harmonic model. This includes a presentation of the speedup for the scenarios described, as well as the effects of a slight variation in the data rate. Of course, these results will vary with filter implementation and other factors affecting execution time, but demonstrate the
potential benefits of using the cubed-sphere model.

Figure 6 illustrates the speedup factors when using the cubed-sphere model in the EKF and UKF. As implied by Table 5 and Eq. 22, speedup factors vary with the filter and scenario. As a result of the relatively low $\tilde{T}_M$, processing of SLR measurements in the Jason-2 scenario exhibits the largest benefit when using the cubed-sphere model. Filter speedup does not reach the levels of the orbit propagation speedup factors.

Fig. 6 Filter speedup factors for the UKF (left) and EKF (right) simulations.

Fig. 7 Change in filter execution times for the UKF (left) and EKF (right) simulation relative to the $20 \times 20$ scenario.

Figure 7 provides the filter execution time normalized by the execution time of the $20 \times 20$ model, which illustrates the change in filter execution time with increasing gravity model degree. When using the cubed-sphere model, filter computation time slightly increases with base model degree
due to the corresponding increase in Chebyshev interpolation. Execution time for filters using the cubed-sphere-model increase by a maximum factor of 1.14, whereas the spherical-harmonic-based filter time increases by as much as a factor of 15 for the UKF. Thus, assuming an increase in memory requirements is acceptable, a filter may use a higher-degree gravity model with little additional computation cost. Such increases are smaller than hundreds of MB.

Fig. 8 Speedup factors for the Jason-2 satellite using GPS observations provided at different rates in the UKF (left) and EKF (right).

For Jason-2 precise orbit determination, the NASA Jet Propulsion Laboratory processes GPS normal points every 5 minutes, but may change to every 30 s.\textsuperscript{4} For Fig. 8, the GPS observations are processed at these reduced frequencies, and the improved speedup factors for the Jason-2 satellite are illustrated. With these larger time gaps between observations, $T_{CS}$ increases and the filter speedup factor improves. JPL uses a square-root information filter for precise orbit determination [39], which is closer to the EKF than the UKF. For this test, the EKF speedup increases by almost a factor of 5 for the 5 min scenario, and 2.5 for the 30 s test.

Figure 9 illustrates the ratio of the execution times for the UKF with the cubed-sphere model and the EKF with the spherical harmonic model. Although the UKF includes higher-order dynamics in the propagation of the state-error covariance matrix, the additional computation cost limits its use. Assuming a system already requires orbit determination with a high-degree spherical harmonic model, this test examines the cost of switching to an UKF with the cubed-sphere model. The figure

\textsuperscript{4} Personal communication with Shailen Desai of the NASA Jet Propulsion Laboratory
indicates that such a switch may be possible, with little or no additional computation cost, for systems using high-degree models and larger data gaps. In the case of the Jason-2 satellite with SLR observations and a gravity model of degree 180 or higher, using an UKF with the cubed-sphere model provides faster orbit determination than the EKF with the spherical harmonic model. In the other scenarios, orbit determination with a $200 \times 200$ cubed-sphere model in the UKF requires a maximum of 50% more computational time than using the spherical harmonic model is the EKF.

![Fig. 9 Execution time of the UKF with the cubed-sphere model, when normalized by the execution time for the EKF with the spherical harmonic model.](image)

**D. Filter Precision**

This section discusses the differences between filter solutions generated with the cubed-sphere model and those computed via the spherical harmonic model. First, a short treatment of absolute accuracy for the GRACE and Jason-2 scenarios is provided, but only to provide a sense of scale for later results. A discussion of changes in the filter estimated state when using different gravity formulations then follows. Filter precision results for the SemiSync satellite are not provided. Given the large altitude, any differences in filter states are at or within machine precision noise.

Figure 10 illustrates the absolute filter accuracy for the GRACE and Jason-2 scenarios, and demonstrates that these orbit determination schemes are accurate to centimeters. Errors decrease with increased gravity model degree, with GRACE and Jason-2 errors decreasing until degree 80
and 50, respectively. This behavior would change with measurement accuracy, i.e. more accurate measurements yield filter solutions that are more dependent on gravity model truncation.

Fig. 10 3D RMS filter performance for the GRACE (left) and Jason-2 (right) scenarios. Solid and dashed lines correspond to UKF and EKF results, respectively.

Figure 11 provides the 3D RMS difference between state results for the EKF using the cubed-sphere and the spherical harmonic models. These 3D RMS values are calculated using all differences over all gravity clones for a given gravity model degree. Relative to a filter accuracy of centimeters, these results demonstrate little deviation between the two filters. Smaller differences are seen for the GRACE and Jason-2 GPS scenarios. Due to the larger measurement errors when using GPS pseudorange, these scenarios use a larger $\sigma_Q$. The increased process noise matrix decreases the contribution of the system dynamics in the filter estimated state, and, thus, reduces differences caused by the two gravity models. Process noise may be used to reduce differences between the cubed-sphere-model- and spherical-harmonic-model-based solutions, but such a procedure would be unnecessary given the filter accuracy.

Figure 11 also provides the RMS differences of the diagonal elements of the covariance matrix, i.e. the filter’s estimated-state variances. These deviations are similar to those of the estimated state, i.e. the variances in the SLR and DORIS scenarios are greater than the GPS scenarios, except in the case of velocity. In this latter case, differences are within 1-2 orders of magnitude.

Figure 12 illustrates the differences between the cubed-sphere- and spherical-harmonic-based solutions using the UKF. Typically, differences for the SLR and DORIS scenarios remain larger. The
differences between the cubed-sphere- and spherical-harmonic-based filters using GPS measurements increase by an order of magnitude when compared to the differences for the EKF.

As seen when comparing Figs. 11 and 12, state-error covariance matrix deviations for the UKF increase by as much as eight digits when compared to those of the EKF. Given maximum state-error covariance values on the order of $10^5$ or $10^6$ mm$^2$, the resulting UKF covariance matrices still agree to at least seven digits. This provides a sufficiently accurate state-error covariance matrix.

The larger deviations result from the difference in the time update in the UKF. The UKF includes second, and possibly third, order effects in the propagation of the covariance matrix [24]. The small differences in the gravity field perturb the low-order, low-magnitude second and third order contributions to this propagation. Thus, differences increase slightly when using the UKF.

V. Conclusions

In this research, the cubed-sphere model was integrated with both the extended and unscented Kalman filters, and changes in computational efficiency and the estimated state were characterized. When using cubed-sphere models designed for optimal precision with the base model, these results demonstrate that orbit determination with the cubed-sphere model yields improved computational efficiency with little change in the estimated trajectory. Specifically, the cubed-sphere model allows
for increasing the degree of the gravity model in the filter force model with little change in computational burden. Thus, orbit determination systems requiring a short latency may use high-degree gravity models for a relatively low computation cost, thereby reducing the effects of force modeling errors on such solutions.

The propagation accuracy of a cubed-sphere-model-based state transition matrix was demonstrated, along with the associated integration speedup. The trajectory propagation with the cubed-sphere model provides a highly-precise state transition matrix when compared to those computed via the spherical harmonic model. Additionally, this test demonstrated that cubed-sphere models representing a $200 \times 200$ spherical harmonic model also yield small differences in integrated states.

For precise orbit determination, or any navigation system requiring complex modeling, the nearly constant evaluation time of the cubed-sphere model for all degrees provides the principle benefit. This study demonstrates that the cubed-sphere model almost removes the dependence of computation time on gravity model degree, thereby removing the cost of increased accuracy when using the common spherical harmonic model. The next step in using the cubed-sphere model is to integrate the software with a third-party precise orbit determination system. Such a test provides a more comprehensive assessment of the model’s capabilities in a more complex system.
Precision requirements for generating the cubed-sphere gravity model may be reduced depending on the estimation and modeling accuracy for a given orbit determination system. This lowers the degree of the model’s interpolants, thus decreasing model evaluation time. Such a configuration would be tailored to the orbit determination requirements for a given mission, and provides an additional computation savings.

With the Jacobian of the acceleration vector now included, the cubed-sphere gravity model will provide faster orbit determination for systems using a batch filter, i.e. a batch least squares. Batch filters better lend themselves to orbit determination with sparse observations, meaning a majority of the computational burden may be due to the propagation of the reference trajectory. Reducing the computation time for batch processing would have implications on future space situational awareness systems that monitor as many as 100,000 space objects.

**Acknowledgments**

A majority of this research was funded by the NASA Graduate Student Researchers Program (NASA/GSRP) through the Johnson Space Center. Gregory Beylkin’s contribution was supported by AFOSR grant FA9550-07-1-0135.

**References**


