

# State Transition Matrix for Perturbed Orbital Motion Using Modified Chebyshev Picard Iteration

Julie L. Read<sup>1</sup> · Ahmad Bani Younes<sup>2</sup> ·  
Brent Macomber<sup>1</sup> · James Turner<sup>2</sup> ·  
John L. Junkins<sup>1</sup>

Published online: 13 August 2015

© The Author(s) 2015. This article is published with open access at Springerlink.com

**Abstract** The Modified Chebyshev Picard Iteration (MCPI) method has recently proven to be highly efficient for a given accuracy compared to several commonly adopted numerical integration methods, as a means to solve for perturbed orbital motion. This method utilizes Picard iteration, which generates a sequence of path approximations, and Chebyshev Polynomials, which are orthogonal and also enable both efficient and accurate function approximation. The nodes consistent with discrete Chebyshev orthogonality are generated using cosine sampling; this strategy also reduces the Runge effect and as a consequence of orthogonality, there is no matrix inversion required to find the basis function coefficients. The MCPI algorithms considered herein are parallel-structured so that they are immediately well-suited for massively parallel implementation with additional speedup. MCPI has a wide range of applications beyond ephemeris propagation, including the propagation of the State Transition Matrix (STM) for perturbed two-body motion. A solution is achieved for a spherical harmonic series representation of earth gravity (EGM2008), although the methodology is suitable for application to any gravity model. Included in this representation the normalized, Associated Legendre Functions are given and verified numerically. Modifications of the classical algorithm techniques, such as rewriting the STM equations in a second-order cascade formulation, gives rise to additional speedup. Timing results for the baseline formulation and this second-order formulation are given.

---

✉ Julie L. Read  
julie.read@tamu.edu

<sup>1</sup> Aerospace Engineering, Texas A&M University, 3141 TAMU, College Station, TX 77843, USA

<sup>2</sup> Aerospace Engineering, Khalifa University, P.O. Box 127788, Abu Dhabi, United Arab Emirates

**Keywords** Modified Chebyshev-Picard Iteration (MCPI) · Orbital mechanics · Celestial mechanics · State transition matrix · Spherical harmonic gravity · Numerical integration · Trajectory propagation

## Introduction

The Modified Chebyshev-Picard Iteration (MCPI) method is used to solve both linear and nonlinear, high precision, long-term orbit propagation problems through iteratively finding an orthogonal function approximation for the entire state trajectory. At each iteration, MCPI finds an entire path integral solution (over a large finite interval, converges over intervals up to three orbits), as opposed to the conventional, incremental step-by-step solution strategy of more familiar numerical integration strategies, i.e., those based on explicit numerical methods. While very long path segments are convergent, computationally optimal path segments are typically a large fraction of an orbit. A major advantage of the MCPI approach is that the use of cosine sampling reduces the well-known Runge Effect, whereby the largest errors are encountered near the boundary of piecewise approximation segments [1]. This algorithm has recently proven to be a powerful tool for solving nonlinear differential equations for both initial value problems (IVP) and boundary value problems (BVP) [1–5]. Comparisons with traditional step-by-step methods [6, 7] have shown that MCPI is more efficient for a prescribed solution accuracy. Significantly, however, unlike conventional integration approaches, it is ideally suited for massive parallel implementations that provide further boosts in the computational performance. Algorithm tests are currently under development for massive parallel implementations, where the performance results will be presented in future papers.

Here we consider the MCPI method to compute the State Transition Matrix (STM) for the case of perturbed motion, which has applications in many areas including celestial mechanics and control systems. Propagation of the STM is useful in determining the sensitivity of the IVP solution to the initial conditions; for instance, the STM predicts how deviations from the desired initial conditions will cause the trajectory of a spacecraft to stray from a nominal path. The STM may be used to approximate the time evolution of a state vector even for highly nonlinear systems, such as the two-body problem perturbed by an arbitrary degree spherical harmonic gravity. In cases where the initial deviation is small at time  $t_0$ , a linear approximation may be used to determine the locally linear state deviation at time  $t$ . This linear approximation is generated using the STM, a matrix of partial derivatives for the instantaneous position and velocity with respect to the initial position and velocity; this means that the STM is initially equal to the identity matrix [8, 9]. These deviations can be approximated, to a problem-dependent accuracy, by using the unperturbed Keplerian STM; however, it is desirable to be able to include a prescribed number of the high order harmonics, and in general, other perturbations. The present paper provides an efficient algorithm to include these higher order gravitational perturbations, as desired/required in the STM.

The present computation of the STM for the spherical harmonic gravity model requires second partials of the gravity potential with respect to spherical geocentric coordinates: radius, latitude, and longitude. These required partial derivatives include second partials of the normalized, Associated Legendre Functions (ALFs), and these functions (used often in science and engineering applications) are related to the Legendre polynomials. A full description and derivation are given in Appendix B, and the equations that require the ALFs are given in the section “[State Transition Matrix Using Spherical Harmonic Gravity](#).” The normalized version of the ALFs is the preferred formulation for the associated recursion used to compute these functions. Without the normalization, the ALFs tend toward weak numerical instability as a higher degree and order gravity model is used. Once the normalized ALFs are computed, the gravity and associated derivatives are computed in part by introducing the appropriate scale factor to generate the unnormalized version of the ALFs. Since a spherical harmonic model is used and this gravity potential is a rigorous solution of the Laplace equation, the ALFs’ second partial expression for an arbitrary order spherical harmonic series is verified using the Laplace equation. Both the trajectory and the STM are computed in a rotating, Earth-centered, Earth-fixed frame, which is transformed into an Earth-centered inertial frame (for subsequent integration) at each iteration. The rotating ECEF frame is used to avoid the explicit time dependence of the potential and the associated Jacobi integral (which is also a constant Hamiltonian for the perturbed motion). Though enforcement of the differential equation is inherent in the present study, we also check to confirm that the STM accurately satisfies (with maximum relative errors of  $< 10^{-14}$  in a Matlab implementation and up to one orbit, for the most precise tuning) the theoretical STM group properties as well as the STM symplectic property associated with natural conservative dynamical systems [8]. The present development of the STM is also verified using finite difference methods.

## Modified Chebyshev Picard Iteration

MCPI is a fusion of Picard iteration with approximation via orthogonal Chebyshev polynomials. A few papers were published on the topic prior to the development of parallel computing capabilities [10–12], but it was not until recent years that this literature was significantly expanded by the research group (Junkins, et al.) at Texas A&M University. MCPI is an iterative, path approximation method for solving smoothly nonlinear systems of ordinary differential equations. The entire state trajectory over a long time arc is approximated at every iteration until a specified tolerance is met [2]. Emile Picard stated that, given an initial condition  $\mathbf{x}(t_0) = \mathbf{x}_0$ , any first order differential equation [1]

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{f}(t, \mathbf{x}(t)), \quad t \in [a, b] \quad (1)$$

with an integrable right hand side may be rearranged without approximation into an integral:

$$\mathbf{x}(t) = \mathbf{x}(t_0) + \int_{t_0}^t \mathbf{f}(\tau, \mathbf{x}(\tau))d\tau \tag{2}$$

Then, given a suitable starting approximation  $\mathbf{x}^0(t)$ , a unique solution to the initial value problem may be found using an iterative sequence of path approximations through Picard iteration as

$$\mathbf{x}^i(t) = \mathbf{x}(t_0) + \int_{t_0}^t \mathbf{f}(\tau, \mathbf{x}^{i-1}(\tau)) d\tau, \quad i = 1, 2, \dots \tag{3}$$

where the integrand of the Picard iteration sequence is approximated using Chebyshev polynomials. For further details on the basics of this novel integration technique and its convergence properties, refer to References [1] and [2]. Since the Chebyshev polynomials are orthogonal, a matrix inverse is avoided when finding basis function coefficients. Because cosine sampling is employed for MCPI, the Runge Effect seen at trajectory boundaries is significantly reduced. Also, initial efforts to implement MCPI using parallel computation have shown additional speedup [13, 14].

The present study considers the gravity-perturbed acceleration  $\ddot{\mathbf{r}} = \mathbf{g}(t, \mathbf{r})$ , which may be represented in state space notation for the MCPI algorithm as

$$\dot{\mathbf{x}} = \begin{pmatrix} \dot{\mathbf{r}} \\ \dot{\dot{\mathbf{r}}} \end{pmatrix}, \quad \begin{pmatrix} \dot{\mathbf{x}}_1 \\ \dot{\mathbf{x}}_2 \end{pmatrix} = \mathbf{f}(t, \mathbf{x}) = \begin{pmatrix} \mathbf{x}_2 \\ \mathbf{g}(t, \mathbf{x}_1) \end{pmatrix} \tag{4}$$

### Basic Equations for State Transition Matrix

The differential equation we wish to integrate for the STM is [8]

$$\dot{\Phi}(t, t_0) = A\Phi(t, t_0) \tag{5}$$

$$A(t) = \begin{bmatrix} 0_{n \times n} & I_{n \times n} \\ G_{n \times n} & 0_{n \times n} \end{bmatrix} \triangleq \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial \mathbf{x}_1} \\ \frac{\partial \mathbf{f}}{\partial \mathbf{x}_2} \end{bmatrix} = \begin{bmatrix} \frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{x}_2}{\partial \mathbf{x}_2} \\ \frac{\partial \mathbf{g}}{\partial \mathbf{x}_1} & \frac{\partial \mathbf{g}}{\partial \mathbf{x}_2} \end{bmatrix} \tag{6}$$

where

$$G_{n \times n} = \begin{bmatrix} \frac{\partial \mathbf{g}(t, \mathbf{r})}{\partial \mathbf{r}} \end{bmatrix} \tag{7}$$

The STM is computed as partials of the linear approximation of the instantaneous departure  $\mathbf{X}(t)$  from the state vector  $\mathbf{x}(t)$ . For the present work, the state consists of position and velocity for an Earth-orbiting satellite:

$$\Phi(t, t_0) = \frac{\partial \mathbf{X}(t)}{\partial \mathbf{X}(t_0)} \tag{8}$$

Then we can write the  $G$  matrix in terms of the partials of the generally perturbed gravitational acceleration case; for example the partial of the first component

of acceleration with respect to the first component in a body-fixed frame is written with respect to geocentric radius, latitude, and longitude:

$$\frac{\partial a_x}{\partial x} = \frac{\partial a_x}{\partial r} \frac{\partial r}{\partial x} + \frac{\partial a_x}{\partial \phi} \frac{\partial \phi}{\partial x} + \frac{\partial a_x}{\partial \lambda} \frac{\partial \lambda}{\partial x} \tag{9}$$

Many of the intermediate partials are computed once and re-used as they appear elsewhere. The individual components of the matrix  $G$  are written explicitly as follows in the next section.

### State Transition Matrix Using Spherical Harmonic Gravity

A spherical harmonic gravity model is used for this development. The Jacobian of the acceleration, needed for Eq. 6, is computed in the body frame but transformed into the inertial frame prior to each integration step; this method is presented in the following section. The full gravitational potential function is defined as [15]

$$U(r, \phi, \lambda) = \frac{\mu}{r} \left[ 1 + \sum_{n=2}^{\infty} \sum_{m=0}^n \left( \frac{Re}{r} \right)^n P_{nm}(\sin \phi) [C_{nm} \cos(m\lambda) \times + S_{nm} \sin(m\lambda)] \right] \tag{10}$$

where  $r$  is the radial distance to the object,  $\phi$  is the geocentric latitude of the object,  $\lambda$  is the longitude of the object,  $Re$  is the magnitude of Earth’s equatorial radius, and  $P_{nm}$  are the Associated Legendre Functions. The spherical harmonic acceleration due to gravity in a body-fixed reference frame may then be represented as

$$\mathbf{a}_g = \frac{\partial U}{\partial r} \left( \frac{\partial r}{\partial \mathbf{r}} \right)^T + \frac{\partial U}{\partial \phi} \left( \frac{\partial \phi}{\partial \mathbf{r}} \right)^T + \frac{\partial U}{\partial \lambda} \left( \frac{\partial \lambda}{\partial \mathbf{r}} \right)^T \tag{11}$$

The expressions comprising the acceleration function are written explicitly as

$$\begin{aligned} \frac{\partial U}{\partial r} &= U_r \\ &= -\frac{\mu}{r^2} \left\{ 1 + \sum_{n=2}^{\infty} \sum_{m=0}^n (n+1) \left( \frac{Re}{r} \right)^2 P_{nm}(\sin \phi) [C_{nm} \cos(m\lambda) \right. \\ &\quad \left. + S_{nm} \sin(m\lambda)] \right\} \end{aligned} \tag{12}$$

$$\frac{\partial U}{\partial \phi} = U_\phi = \frac{\mu}{r} \sum_{n=2}^{\infty} \sum_{m=0}^n \left( \frac{Re}{r} \right)^n \frac{\partial P_{nm}(\sin \phi)}{\partial \phi} [C_{nm} \cos(m\lambda) + S_{nm} \sin(m\lambda)] \tag{13}$$

$$\frac{\partial U}{\partial \lambda} = U_\lambda = \frac{\mu}{r} \sum_{n=2}^{\infty} \sum_{m=0}^n \left( \frac{Re}{r} \right)^n P_{nm}(\sin \phi) m [S_{nm} \cos(m\lambda) - C_{nm} \sin(m\lambda)] \tag{14}$$

Next we find expressions needed to compute the Jacobian of the acceleration in Eq. 7. The individual components of the gravity perturbed acceleration in the Earth-centered inertial frame are

$$a_X = \frac{\partial U}{\partial r} \left( \frac{\partial r}{\partial x} \right) + \frac{\partial U}{\partial \phi} \left( \frac{\partial \phi}{\partial x} \right) + \frac{\partial U}{\partial \lambda} \left( \frac{\partial \lambda}{\partial x} \right) \tag{15}$$

$$a_Y = \frac{\partial U}{\partial r} \left( \frac{\partial r}{\partial y} \right) + \frac{\partial U}{\partial \phi} \left( \frac{\partial \phi}{\partial y} \right) + \frac{\partial U}{\partial \lambda} \left( \frac{\partial \lambda}{\partial y} \right) \tag{16}$$

$$a_Z = \frac{\partial U}{\partial r} \left( \frac{\partial r}{\partial z} \right) + \frac{\partial U}{\partial \phi} \left( \frac{\partial \phi}{\partial z} \right) + \frac{\partial U}{\partial \lambda} \left( \frac{\partial \lambda}{\partial z} \right) \tag{17}$$

For the partial fraction expansions of the components of  $G$ , we use the general formula:

$$\frac{\partial}{\partial \beta} (\cdot) = \frac{\partial}{\partial r} (\cdot) \left( \frac{\partial r}{\partial \beta} \right) + \frac{\partial}{\partial \phi} (\cdot) \left( \frac{\partial \phi}{\partial \beta} \right) + \frac{\partial}{\partial \lambda} (\cdot) \left( \frac{\partial \lambda}{\partial \beta} \right) \quad ; \quad \beta \rightarrow x, y, z \tag{18}$$

The individual components of  $G$  then follow a pattern. The first few expressions are given here, where  $U_r = \frac{\partial U}{\partial r}$ ,  $U_\phi = \frac{\partial U}{\partial \phi}$ , and  $U_\lambda = \frac{\partial U}{\partial \lambda}$  as given in Eqs. 12 – 14, and all  $G$  components are given in Appendix A.

$$\begin{aligned} G_{11} = \frac{\partial}{\partial x} (a_X) &= \left( \frac{\partial r}{\partial x} \right)^2 \left( \frac{\partial U_r}{\partial r} \right) + \left( \frac{\partial \phi}{\partial x} \right)^2 \left( \frac{\partial U_\phi}{\partial \phi} \right) + \left( \frac{\partial \lambda}{\partial x} \right)^2 \left( \frac{\partial U_\lambda}{\partial \lambda} \right) \\ &+ 2 \left( \frac{\partial r}{\partial x} \right) \left( \frac{\partial \phi}{\partial x} \right) \left( \frac{\partial U_r}{\partial \phi} \right) + 2 \left( \frac{\partial r}{\partial x} \right) \left( \frac{\partial \lambda}{\partial x} \right) \left( \frac{\partial U_r}{\partial \lambda} \right) + 2 \left( \frac{\partial \phi}{\partial x} \right) \left( \frac{\partial \lambda}{\partial x} \right) \left( \frac{\partial U_\phi}{\partial \lambda} \right) \\ &+ \left( \frac{\partial U}{\partial r} \right) \left( \frac{\partial^2 r}{\partial x^2} \right) + \left( \frac{\partial U}{\partial \phi} \right) \left( \frac{\partial^2 \phi}{\partial x^2} \right) + \left( \frac{\partial U}{\partial \lambda} \right) \left( \frac{\partial^2 \lambda}{\partial x^2} \right) \end{aligned} \tag{19}$$

$$\begin{aligned} G_{12} = \frac{\partial}{\partial y} (a_X) &= \left( \frac{\partial r}{\partial y} \right) \left( \frac{\partial r}{\partial x} \right) \left( \frac{\partial U_r}{\partial r} \right) + \left( \frac{\partial r}{\partial y} \right) \left( \frac{\partial \phi}{\partial x} \right) \left( \frac{\partial U_r}{\partial \phi} \right) + \left( \frac{\partial r}{\partial y} \right) \left( \frac{\partial \lambda}{\partial x} \right) \left( \frac{\partial U_r}{\partial \lambda} \right) \\ &+ \left( \frac{\partial \phi}{\partial y} \right) \left( \frac{\partial r}{\partial x} \right) \left( \frac{\partial U_\phi}{\partial r} \right) + \left( \frac{\partial \phi}{\partial y} \right) \left( \frac{\partial \phi}{\partial x} \right) \left( \frac{\partial U_\phi}{\partial \phi} \right) + \left( \frac{\partial \phi}{\partial y} \right) \left( \frac{\partial \lambda}{\partial x} \right) \left( \frac{\partial U_\phi}{\partial \lambda} \right) \\ &+ \left( \frac{\partial \lambda}{\partial y} \right) \left( \frac{\partial r}{\partial x} \right) \left( \frac{\partial U_\lambda}{\partial r} \right) + \left( \frac{\partial \lambda}{\partial y} \right) \left( \frac{\partial \phi}{\partial x} \right) \left( \frac{\partial U_\lambda}{\partial \phi} \right) + \left( \frac{\partial \lambda}{\partial y} \right) \left( \frac{\partial \lambda}{\partial x} \right) \left( \frac{\partial U_\lambda}{\partial \lambda} \right) \\ &+ \left( \frac{\partial U}{\partial r} \right) \left( \frac{\partial^2 r}{\partial x \partial y} \right) + \left( \frac{\partial U}{\partial \phi} \right) \left( \frac{\partial^2 \phi}{\partial x \partial y} \right) + \left( \frac{\partial U}{\partial \lambda} \right) \left( \frac{\partial^2 \lambda}{\partial x \partial y} \right) \end{aligned} \tag{20}$$

$$\begin{aligned} G_{13} = \frac{\partial}{\partial z} (a_X) &= \left( \frac{\partial r}{\partial z} \right) \left( \frac{\partial r}{\partial x} \right) \left( \frac{\partial U_r}{\partial r} \right) + \left( \frac{\partial r}{\partial z} \right) \left( \frac{\partial \phi}{\partial x} \right) \left( \frac{\partial U_r}{\partial \phi} \right) + \left( \frac{\partial r}{\partial z} \right) \left( \frac{\partial \lambda}{\partial x} \right) \left( \frac{\partial U_r}{\partial \lambda} \right) \\ &+ \left( \frac{\partial \phi}{\partial z} \right) \left( \frac{\partial r}{\partial x} \right) \left( \frac{\partial U_\phi}{\partial r} \right) + \left( \frac{\partial \phi}{\partial z} \right) \left( \frac{\partial \phi}{\partial x} \right) \left( \frac{\partial U_\phi}{\partial \phi} \right) + \left( \frac{\partial \phi}{\partial z} \right) \left( \frac{\partial \lambda}{\partial x} \right) \left( \frac{\partial U_\phi}{\partial \lambda} \right) \\ &+ \left( \frac{\partial \lambda}{\partial z} \right) \left( \frac{\partial r}{\partial x} \right) \left( \frac{\partial U_\lambda}{\partial r} \right) + \left( \frac{\partial \lambda}{\partial z} \right) \left( \frac{\partial \phi}{\partial x} \right) \left( \frac{\partial U_\lambda}{\partial \phi} \right) + \left( \frac{\partial \lambda}{\partial z} \right) \left( \frac{\partial \lambda}{\partial x} \right) \left( \frac{\partial U_\lambda}{\partial \lambda} \right) \\ &+ \left( \frac{\partial U}{\partial r} \right) \left( \frac{\partial^2 r}{\partial x \partial z} \right) + \left( \frac{\partial U}{\partial \phi} \right) \left( \frac{\partial^2 \phi}{\partial x \partial z} \right) + \left( \frac{\partial U}{\partial \lambda} \right) \left( \frac{\partial^2 \lambda}{\partial x \partial z} \right) \end{aligned} \tag{21}$$

The spherical coordinate partials with respect to Cartesian coordinates are given by

$$\begin{aligned} \frac{\partial r}{\partial \alpha} = \frac{\alpha}{r}; \quad \alpha \rightarrow x, y, z \quad & \frac{\partial \phi}{\partial x} = \frac{-xz}{r^2 \sqrt{x^2+y^2}} \quad \frac{\partial \phi}{\partial y} = \frac{-yz}{r^2 \sqrt{x^2+y^2}} \quad \frac{\partial \phi}{\partial z} = \frac{(1-\frac{z^2}{r^2})}{\sqrt{x^2+y^2}} \\ & \frac{\partial \lambda}{\partial x} = \frac{-y}{x^2+y^2} \quad \frac{\partial \lambda}{\partial y} = \frac{x}{x^2+y^2} \quad \frac{\partial \lambda}{\partial z} = 0 \end{aligned} \tag{22}$$

Then, the second partials of the gravity potential are found, where the partials of the ALFs are described in the following section. The matrix of second partials is symmetric, and the six expressions for the distinct elements are as follows:

$$\frac{\partial^2 U}{\partial r^2} = \frac{\mu}{r^3} \left\{ 2 + \sum_{n=2}^{\infty} \sum_{m=0}^n (n+1)(n+2) \times \left(\frac{Re}{r}\right)^n P_{nm}(\sin\phi) [C_{nm} \cos(m\lambda) + S_{nm} \sin(m\lambda)] \right\} \tag{23}$$

$$\frac{\partial^2 U}{\partial r \partial \phi} = -\frac{\mu}{r^2} \sum_{n=2}^{\infty} \sum_{m=0}^n (n+1) \left(\frac{Re}{r}\right)^n \frac{\partial P_{nm}(\sin\phi)}{\partial \phi} \times [C_{nm} \cos(m\lambda) + S_{nm} \sin(m\lambda)] \tag{24}$$

$$\frac{\partial^2 U}{\partial r \partial \lambda} = -\frac{\mu}{r^2} \sum_{n=2}^{\infty} \sum_{m=0}^n (n+1) \left(\frac{Re}{r}\right)^n P_{nm}(\sin\phi) \times [S_{nm} \cos(m\lambda) - C_{nm} \sin(m\lambda)] \tag{25}$$

$$\frac{\partial^2 U}{\partial \phi^2} = \frac{\mu}{r} \sum_{n=2}^{\infty} \sum_{m=0}^n \left(\frac{Re}{r}\right)^n \frac{\partial^2 P_{nm}(\sin\phi)}{\partial \phi^2} \times [C_{nm} \cos(m\lambda) + S_{nm} \sin(m\lambda)] \tag{26}$$

$$\frac{\partial^2 U}{\partial \phi \partial \lambda} = \frac{\mu}{r} \sum_{n=2}^{\infty} \sum_{m=0}^n \left(\frac{Re}{r}\right)^n \frac{\partial P_{nm}(\sin\phi)}{\partial \phi} \times m [S_{nm} \cos(m\lambda) - C_{nm} \sin(m\lambda)] \tag{27}$$

$$\frac{\partial^2 U}{\partial \lambda^2} = -\frac{\mu}{r} \sum_{n=2}^{\infty} \sum_{m=0}^n \left(\frac{Re}{r}\right)^n P_{nm}(\sin\phi) \times m^2 [S_{nm} \sin(m\lambda) + C_{nm} \cos(m\lambda)] \tag{28}$$

### Computation of Associated Legendre Functions

A key component of the spherical harmonic gravity calculations is the set of Associated Legendre Functions,  $P_{nm}$ . These functions are related to the Legendre polynomials,  $P_n(u)$  [16].

$$P_{n0}(u) = P_n(u) = \frac{1}{2^n n!} \frac{d^n}{du^n} (u^2 - 1)^n \tag{29}$$

$$P_{nm}(u) = (1 - u^2)^{\frac{m}{2}} \frac{d^m}{du^m} P_n(u) = (1 - u^2)^{\frac{m}{2}} \frac{1}{2^m n!} \frac{d^{m+n}}{du^{m+n}} (u^2 - 1)^n \tag{30}$$

The equation for the gravity potential only contains a singularity at  $r = 0$ , but the gravity force in spherical coordinates, given below, contains a singularity at the poles  $\phi = +\frac{\pi}{2}, -\frac{\pi}{2}$ . This special case at the poles is addressed by setting the  $g_x$  and  $g_y$  components of acceleration to zero and computing  $g_z$  as a function of  $\frac{\partial U}{\partial r}$  only (zeroing out the  $\frac{\partial U}{\partial \phi}$  term).

$$f = \nabla U = \frac{\partial U}{\partial r} \mathbf{u}_r + \frac{1}{r} \frac{\partial U}{\partial \phi} \mathbf{u}_\phi + \frac{1}{r \cos \phi} \frac{\partial U}{\partial \lambda} \mathbf{u}_\lambda \tag{31}$$

In many applications the potential function is represented using the so-called derived Associated Legendre Functions. The derived ALFs are defined as follows [17, 18]:

$$A_{nm}(u) = \frac{1}{2^n n!} \frac{d^{m+n}}{du^{m+n}} (u^2 - 1)^n = \frac{d^m}{du^m} P_n(u) \tag{32}$$

Following the usual patterns, we can avoid weak numerical instability of the typical recursions used to compute the  $A_{nm}$  (see Table 1 of Reference [16]). To overcome this problem, the derived Associated Legendre Functions and spherical harmonic coefficients are normalized to improve accuracy for high degree and order gravity models. First, define  $A_{nm}$  as an un-normalized, derived Associated Legendre Function and  $\bar{A}_{nm}$  as a normalized, derived Associated Legendre Function as given in Reference [17]. Similarly,  $P_{nm}$  is an un-normalized, Associated Legendre Function, while  $\bar{P}_{nm}$  is a normalized, Associated Legendre Function. Since the spherical harmonics correspond to a linear subspace of 2-D Fourier Series, a typical normalization factor is found; from Ch. 3, formula (7) of Reference [18], the derived ALF may be expanded, giving rise to a coefficient that generates numerically stable recursive calculations. This coefficient is often used in geodesy applications [16, 19–23]. Then, the normalized and un-normalized, derived ALFs are related by this normalization, written generally for  $n \neq m \neq 0$  as [16]

$$\bar{A}_{nm} = \left[ \frac{(n - m)! (2n + 1) (2 - \delta_{0m})}{(n + m)!} \right]^{1/2} A_{nm} = N_{nm} A_{nm} \tag{33}$$

where  $N_{nm}$  is a scaling factor and the Kronecker delta function is defined to be

$$\delta_{0m} = \begin{cases} 1 & m = 0 \\ 0 & m \neq 0 \end{cases} \tag{34}$$

The recursion formula chosen for the current work is the normalized, derived Associated Legendre Functions from Table 2, option I of Reference [16] for  $u = \sin \phi$ :

$$\begin{aligned} \bar{A}_{nm} = u & \left[ \frac{(2n + 1) (2n - 1)}{(n - m) (n + m)} \right]^{1/2} \bar{A}_{n-1,m} \\ & - \left[ \frac{(2n + 1) (n - m - 1) (n + m - 1)}{(2n - 3) (n + m) (n - m)} \right]^{1/2} \bar{A}_{n-2,m} \end{aligned} \tag{35}$$

Note that the specific cases of  $n = m$  and  $m = 0$  may each use a more specific formula for efficiency.

Since a normalized version of the ALFs is used, the final un-normalized result may be obtained by applying the appropriate scale factor. For efficiency, the scale factor  $S_{nm}$  for three cases are computed separately:

$$m = 0 : \quad S_{nm} = \sqrt{\frac{n(n + 1)}{2}} \tag{36}$$

$$m = n : \quad S_{nm} = 0 \tag{37}$$

$$m \neq n \neq 0 : \quad S_{nm} = \sqrt{(n - m) (m + n + 1)} \tag{38}$$



Computation of the STM for the spherical harmonic gravity case requires the partial derivative of  $P_{nm}$  with respect to  $\phi$ , where  $C_{nm}$  and  $S_{nm}$  are the normalized Stokes coefficients determined from satellite motion observations, and  $N_{nm}$  is the scale factor given in Eq. 33. In Appendix B, the equations for the first and second partials of  $P_{nm}$  with respect to  $\phi$  are derived in detail. For more information, these partials are also given in Reference [24].

The first and second partials of the ALFs are incorporated in the calculations for the partials of the gravity potential,  $U$ , through the computation of the corresponding ALFs and the appropriate scale factors. In the code, the term  $P_{n,m+1}(\sin \phi)$  is multiplied by the scale factor  $S_{nm}$ , and the term  $P_{n,m+2}(\sin \phi)$  is multiplied by the scale factor  $S_{n,m+1}$  to compensate for the original normalization of the ALFs.

### Verification for Second Partial of the Associated Legendre Functions with Respect to Latitude

The second partial of the gravity potential with respect to geocentric latitude is checked since it is a rigorous solution of Laplace's equation [23]:

$$r^2 \nabla^2 U = \frac{\partial}{\partial r} \left( r^2 \frac{\partial U}{\partial r} \right) + \frac{1}{\cos \phi} \frac{\partial}{\partial \phi} \left( \cos \phi \frac{\partial U}{\partial \phi} \right) + \frac{1}{\cos^2 \phi} \frac{\partial^2 U}{\partial \lambda^2} = 0 \quad (39)$$

which is simplified as

$$\frac{\partial^2 U}{\partial \phi^2} = - \left[ r^2 \frac{\partial^2 U}{\partial r^2} + 2r \frac{\partial U}{\partial r} - \tan \phi \frac{\partial U}{\partial \phi} + \frac{1}{\cos^2 \phi} \frac{\partial^2 U}{\partial \lambda^2} \right] \quad (40)$$

This equation is also compared numerically with Eq. 26 to further confirm that the expression for the second partial of the ALF with respect to latitude is correct [25]. See the next section for resulting plots.

### Baseline Model Simulation Results

All simulation results presented here are computed on a Windows 8 machine using Matlab R2013a. First, to show that the second partial of the normalized ALFs is computed correctly in MCPI, a finite difference approximation method is incorporated into the simulation. Since the present STM study gives efficient and accurate results for all elliptical orbits, the user may specify initial conditions and also the desired tolerance. MCPI may also be tuned based on the number of segments per orbit, relative segment length, and order of approximation [26, 27]. Here, we focus on a Low-Earth Orbit (LEO) example for the following results with one segment per orbit. The user may specify initial conditions and also the desired tolerance. The chosen initial conditions for this orbit are given to be  $\mathbf{r}_0 = [2.865408457 \ 5.191131097 \ 2.848416876] \times 10^6 \text{ m}$ ,  $\mathbf{v}_0 = [-5.386247766 \ -0.3867151905 \ 6.123151881] \times 10^3 \text{ m/s}$ , with a period of  $6.218728118 \times 10^3 \text{ s}$ . All solutions are obtained for one orbit period using a 10<sup>th</sup> degree and order gravity, Earth rotation of  $\omega = 7.2921\text{e}^{-5} \text{ rad/s}$ , and 100

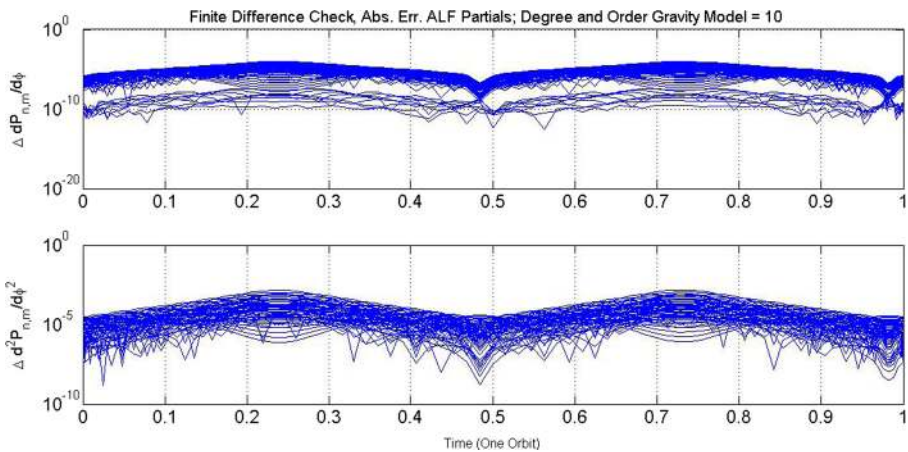
sample points (using a cosine sampling scheme). The RMS error between the initial and final position is found to be  $[1.225 \ 2.054 \ 4.925] \times 10^{-3} \ m$ , and the RMS error between the initial and final velocity is  $[3.515 \ 2.690 \ 3.267] \times 10^{-6} \ m/s$ .

As can be seen in Fig. 1, the first and second partials of the Legendre Functions match closely with the finite difference check approximations (the partials use a normalization scheme as described in the present paper). The ALFs and the first and second partials of the ALFs are computed as lower triangular matrices that depend on  $n$  and  $m$ , which are the upper summation limits of Eq. 10. Therefore, the size of these matrices are defined by user inputs.

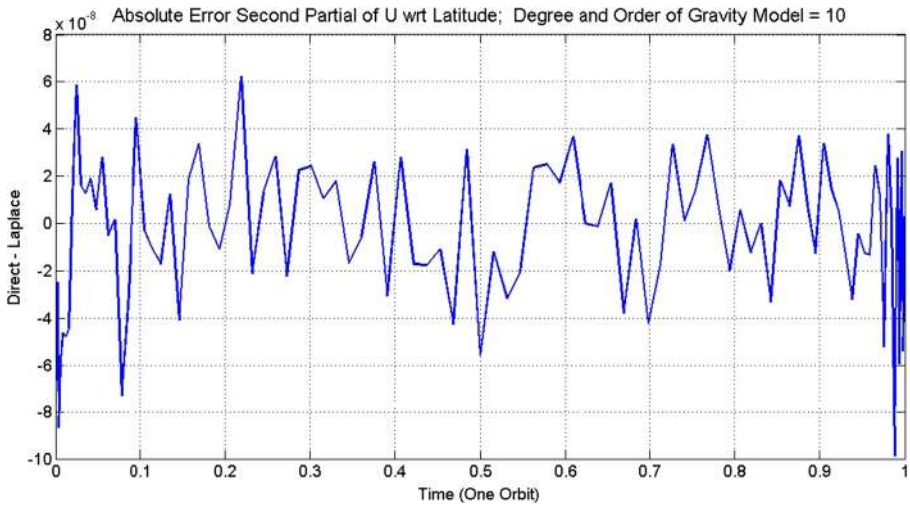
Similarly, to verify that the formulation developed here for the STM is correct using MCPI, a finite difference check of both the second partials of the gravity potential and the STM matrix are completed and give results comparable in accuracy to the figure above.

An additional verification of the second partial with respect to latitude,  $\phi$ , uses the Laplace Equation as described in the previous section. The second partial  $\frac{\partial^2 U}{\partial \phi^2}$  obtained using this method is compared with the second partial computed using the expression derived in Eq. 26, which is directly a function of the ALFs. The result of this comparison is shown in Figs. 2 and 3. The relative error plot shows spikes that are due to the second partial  $\frac{\partial^2 U}{\partial \phi^2}$  approaching zero as a result of the normalization factor in the denominator.

The symplectic check [9] reveals an accuracy of at least  $10^{-10}$  for all error components, as is seen in Fig. 4. Note that each component of the  $6 \times 6$  matrix of error components is plotted here. A final finite difference check for the STM shows that it is very accurate. Each column of  $\phi$  is checked using a two-sided finite difference check. The difference between the first column of the STM and the corresponding finite difference check is shown in Fig. 5, though all 6 columns show comparable accuracy for this check. The top half of Fig. 5 gives the finite difference check



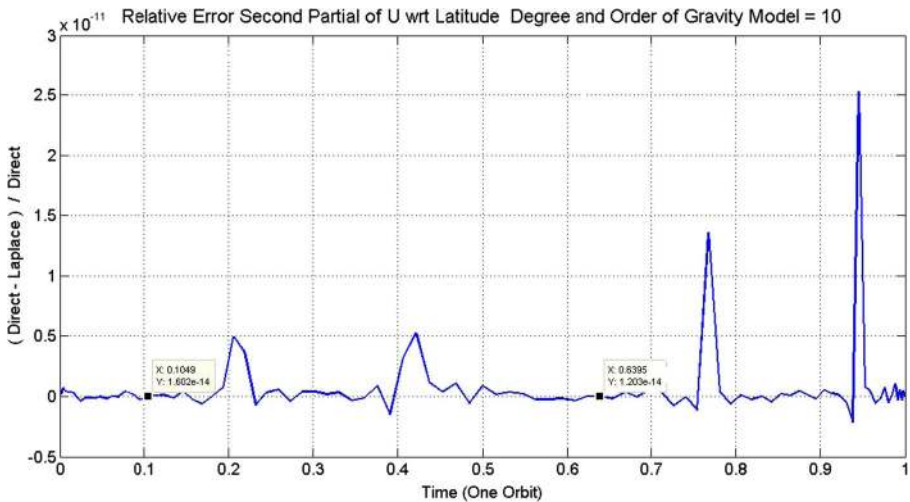
**Fig. 1** Finite difference check for first and second partial derivatives of ALFs



**Fig. 2** Absolute error check for second partial of gravity potential with respect to latitude

components on the STM’s top half of the first column, while the bottom half of the figure gives the components on the STM’s bottom half of the first column.

To verify that the STM differential equation holds for arbitrary perturbations, the group properties of the STM are rigorously verified. A representative set of intermediate points are checked using the chain rule and are satisfied with an accuracy of at least  $10^{-9}$  tolerance. The inverse property of the STM is also verified to high



**Fig. 3** Relative error check for second partial of gravity potential with respect to latitude

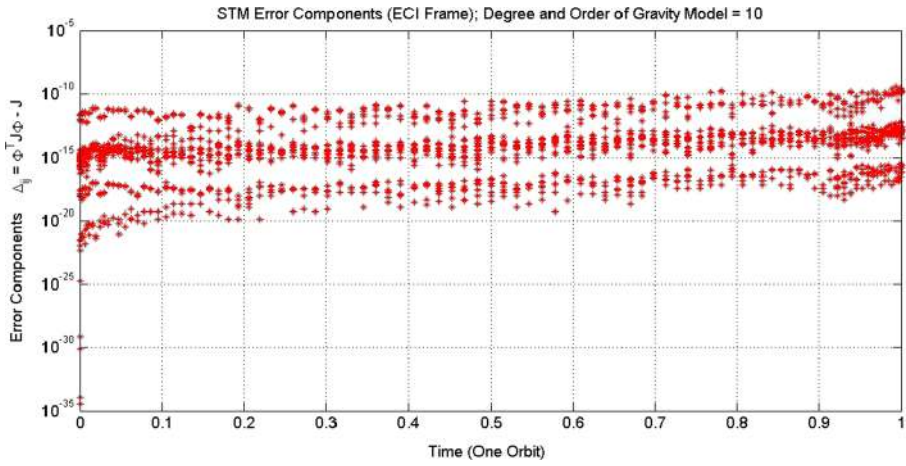


Fig. 4 Symplectic check for final STM over one orbit

precision. The max error over 1000 tests for the residual error  $\|\dot{\phi} - A\phi\|$  is  $1 \times 10^{-15}$ , while the other group properties are spot checked for accuracy:

$$\phi(t_i, t_j) = \phi(t_j, t_i)^{-1} \tag{41}$$

$$\phi(t_i, t_j) = \phi(t_i, t_k) \phi(t_k, t_j) \tag{42}$$

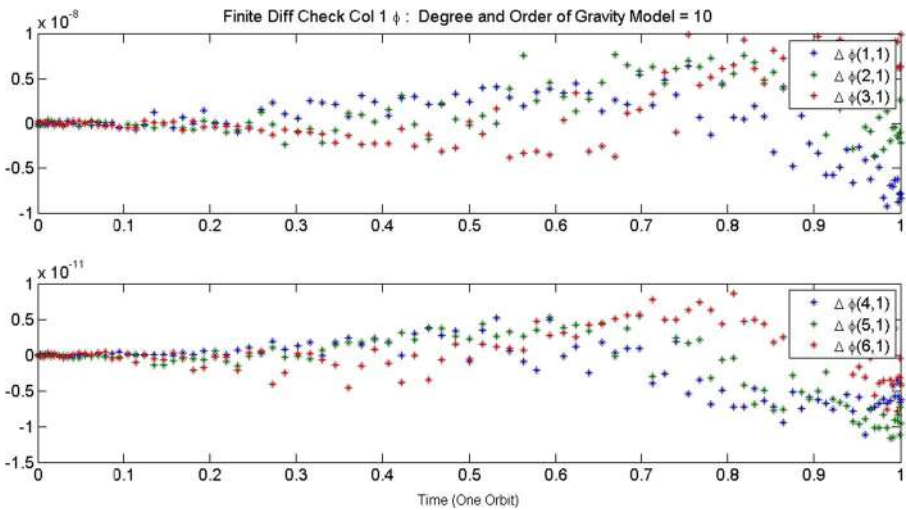


Fig. 5 Finite difference check for first column of STM over one orbit

It should not be surprising that MCPI can capture the solution accurately over a larger time interval than, for example, the Gauss Jackson 8<sup>th</sup> order method, since the typical order of the MCPI approximation is 40 instead of 8.

### Optimization of State Transition Matrix Calculations

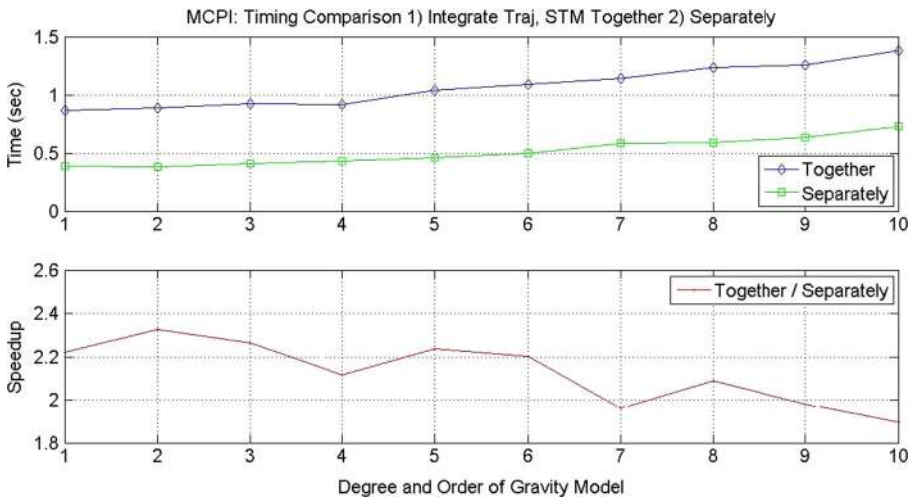
The baseline STM algorithm, given in detail in the present work, may be further optimized and enhanced. Because the STM requires only as input the position vector at every sample node, the trajectory may first be integrated. Next, the STM may be propagated using the converged trajectory solution. This method is more efficient, as is shown by the comparison in Fig. 6.

This approach to have the trajectory converge a priori does not require that the same nodal pattern or time segment length be used for the STM.

### Cascade Method

The MCPI method may solve either first- or second-order differential equations. However, some numerical integrators, such as RKN12(10), require a second-order formulation. To increase efficiency of STM calculations, a second-order differential equation may be used in place of Eq. 1; this method is called the MCPI Cascade Method [2]. The STM differential equation for the conservative case is rearranged to solve a pair of second-order equations as follows. Since  $\dot{\phi} = A\phi$ , or

$$\begin{bmatrix} \dot{\phi}_{11} & \dot{\phi}_{12} \\ \dot{\phi}_{21} & \dot{\phi}_{22} \end{bmatrix} = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} \\ G_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix} \tag{43}$$



**Fig. 6** Timing comparison of trajectory and state transition matrix: propagated both simultaneously and separately over one orbit

we write the individual components using the  $G$  matrix from Eq. 7 as

$$\dot{\phi}_{11} = \phi_{21} \tag{44}$$

$$\dot{\phi}_{12} = \phi_{22} \tag{45}$$

$$\dot{\phi}_{21} = G\phi_{11} \tag{46}$$

$$\dot{\phi}_{22} = G\phi_{12} \tag{47}$$

Taking the time derivative of Eqs. 44 and 45 gives

$$\ddot{\phi}_{11} = \dot{\phi}_{21} \tag{48}$$

$$\ddot{\phi}_{12} = \dot{\phi}_{22} \tag{49}$$

Substituting Eq. 46 into Eq. 48 and Eq. 47 into Eq. 49 gives the only two second-order differential equations required for integration:

$$\ddot{\phi}_{11} = G\phi_{11} \tag{50}$$

$$\ddot{\phi}_{12} = G\phi_{12} \tag{51}$$

The other two sub-matrices of the STM,  $\phi_{21}$  and  $\phi_{22}$ , are obtained by taking the time derivative of the two sub-matrices that are obtained from the converged solution,  $\dot{\phi}_{21}$  and  $\dot{\phi}_{22}$ :

$$\phi_{21} = \dot{\phi}_{11} \tag{52}$$

$$\phi_{22} = \dot{\phi}_{12} \tag{53}$$

The initial condition for the STM is known to be

$$\begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix} = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \tag{54}$$

so for the second-order formulation, the initial conditions are

$$\begin{bmatrix} \dot{\phi}_{11} & \dot{\phi}_{12} \end{bmatrix} = \begin{bmatrix} \phi_{21} & \phi_{22} \end{bmatrix} = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \tag{55}$$

$$\begin{bmatrix} \phi_{11} & \phi_{12} \end{bmatrix} = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \tag{56}$$

The initial conditions automatically satisfied are  $\phi_{21} = 0_{3 \times 3}$  and  $\phi_{22} = I_{3 \times 3}$ . We mention that the above results for the cascade method can be readily generalized to accommodate velocity dependence in the force model (e.g., drag). In this case of velocity dependence, Eq. 5 generalizes as

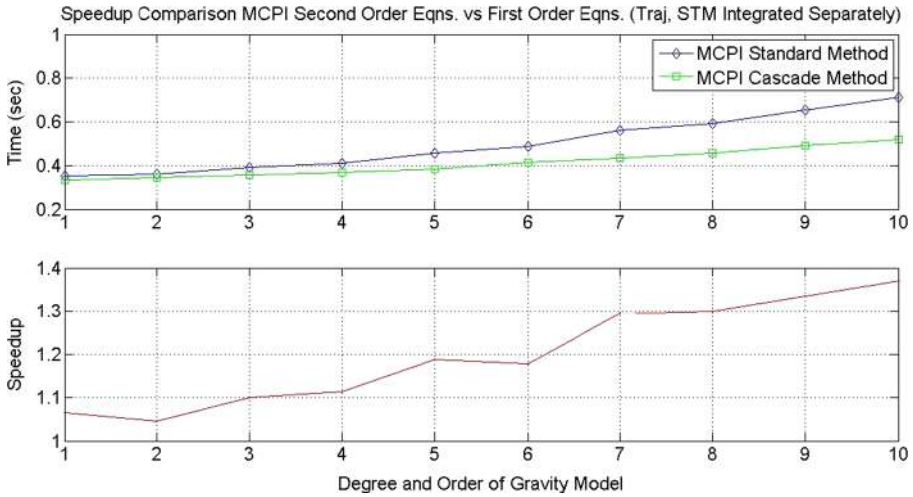
$$\begin{bmatrix} \dot{\phi}_{11} & \dot{\phi}_{12} \\ \dot{\phi}_{21} & \dot{\phi}_{22} \end{bmatrix} = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} \\ G_{3 \times 3} & D_{3 \times 3} \end{bmatrix} \begin{bmatrix} \phi_{11} & \phi_{12} \\ \phi_{21} & \phi_{22} \end{bmatrix} \tag{57}$$

$$D_{3 \times 3} = \begin{bmatrix} \frac{\partial \mathbf{g}(t, \mathbf{r})}{\partial \dot{\mathbf{r}}} \end{bmatrix} \tag{58}$$

And the resulting formulation is the same except for two additional terms of the form

$$\ddot{\phi}_{11} = G\phi_{11} + D \frac{\dot{\phi}_{11}}{dt} \tag{59}$$

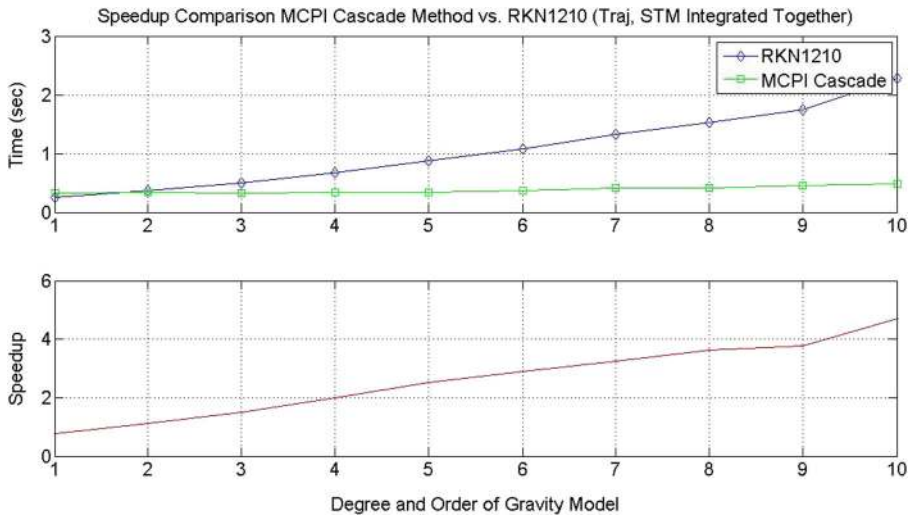
$$\ddot{\phi}_{12} = G\phi_{12} + D \frac{\dot{\phi}_{12}}{dt} \tag{60}$$



**Fig. 7** Timing comparison of baseline vs. cascade method for computing trajectory and subsequently STM from converged position

### Cascade Method Simulation Results

As shown in Figs. 7 and 8, using the Cascade Method to compute the STM gives additional speedup when compared with the baseline method. Figure 7 compares the standard, baseline MCPI model with the MCPI cascade method for computing the trajectory and STM. The results of the cascade method are verified using the RKN12(10) propagator [28], and the resulting timing comparison is given in Fig. 8.



**Fig. 8** Timing comparison for computing trajectory and STM of RKN12(10) with MCPI cascade method

Both MCPI and RKN12(10) are tuned such that best performance is achieved for an energy check that gives machine precision.

### Conclusion

This paper describes how to compute the gravitational potential and associated derivatives necessary for propagation of the State Transition Matrix using the Modified Picard Chebyshev Iteration method. Though the present work focuses on the spherical harmonic gravity model to address gravity perturbations, the STM computation using MCPI may be generally implemented using any other gravity model, and extensions to include other perturbations such as drag are straightforward. Results verifying the accuracy of the baseline solution are presented.

Optimizations, including a second-order formulation for the STM, are presented. This formulation is readily solved using the cascade form of the MCPI formulation, and this is the method adopted for the present study. Matlab timing comparisons of this so-called cascade method are shown to be more efficient than the baseline algorithm for computing the STM. Because a different nodal pattern and precision requirement may be used for the a priori converged trajectory than the STM, extensive computational tests indicate that tuning for the optimal state computation invariably gives conservative STM computation.

C code for the method presented here is currently being developed, which will give additional insight into the runtimes. More importantly, techniques for further accelerating the computations, such as using parallel processing, are expected to exploit the inherent parallel structure of the MCPI method if many simultaneous orbits and their associated STM are required. This class of problems will likely be encountered when these methods are used to propagate space objects and use the STM to approximate the associated covariance matrix evolution [29].

**Acknowledgments** The sponsors for the present work are AFOSR (Julie Moses) and AFRL (Alok Das). Thanks to members of the Texas A&M MCPI research team: Donghoon Kim, Robyn Woollands, Austin Probe, and Abhay Masher.

Thanks to Dr. Bob Gottlieb (Odyssey Space Research) and Dr. Terry Feagin (UHCL) for their discussions of numerical propagators, normalized gravity, and STM calculations.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

### Appendix A: Components of Jacobian Matrix G

$$\begin{aligned}
 G_{11} = \frac{\partial}{\partial x} (a_X) = & \left(\frac{\partial r}{\partial x}\right)^2 \left(\frac{\partial U_r}{\partial r}\right) + \left(\frac{\partial \phi}{\partial x}\right)^2 \left(\frac{\partial U_\phi}{\partial \phi}\right) + \left(\frac{\partial \lambda}{\partial x}\right)^2 \left(\frac{\partial U_\lambda}{\partial \lambda}\right) \\
 & + 2 \left(\frac{\partial r}{\partial x}\right) \left(\frac{\partial \phi}{\partial x}\right) \left(\frac{\partial U_r}{\partial \phi}\right) + 2 \left(\frac{\partial r}{\partial x}\right) \left(\frac{\partial \lambda}{\partial x}\right) \left(\frac{\partial U_r}{\partial \lambda}\right) + 2 \left(\frac{\partial \phi}{\partial x}\right) \left(\frac{\partial \lambda}{\partial x}\right) \left(\frac{\partial U_\phi}{\partial \lambda}\right) \quad (61) \\
 & + \left(\frac{\partial U}{\partial r}\right) \left(\frac{\partial^2 r}{\partial x^2}\right) + \left(\frac{\partial U}{\partial \phi}\right) \left(\frac{\partial^2 \phi}{\partial x^2}\right) + \left(\frac{\partial U}{\partial \lambda}\right) \left(\frac{\partial^2 \lambda}{\partial x^2}\right)
 \end{aligned}$$





$$\begin{aligned}
 G_{33} = \frac{\partial}{\partial z} (a_Z) &= \left(\frac{\partial r}{\partial z}\right)^2 \left(\frac{\partial U_r}{\partial r}\right) + \left(\frac{\partial \phi}{\partial z}\right)^2 \left(\frac{\partial U_\phi}{\partial \phi}\right) + \left(\frac{\partial \lambda}{\partial z}\right)^2 \left(\frac{\partial U_\lambda}{\partial \lambda}\right) \\
 &+ 2 \left(\frac{\partial r}{\partial z}\right) \left(\frac{\partial \phi}{\partial z}\right) \left(\frac{\partial U_r}{\partial \phi}\right) + 2 \left(\frac{\partial r}{\partial z}\right) \left(\frac{\partial \lambda}{\partial z}\right) \left(\frac{\partial U_r}{\partial \lambda}\right) + 2 \left(\frac{\partial \phi}{\partial z}\right) \left(\frac{\partial \lambda}{\partial z}\right) \left(\frac{\partial U_\phi}{\partial \lambda}\right) \\
 &+ \left(\frac{\partial U}{\partial r}\right) \left(\frac{\partial^2 r}{\partial z^2}\right) + \left(\frac{\partial U}{\partial \phi}\right) \left(\frac{\partial^2 \phi}{\partial z^2}\right) + \left(\frac{\partial U}{\partial \lambda}\right) \left(\frac{\partial^2 \lambda}{\partial z^2}\right)
 \end{aligned} \tag{69}$$

### Appendix B: Partial of Associated Legendre Functions

Computation of the STM for the spherical harmonic gravity case requires the partial derivative of  $P_{nm}$  with respect to  $\phi$ , where  $C_{nm}$  and  $S_{nm}$  are the normalized Stokes coefficients determined from satellite motion observations, and  $N_{nm}$  is a scale factor. The first and second partials of the ALFs are incorporated in the calculations for the partials of the gravity potential,  $U$ , through the computation of the corresponding ALFs and the appropriate scale factors. The following known relationships are used for these derivations [16].

$$\frac{\partial}{\partial u} A_{nm}(u) = A_{n,m+1}(u) \tag{70}$$

$$\frac{\partial^2}{\partial u^2} A_{nm}(u) = A_{n,m+2}(u) \tag{71}$$

$$\frac{\partial}{\partial \phi} A_{nm}(\sin \phi) = \frac{\partial A_{nm}(\sin \phi)}{\partial \sin \phi} \frac{\partial \sin \phi}{\partial \phi} = \frac{\partial A_{nm}(\sin \phi)}{\partial \sin \phi} \cos \phi \tag{72}$$

It is known that the derived ALFs are related to the conventional Legendre functions in terms of latitude  $\phi$  through [16]

$$P_{nm}(\sin \phi) = \cos^m \phi A_{nm}(\sin \phi) \tag{73}$$

Starting with the above equation and using the product rule,

$$\frac{\partial P_{nm}(\sin \phi)}{\partial \phi} = \cos^m \phi \frac{\partial A_{nm}(\sin \phi)}{\partial \phi} - m \sin \phi \cos^{m-1} \phi A_{nm}(\sin \phi) \tag{74}$$

This equation is rewritten using Eq. 73 as

$$\begin{aligned}
 \frac{\partial P_{nm}(\sin \phi)}{\partial \phi} &= \cos^m \phi \frac{\partial A_{nm}(\sin \phi)}{\partial \phi} - m \tan \phi \cos^m \phi A_{nm}(\sin \phi) \\
 &= \cos^m \phi \frac{\partial A_{nm}(\sin \phi)}{\partial \phi} - m \tan \phi P_{nm}(\sin \phi)
 \end{aligned} \tag{75}$$

Using Eqs. 70 and 73, the final equation becomes

$$\frac{\partial P_{nm}(\sin \phi)}{\partial \phi} = P_{n,m+1}(\sin \phi) - m \tan \phi P_{nm}(\sin \phi) \tag{76}$$

Similarly, an expression for  $\frac{\partial^2 P_{nm}}{\partial \phi^2}$  is derived.

$$\frac{\partial^2 P_{nm}(\sin \phi)}{\partial \phi^2} = \frac{\partial P_{n,m+1}(\sin \phi)}{\partial \phi} - m \sec^2 \phi P_{nm}(\sin \phi) - m \tan \phi \frac{\partial P_{nm}(\sin \phi)}{\partial \phi} \tag{77}$$

Substituting Eq. 76 into this expression gives

$$\frac{\partial^2 P_{nm}(\sin \phi)}{\partial \phi^2} = \frac{\partial P_{n,m+1}(\sin \phi)}{\partial \phi} - m \sec^2 \phi P_{nm}(\sin \phi) - m \tan \phi [P_{n,m+1}(\sin \phi) - m \tan \phi P_{nm}(\sin \phi)] \quad (78)$$

Equation 76 is used to write

$$\frac{\partial P_{n,m+1}(\sin \phi)}{\partial \phi} = P_{n,m+2}(\sin \phi) - (m+1) \tan \phi P_{n,m+1}(\sin \phi) \quad (79)$$

Therefore, the final expression for the second partial is

$$\frac{\partial^2 P_{nm}(\sin \phi)}{\partial \phi^2} = P_{n,m+2}(\sin \phi) - (2m+1) \tan \phi P_{n,m+1}(\sin \phi) + m(m \tan^2 \phi - \sec^2 \phi) P_{nm}(\sin \phi) \quad (80)$$

In the code, the term  $P_{n,m+1}(\sin \phi)$  is multiplied by the scale factor  $S_{nm}$ , and the term  $P_{n,m+2}(\sin \phi)$  is multiplied by the scale factor  $S_{n,m+1}$  to compensate for the original normalization of the ALFs. The final, normalized equations are then

$$\frac{\partial P_{nm}(\sin \phi)}{\partial \phi} = P_{n,m+1}(\sin \phi) S_{nm} - m \tan \phi P_{nm}(\sin \phi) \quad (81)$$

$$\frac{\partial^2 P_{nm}(\sin \phi)}{\partial \phi^2} = P_{n,m+2}(\sin \phi) S_{n,m+1} - (2m+1) \tan \phi P_{n,m+1}(\sin \phi) S_{nm} + m(m \tan^2 \phi - \sec^2 \phi) P_{nm}(\sin \phi) \quad (82)$$

## References

- Bai, X., Junkins, J.L.: Modified Chebyshev-Picard iteration methods for orbit propagation. *J. Astronaut. Sci.* **58**(4), 583–613 (2011)
- Bai, X.: Modified chebyshev-picard iteration methods for solution of initial value and boundary value problems. PhD Dissertation, Texas A&M University, College Station, TX (2010)
- Woollands, R., Junkins, J.L.: A new solution for the General Lambert problem. In: Proceeding of 2015 AAS GN&C Conference, Breckenridge, CO (2014)
- Macomber, B., Woollands, R., et al.: Modified Chebyshev Picard iteration for efficient numerical integration of ordinary differential equations. In: Proceeding of 2013 AAS GN&C Conference, Breckenridge, CO (2013)
- Kim, D., Junkins, J.L.: Multi-segment adaptive modified Chebyshev Picard iteration method. In: Proceeding of 2014 AAS/AIAA Space Flight Mechanics Conference, Santa Fe, NM (2014)
- Woollands, R. et al.: Validation of accuracy and efficiency of long-arc orbit propagation using the method of manufactured solutions and the round-trip-closure method. In: Proceeding of Advanced Maui Optical and Space Surveillance Technologies Conference (2014)
- Probe, A. et al.: Terminal convergence approximation modified Chebyshev Picard iteration for efficient orbit propagation. In: Proceedings of Advanced Maui Optical and Space Surveillance Technologies Conference (2014)
- Battin, R. *An Introduction to the Mathematics and Methods of Astrodynamics*, Revised Edition. CRC Press - Taylor and Francis, New York, NY (1990)
- Schaub, H., Junkins, J.L. *Analytical Mechanics of Space Systems*, 3<sup>rd</sup> Edition. AIAA Education Series, Reston, VA (2009)
- Clenshaw, C.W., Norton, H.J.: The solution of nonlinear ordinary differential equations in Chebyshev series. *Comput. J.* **6**, 88–92 (1983)
- Feagin, T., Nacozy, P.: Matrix formulation of the Picard method for parallel computation. *Celest. Mech. Dyn. Astron.* **29**, 107–115 (1983)

12. Shaver, J.: Formulation and Evaluation of Parallel Algorithms for the Orbit Determination Problem. PhD Dissertation, Massachusetts Institute of Technology (1980)
13. Bai, X., Junkins, J.L.: Solving Initial Value Problems by the Picard-Chebyshev Method with Nvidia GPUs. In: AAS 10-197, Proceedings of 20th Spaceflight Mechanics Meeting, San Diego, CA (2010)
14. Macomber, B., Probe, A., et al.: Parallel Modified-Chebyshev Picard Iteration for Orbit Catalog Propagation and Monte Carlo Analysis. In: 38th Annual AAS/AIAA Guidance and Control Conference, Breckenridge, CO. Jan 30 - Feb 4 (2015)
15. Hofmann-Wellenhof, B., Moritz, H.: Physical Geodesy. Springer-Verlag Wien (2005)
16. Lundberg, J.B., Schutz, B.: Recursion formulas of legendre functions for use with nonsingular geopotential models. *J. Guid. Control.* **11**(1), 31–38 (1988)
17. Pines, S.: Uniform Representation of the Gravitational Potential and its Derivatives. *AIAA J.* **11**, 1508–1511 (1973)
18. Hobson, E.: Theory of Spherical and Ellipsoidal Harmonics. Cambridge University Press (1931)
19. Colombo, O.L.: Numerical methods for harmonic analyses on the sphere. Air Force Geophysics Laboratory; Air Force Systems Command (1981)
20. Rapp, R.H.: A fortran program for the computation of gravimetric quantities from high degree spherical harmonic expansions. Reports of the Department of Geodetic Science and Surveying, Report No. 334 (1973)
21. Gottlieb, R.G.: Fast gravity, gravity partials, normalized gravity, gravity gradient torque and magnetic field: Derivation, code, and data. Lyndon B. Johnson Space Center (February 1993)
22. Vallado, D.A. Fundamentals of Astrodynamics and Applications, 3<sup>rd</sup> Edition. Microcosm Press and Springer (2007)
23. Kaula, W.M.: Theory of Satellite Geodesy: Applications of Satellites to Geodesy. Blaisdell Publishing Company (2000)
24. Reed, G.B.: Application of Kinematical Geodesy for Determining the Short Wave Length Components of the Gravity Field by Satellite Gradiometry. Reports of the Department of Geodetic Science, Report No. 201 (1973)
25. Schutz, B.: Discussion of normalized associated legendre functions, Austin, TX, May 30 (2014)
26. Macomber, B., Probe, A., Woollands, R., Junkins, J.L.: Automated Tuning Parameter Selection for Orbit Propagation with Modified Chebyshev Picard Iteration." AAS/AIAA Space Flight Mechanics Meeting, Williamsburg, VA (2015)
27. Macomber, B.: Enhancements of Chebyshev Picard Iteration Efficiency for Generally Perturbed Orbits and Constrained Dynamical Systems. PhD Dissertation, Texas A&M University (2015)
28. RKN1210 - A 12th/10th order Runge-Kutta-Nystrom integrator. Matlab Central Web Site. <http://www.mathworks.com/matlabcentral/fileexchange/25291-rkn1210-a-12th-10th-order-runge-kutta-nystr%C3%B6m-integrator>
29. Sabol, C., Schumacher, P., et al.: Linearized Orbit Covariance Generation and Propagation Analysis via Simple Monte Carlo Simulations. In: Proceedings of the AAS/AIAA Space Flight Mechanics Meeting (Advances in the Astronautical Sciences, 2010, 136:509–525)