AAS 16-520

MONTE CARLO PROPAGATION OF ORBITAL ELEMENTS USING MODIFIED CHEBYSHEV PICARD ITERATION

Julie L. Read[*], Tarek A. Elgohary[†], Austin B. Probe[‡], and John L. Junkins[§]

Prior works have shown promising efficiency while propagating perturbed two-body motion using orbital elements combined with a novel integration technique. While previous studies show that Modified Chebyshev Picard Iteration (MCPI) is a powerful tool used to propagate position and velocity, instead using orbital elements to propagate the state vector reduces the number of MCPI iterations required, which is especially useful for reducing the computation time when including computationally-intensive calculations such as Spherical Harmonic gravity, and it also converges for a larger number of revolutions using a single segment (up to 17 orbits compared with 3 orbits for the cartesian, fully spherical harmonic low-Earth orbit case). Results for the Classical Orbital Elements and the Modified Equinoctial Orbital Elements (the latter provides singularity-free solutions) show that state propagation using these variables is inherently well-suited to the propagation method chosen. The present study incorporates a Monte Carlo analysis using a local Taylor Series model that reduces the computational time and provides a high-accuracy solution for propagating the Modified Equinoctial Orbital Elements both in serial and in parallel on a compute cluster.

MCPI is an iterative numerical method used to solve linear and nonlinear, ordinary differential equations (ODEs). It is a fusion of orthogonal Chebyshev function approximation with Picard iteration that approximates a long-arc trajectory at every iteration. Previous studies have shown that it outperforms the state of the practice numerical integrators of ODEs in a serial computing environment; since MCPI is inherently massively parallelizable, this capability increases the computational efficiency of the method presented.

MODIFIED CHEBYSHEV PICARD ITERATION

MCPI[1] is an iterative, path approximation method for solving smoothly nonlinear systems of ordinary differential equations. Clenshaw and Norton[2] first proposed combining the orthogonal Chebyshev polynomials with Picard iteration. Later authors including Shaver, Feagin and Nacozy, and Fukushima further refined the Chebyshev-Picard framework and also pointed out the parallel computing implications of the method.[3–5] More recent developments in parallelizing MCPI give expected increase in efficiency.[6–8]

MCPI is a fusion of Picard iteration, which generates a sequence of path approximations, and Chebyshev Polynomials, which are orthogonal and also enable both efficient and accurate function approximation. This 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

[*]Graduate Research Assistant, Texas A&M University, College Station, TX, USA.
[†]Postdoctoral Researcher, Texas A&M University, College Station, TX, USA.
[‡]Graduate Research Assistant, Texas A&M University, College Station, TX, USA.
[§]Distinguished Professor, Texas A&M University, College Station, TX, USA.

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 using Cartesian coordinates), as opposed to the conventional, incremental step-by-step solution strategy of more familiar numerical integration strategies, such as those based on explicit numerical methods. Significantly, however, unlike conventional integration approaches, it is ideally suited for massive parallel implementations that provide further boosts in the computational performance. Algorithm tests for the present study are currently under development, including massive parallel implementations, where the performance results will be presented in future papers.

In recent years, the research group (Junkins, et. al.) at Texas A&M University has significantly expanded the literature on MCPI. An overview of the the method is given here, while further details may be found in the references.[1,6,9,10] Emile Picard stated that, given an initial condition $\mathbf{x}(t_0) = \mathbf{x}_0$, any first order differential equation

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

with an integrable right hand side may be rearranged, without approximation[9] into an integral:

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

For a given 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

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

Here, the integrand of the Picard iteration sequence is approximated using Chebyshev polynomials. Because the Cheybshev polynomials are orthogonal, a matrix inverse is not necessary to find the basis function coefficients. Also, the Runge Effect (often seen at trajectory boundaries) is greatly reduced due to a cosine sampling scheme. More details on the basics of the MCPI method may be found in the references.[1,9] Additionally, initial efforts to implement MCPI using parallel computation have shown additional speedup.[7,8,11]

Shaver[3] integrated orbital motion using cartesian coordinates (Earth-Centered-Inertial, or ECI) as well as equinoctial orbital elements. He noted that the smooth nature of the element rates makes this set of elements easy to approximate with low-order Chebyshev polynomial series, and that using the Variation of Parameters formulation leads to convergence in significantly fewer iterations. A drag model and low order gravity model were included in these results. Another previous study by Hyun Jo and Choi[12] using the J2 gravity term concluded that using Modified Equinoctial Elements gives a more accurate solution than Classical Orbital Elements. The present study expands Shaver's results on modern processors and incorporates a high order gravity model to gain insight into the convergence domain and accuracy of using orbital elements.

MODIFIED EQUINOCTIAL ORBITAL ELEMENTS

A set of orbital elements is considered; previous work[13] has shown that these Modified Equinoctial Orbital Elements (MEEs) are inherently well-suited to using the MCPI method and converges

over a larger number of orbits than is possible for the Cartesian case. Propagating these MEEs for the Initial Value Problem has shown that the number of MCPI iterations is decreased, leading to a smaller number of full gravity function calls. In addition, we anticipate the state/co-state differential equation systems (corresponding to optimal transfer formulations based on the MEE variation of parameter differential equations) to converge over much longer time intervals.[14]

A similar orbital element set was used more than a century ago by Lagrange to study secular effects due to planetary perturbations and is well suited for orbits with small eccentricities and inclinations. Broucke and Cefola[15] showed the original Equinoctial Elements set to be free of singularities for zero eccentricities and both zero and ninety degree inclinations and also developed a large number of properties and equations for the set. Applications of the Equinoctial Orbital Elements include differential correction in orbit determination and integration of orbits that include special and general perturbations. Broucke and Cefola[15] also developed an explicit form of the state transition matrix (or matrizant), which may be applied in guidance theory and general-perturbations planetary theory with rectangular coordinates.

Brouwer and Clemence[16] discussed the differential correction orbits with several orbital element sets. The Equinoctial Orbital Elements as defined by Broucke and Cefola are similar to the Set III elements (which are represented by nonintegrable differential relations) discussed by Brouwer and Clemence, and they utilize the *h* and *k* elements.

The MEEs are a variation of the original Equinoctial Orbital Elements and are defined in terms of the Classical Orbital Elements as[17,18]

$$p = a(1 - e^2) \tag{4}$$

$$f = ecos(\omega + \Omega) \tag{5}$$

$$g = esin(\omega + \Omega) \tag{6}$$

$$h = tan\left(\frac{i}{2}\right) cos(\Omega) \tag{7}$$

$$k = tan\left(\frac{i}{2}\right) sin(\Omega) \tag{8}$$

$$L = \Omega + \omega + \nu \tag{9}$$

where L is the true longitude. The physical meaning behind these variables is given by Danielson.[19] The inverse relationship is

$$\Omega = tan^{-1}\left(\frac{k}{h}\right) \tag{10}$$

$$\bar{\omega} = \omega + \Omega = tan^{-1}\left(\frac{g}{f}\right) \tag{11}$$

$$\omega = \bar{\omega} - \Omega \tag{12}$$

$$e = \sqrt{f^2 + g^2} \tag{13}$$

$$a = p/(1 - e^2) \tag{14}$$

$$\nu = L - \bar{\omega} \tag{15}$$

The singularity-free equinoctial formulation utilizes the longitudes λ, F, L instsead of the classical anomalies Mean Anomaly, Eccentric Anomaly, and True Anomaly, M, E, ν respectively:[19]

$$\lambda = M + \omega + \Omega \tag{16}$$

$$F = E + \omega + \Omega \tag{17}$$

$$L = \nu + \omega + \Omega \tag{18}$$

The MEE set utilizes p, the semilatus rectum instead of a, the semimajor axis and also L, the true longitude instead of λ, the mean longitude in contrast with the original Equinoctial elements set. It is advantageous to write Kepler's equation in terms of the eccentric longitude F, rather than the eccentric anomaly E, to compute the position vector. This equation and the corresponding radius vector may then be written as

$$\lambda = F + gcos(F) - fsin(F) \tag{19}$$

$$r = a[1 - gsin(F) - fcos(F)] \tag{20}$$

These quantities remain well-defined for the cases of circular or equatorial orbits, eliminating such singular cases known to exist for the Classical Orbital elements. The radius may alternatively be written as

$$r = p/(1 + fcos(L) + gsin(L)) \tag{21}$$

The transformation to and from Classical Orbital Elements and Modified Equinoctial Elements is easily derived and is given by Jung.[20] Since the integration of perturbed orbits requires the transformation between orbital elements and Cartesian coordinates in order to compute the perturbing acceleration, the transformation between the equinoctial frame and the Cartesian frame (and vice versa) is given in detail by Celofa and Broucke.[21] Analogously to the Classical Orbital Elements case, a direction cosine matrix may be used to relate the variables in the equinoctial frame to the Cartesian frame.[15]

$$[NE] = \frac{1}{1 + h^2 + k^2} \begin{bmatrix} 1 - h^2 + k^2 & 2hk & 2h \\ 2hk & 1 + h^2 - k^2 & -2k \\ -2h & 2k & 1 - h^2 - k^2 \end{bmatrix} \tag{22}$$

For this study, the Gauss' equations for the variation of the Modified Equinoctial Orbital Elements are the preferred expressions.[22] The chosen equations are[17,18]

$$\frac{dp}{dt} = \frac{2pC}{w}\sqrt{\frac{p}{\mu}} \tag{23}$$

$$\frac{df}{dt} = \sqrt{\frac{p}{\mu}} \left\{ Ssin(L) + \frac{\big[(w+1)cos(L) + f\big]C}{w} - \frac{g\big(hsin(L) - kcos(L)\big)N}{w} \right\} \tag{24}$$

$$\frac{dg}{dt} = \sqrt{\frac{p}{\mu}} \left\{ -Scos(L) + \frac{\big[(w+1)sin(L) + g\big]C}{w} + \frac{f\big(hsin(L) - kcos(L)\big)N}{w} \right\} \tag{25}$$

$$\frac{dh}{dt} = \sqrt{\frac{p}{\mu}} \frac{s^2 N}{2w} cos(L) \tag{26}$$

$$\frac{dk}{dt} = \sqrt{\frac{p}{\mu}} \frac{s^2 N}{2w} sin(L) \tag{27}$$

$$\frac{dL}{dt} = \sqrt{\mu p} \left(\frac{w}{p}\right)^2 + \sqrt{\frac{p}{\mu}} \frac{\big(hsin(L) - kcos(L)\big)N}{w} \tag{28}$$

where $s^2 = 1 + h^2 + k^2$, $w = \frac{p}{r} = 1 + fcos(L) + gsin(L)$ and C, S, N are the components of the perturbing acceleration in the directions perpendicular to the radius vector in the direction of motion, along the radius vector outward, and normal to the orbital plane in the direction of the angular momentum vector, respectively.

Gauss' Variational Equations may be used to compute the orbital elements as a function of time for a disturbance acceleration $\mathbf{a}_d$ that is both conservative and nonconservative. The orbital element variations may be integrated by mapping the acceleration vector into the LVLH frame. Additionally, if the disturbance $\mathbf{a}_d$ is due to a control thrust, these equations show the resulting effect on the orbit.

Using the MEEs gives an increase in the domain of convergence, as is seen in Figure (1) as a function of degree and order spherical harmonic gravity.[13] If a single orbit is considered, where one segment for the entire orbit is used, a reduction in both the number of MCPI iterations and correspondingly, the number of gravity function calls is seen. These results are shown in Figures (2) - (4). An initial timing comparison in Matlab, Figure (3), shows that for the single orbit case, using a higher degree and order gravity leads to decreased computation time. These results are obtained using a spherical harmonic gravity Mex file, which is C code compiled to execute as a function in Matlab. The number of gravity function calls (the most computationally expensive portion of the algorithm) are compared over one orbit in Figure (4); the number of calls is correspondingly reduced as a result of reducing the number of MCPI iterations. Further studies using C code with parallel computing on a compute cluster also show decreased computation time for a high degree and order gravity Monte Carlo simulation.

TAYLOR SERIES GRAVITY EXPANSION

A local Taylor Series method is implemented for Monte Carlo simulation. A full spherical harmonic gravity field (40,40) is used for the first MCPI iterations, to converge to the nominal trajectory. Once MCPI converges on this solution, it is stored in memory at every MCPI node for computation of the local Taylor Series approximation near each node. The local Taylor series approximation is justified because all of the terminal iterations are in the close neighborhood of the final converged solution, and we can establish on the basis of prior simulations the maximum distance one must move away from the expansion point to invalidate the local Taylor series approximation to a given accuracy. Thus, the efficiently computable Taylor series gravity approximation compares with known precision to the high-fidelity (spherical harmonic) full gravity solution. The low-fidelity (zonal) gravity is updated using the Taylor series in the following manner, where g_F is the full gravity, g_z is the zonal gravity, $\mathbf{r_1}$ is the position under consideration for the approximating trajectory,

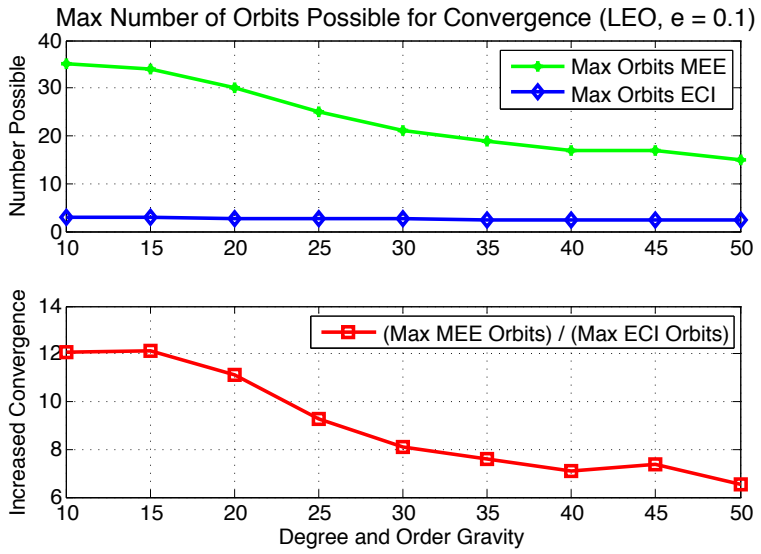

Figure 1. Convergence Improvement (LEO) MEE vs. Cartesian Solutions

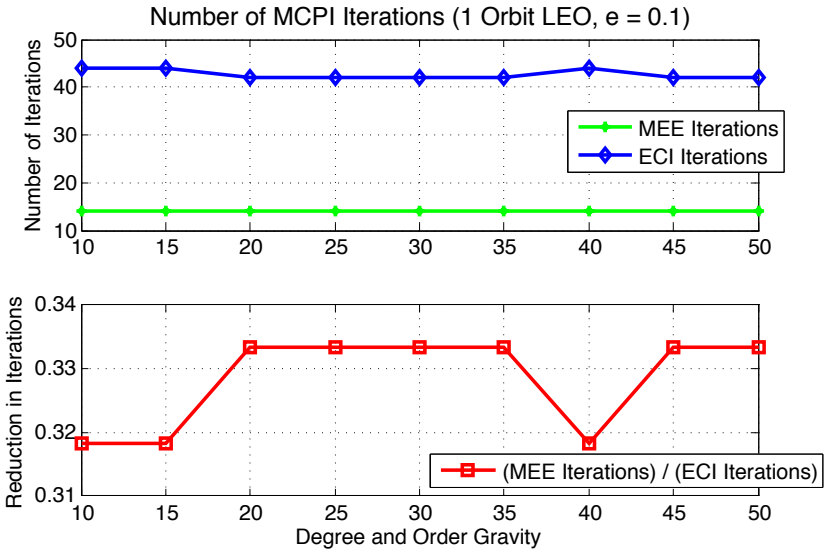

Figure 2. Number of MCPI Iterations Over One Orbit (LEO) MEE vs. Cartesian Solutions

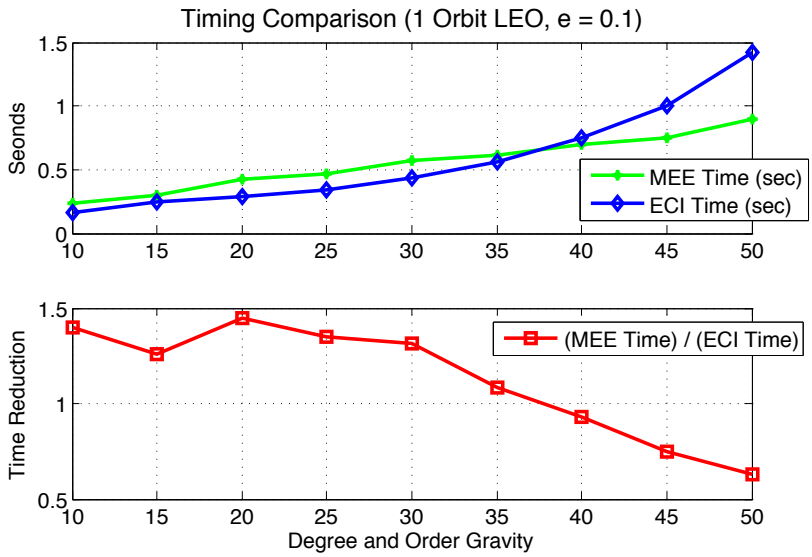

Figure 3. Computation Time Over One Orbit (LEO) MEE vs. Cartesian Solutions

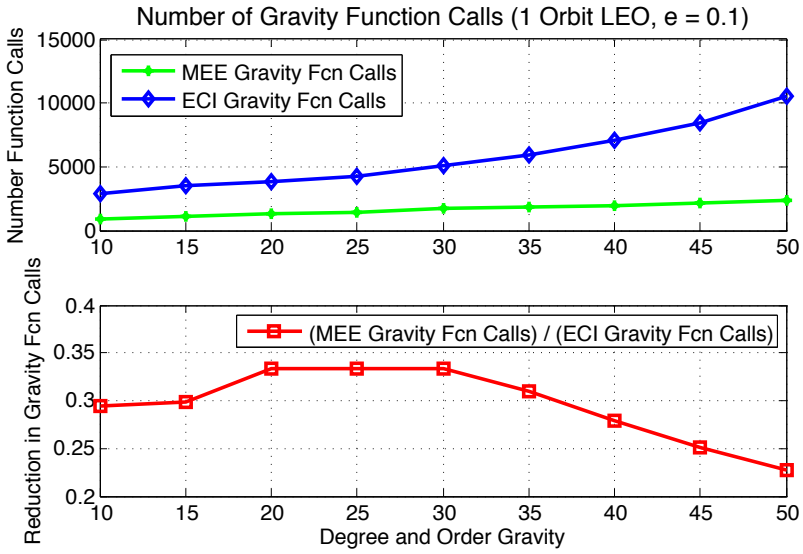

Figure 4. Number of Gravity Function Calls Over One Orbit (LEO) MEE vs. Cartesian Solutions

$\mathbf{r_0}$ is the position (a typical Taylor expansion point) used to compute the nominal trajectory, and ∇ is the gradient function:

$$[\mathbf{g_F}(\mathbf{r_1}) - \mathbf{g_z}(\mathbf{r_1})] \cong [\mathbf{g_F}(\mathbf{r_0}) - \mathbf{g_z}(\mathbf{r_0})] + \nabla[\mathbf{g_F}(\mathbf{r_0}) - \mathbf{g_z}(\mathbf{r_0})][\mathbf{r_1} - \mathbf{r_0}] + \mathbf{H.O.T.} \tag{29}$$

Rearranging this equation and neglecting the higher-order terms provides an approximation of the full gravity that allows for more efficient computation of many Monte Carlo simulations:

$$\mathbf{g_F}(\mathbf{r_1}) = \mathbf{g_z}(\mathbf{r_1}) + \mathbf{c_0} + [\mathbf{A_0}][\mathbf{\Delta r}] \tag{30}$$

where a constant term, plus the gradient times the position difference, are used:

$$c_0 = [\mathbf{g_F}(\mathbf{r_0}) - \mathbf{g_z}(\mathbf{r_0})] \tag{31}$$

$$\mathbf{A_0} = \nabla[\mathbf{g_F}(\mathbf{r_0}) - \mathbf{g_z}(\mathbf{r_0})] \tag{32}$$

$$\nabla\mathbf{r} = [\mathbf{r_1} - \mathbf{r_0}] \tag{33}$$

This method has previously shown to very significantly decrease the computation time of Cartesian MCPI iterations for Monte Carlo simulations.[11] Of significance however, we anticipate the state/co-state differential equation systems (corresponding to optimal transfer formulations based on the MEE variation of parameter differential equations) to converge over much longer time intervals (several orbits, as opposed to a fraction of one orbit for the Cartesian case). This will almost certainly lead to substantial advantages for the case of the MEE formulation for optimal orbit transfers and the generation of extremal field maps for multi-rev transfers. Specifically, MCPI can only solve (without shooting) for optimal transfers over a fraction of an orbit for the Cartesian case, but we anticipate being able to solve optimal transfers for 3 to 5 orbits without a shooting method using MCPI for the MEE state/co-state system of differential equations.

MONTE CARLO SIMULATION

Matlab R2013a

Simulation results are obtained on a Windows 8 machine using Matlab R2013a, where all MCPI results are tuned such that the best performance is achieved while still maintaining an energy check (constant hamiltonian). For the present study, four cases are considered: LEO (e = 0.1), MEO (e = 0.3), GTO (e = 0.6), and HEO (e = 0.7). The results for LEO and MEO are given here. Figures (9) - (12) show the Monte Carlo point clouds for both the MEE solution and the solution using Cowell's method (integration of the Cartesian acceleration form of the equations of motion using ECI coordinates) for the LEO case. The Cartesian and MEE Solutions match to 10 digits of accuracy in the final state point clouds; the overlapping points showing different colors is simply an artifact of the graphics (a 3-D rotation in Matlab shows that they overlap very well). Implementation on a compute cluster further increases the efficiency of this method, as described in the next section.

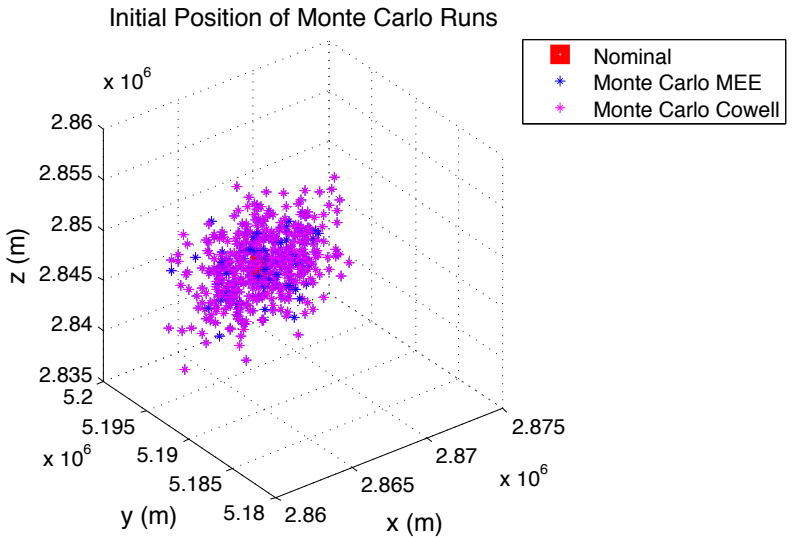

Figure 5. Initial Position Point Cloud for Monte Carlo Simulation (LEO) Comparing MEE Solutions with Cartesian Solutions

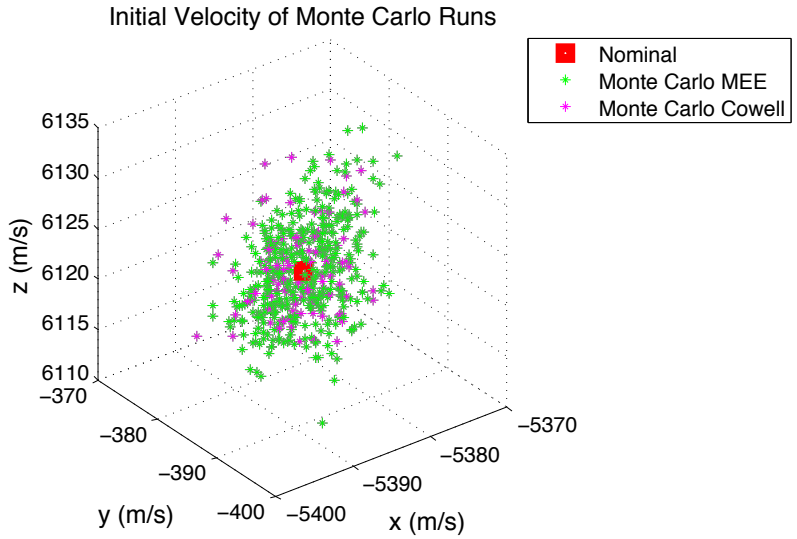

Figure 6. Initial Velocity Point Cloud for Monte Carlo Simulation (LEO) Comparing MEE Solutions with Cartesian Solutions

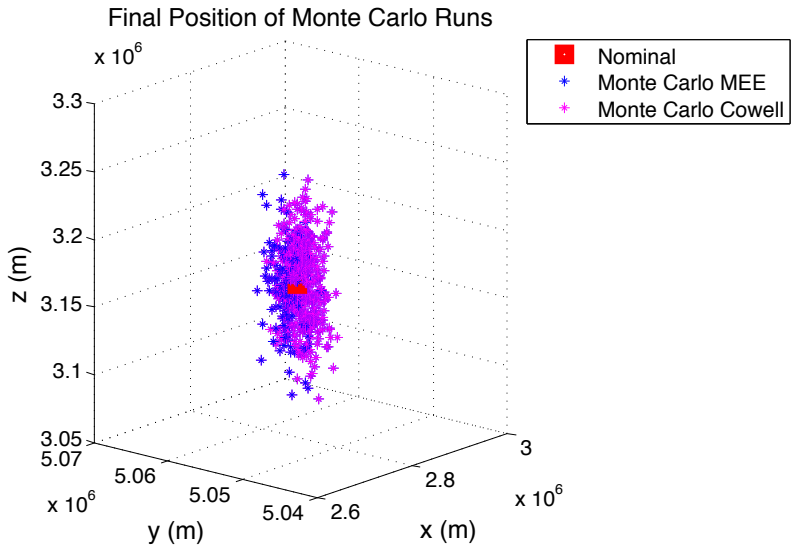

Figure 7. Final Position Point Cloud for Monte Carlo Simulation (LEO) Comparing MEE Solutions with Cartesian Solutions

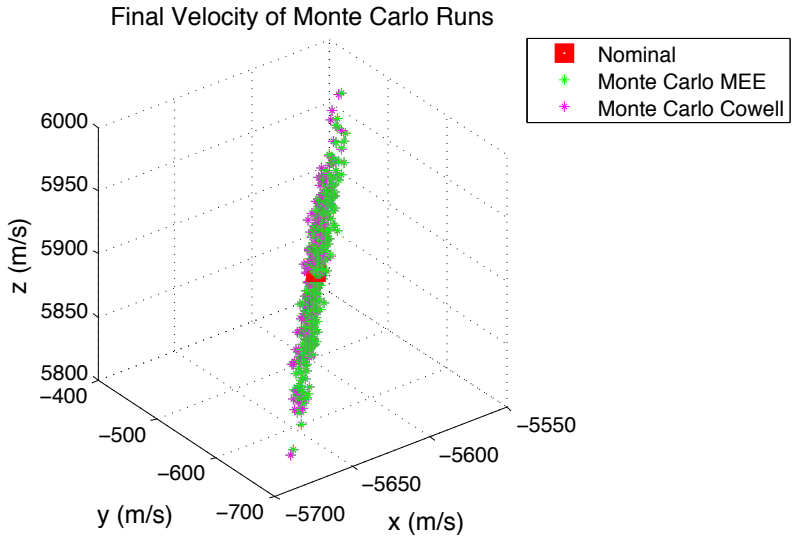

Figure 8. Final Position Point Cloud for Monte Carlo Simulation (LEO) Comparing MEE Solutions with Cartesian Solutions

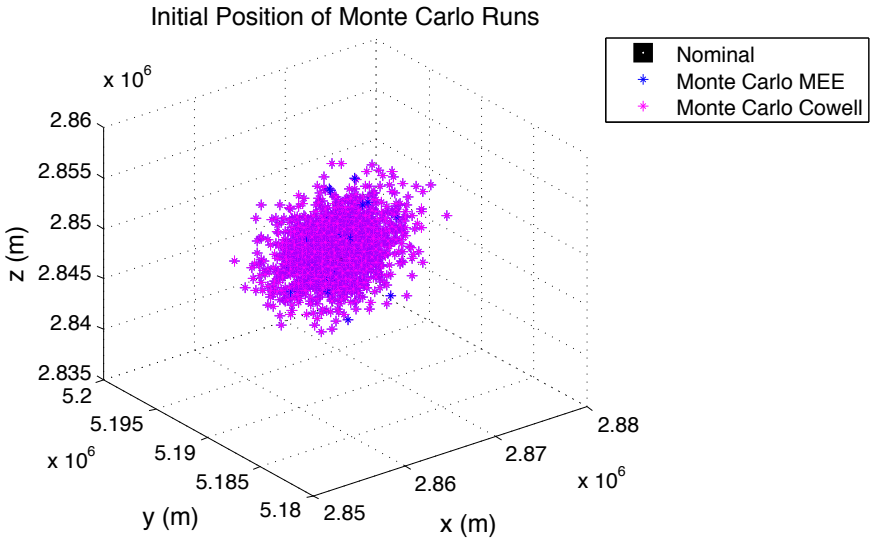

Figure 9. Initial Position Point Cloud for Monte Carlo Simulation (MEO) Comparing MEE Solutions with Cartesian Solutions

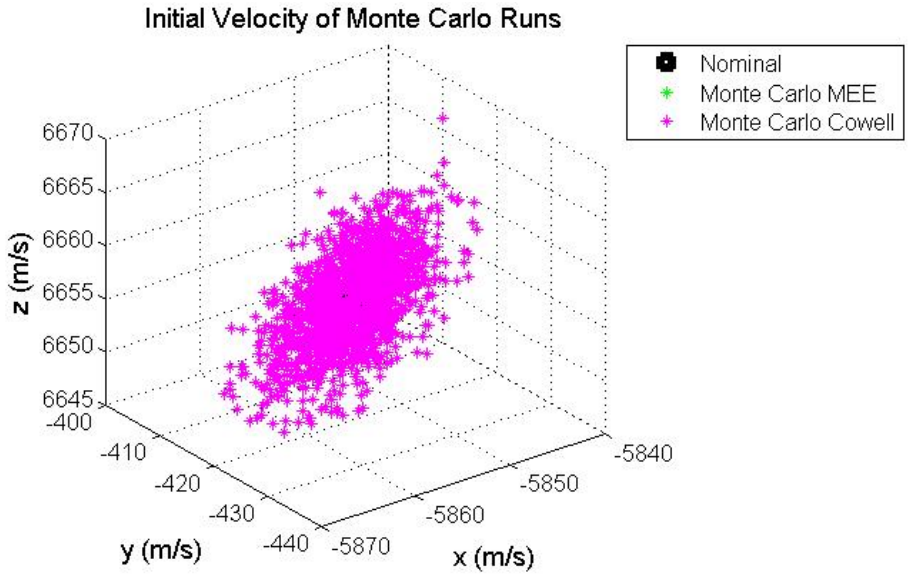

Figure 10. Initial Velocity Point Cloud for Monte Carlo Simulation (MEO) Comparing MEE Solutions with Cartesian Solutions

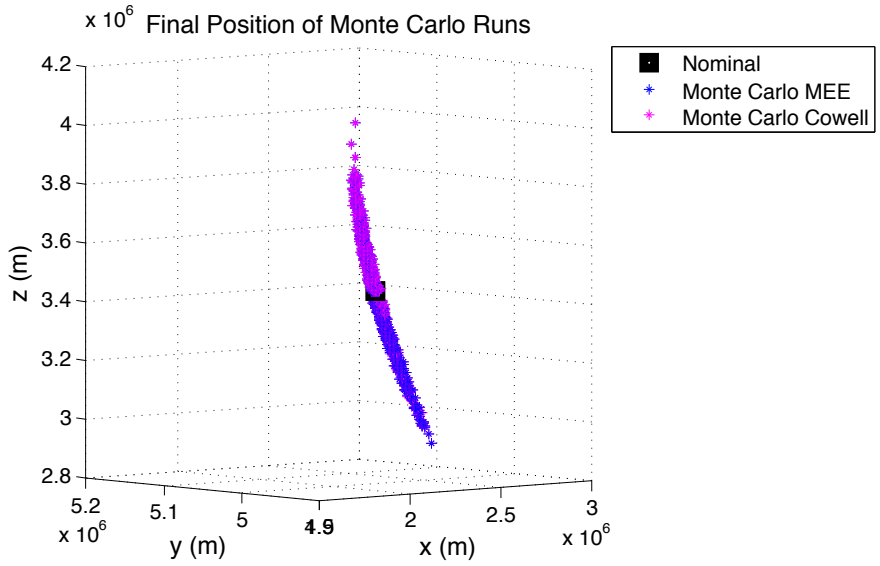

Figure 11. Final Position Point Cloud for Monte Carlo Simulation (MEO) Comparing MEE Solutions with Cartesian Solutions

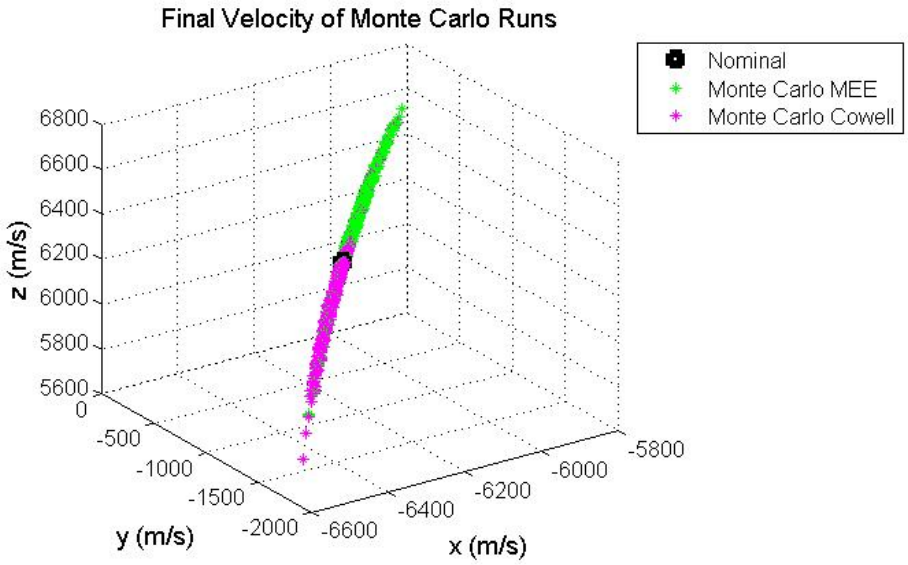

Figure 12. Final Position Point Cloud for Monte Carlo Simulation (MEO) Comparing MEE Solutions with Cartesian Solutions

All Monte Carlo trajectories may be implemented in parallel on the compute cluster, decreasing the computation time.

Figure (13) shows the speedup achieved when using the local Taylor Series gravity model vs. using spherical harmonic gravity for a one-orbit LEO case. An initial study on a compute cluster using only the constant offset term of the Taylor Series expansion, for intermediate MCPI iterations on each neighboring trajectory, shows some speedup as well.

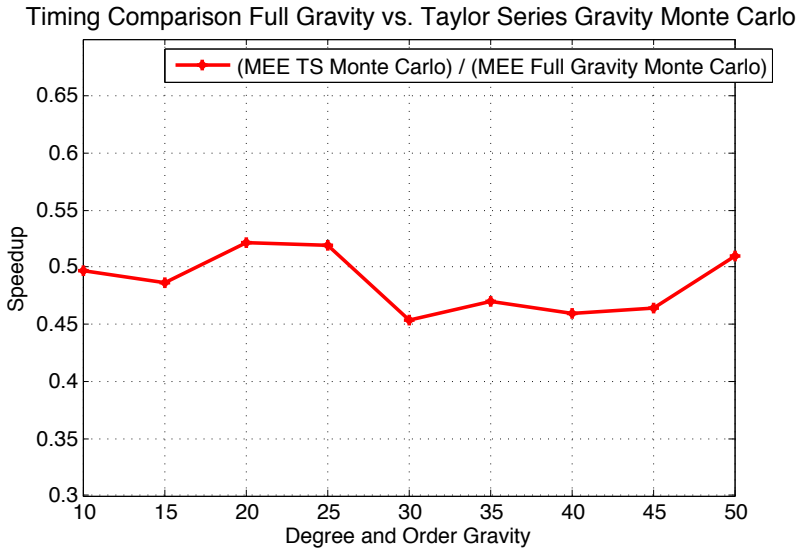

Figure 13. Timing Comparison Over 1 LEO Orbit Using Taylor Series Gravity vs. Spherical Harmonic Gravity

Compute Cluster

The simulation results presented in this section are executed on the Texas A&M University's Land, Air, and Space Robotics (LASR) Laboratory Space Situational Awareness (SSA) Cluster. The LASR SSA Cluster is a 16 node compute cluster dedicated to astrodynamics research. Each of the compute nodes has a pair of Intel Xeon 2.6GHz CPUs and 64GB of RAM. In total the LASR SSA Cluster has 192 cores and a theoretical maximum compute capacity of approximately 1.99 TeraFlops. A Message Passing Inteface (MPI) is used for this configuration.

This simulation takes advantage of existing MCPI optimization schemes such as segmenting one orbit per segment, using radial adaptive gravity, and a version of Taylor Series gravity that incorporates the constant offset term only.[11,23,24] However, this study currently computes the full gravity at least once for each trajectory, rather than just for the nominal trajectory, while incorporating the Taylor Series gravity for intermediate iterations. Future simulations will expand such that the same method is used as in the Matlab studies, to include the second term of the Taylor Series expansion (which includes the Jacobian term) and to use this method for computing trajectories neighboring the nominal. One million MEO trajectories are used for this simulation, where the initial conditions

are specified to be

$$\mathbf{r_0} = [-6365.554 \quad 2087.458 \quad 878.918]\, km \tag{34}$$

$$\mathbf{v_0} = [-1.635 \quad -6.597762 \quad 3.5058499]\, km/s \tag{35}$$

The initial and final position of the trajectories are shown in Figure (14). The total computation time for the Cartesian case is 58,478 seconds, while the total computation time for the MEE Case is 52,969 seconds. When taking into account that the cluster utilizes 192 cores for each simulation, this gives a compuation time of about 5.1 minutes for the Cartesian case and about 4.6 minutes for the MEE case (a 10% reduction in computation time).

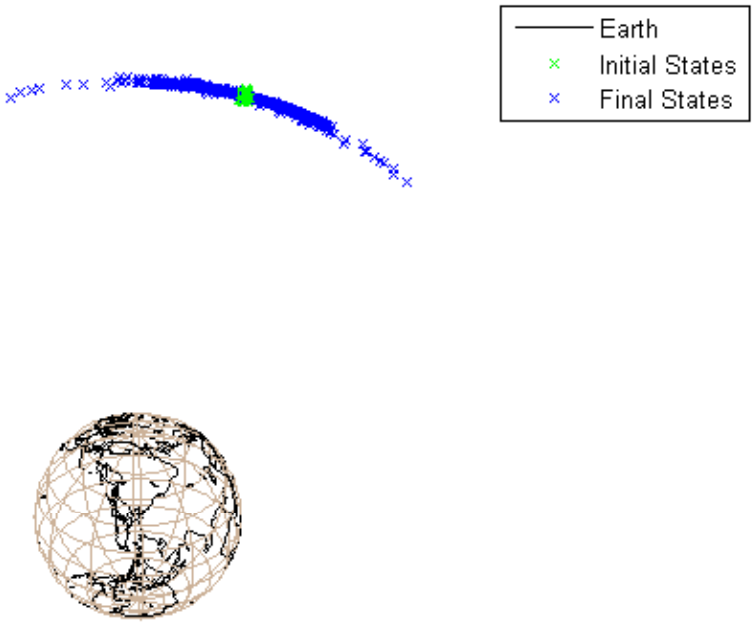

Figure 14. Monte Carlo Using One Million Trajectories in a MEO Orbit

CONCLUSION

Propagation of the Modified Equinoctial Orbital Elements using Modified Chebyshev Picard Iteration allows for an efficient computation of many Monte Carlo simulations, particularly for higher degree and order gravity and when multiple orbits may be used for a single MCPI segment. In this manner, the probability density distribution and its first few moments may be inferred, giving insight into the initial state errors and uncertainty parameters. A local Taylor Series algorithm allows

for decreased computation time for the fully perturbed, spherical harmonic model. Since the local Taylor Series approximation computes neighboring trajectories of the nominal, the approximations may be computed in parallel on a compute cluster, thereby reducing the computation time greatly. These results allow for insight into the uncertainties of the initial conditions. The optimum number of orbits per MCPI segment will almost certainly be a number of orbits less than the maximum number over which MCPI converges for MEE coordinates. We are presently engaged in optimization of the MEE process, and expect the worst case to be a significant speedup compared to the Cartesian case. However, we do not expect order of magnitude speedups will result on a serial machine. These results are expected to show considerable speedup when implemented on a compute cluster.

ACKNOWLEDGMENT

We thank our sponsors: AFOSR (Julie Moses), AFRL (Alok Das, et al), and Applied Defense Solutions (Matt Wilkins) for their support and collaboration through various grants.

Thanks to additional members of the Texas A&M MCPI research team: Robyn Woollands, Abhay Masher, Nathan Budd, Chris Shelton, and Dr. Ahmad Bani Younes.

REFERENCES

[1] X. Bai, "Modified chebyshev-picard iteration methods for solution of initial value and boundary value problems," *PhD Dissertation*, 2010.

[2] C. Clenshaw and H. Norton, "The solution of nonlinear ordinary differential equations in chebyshev series," *The Computer Journal*, Vol. 6, 1963, pp. 88–92.

[3] J. Shaver, "Formulation and Evaluation of Parallel Algorithms for the Orbit Determination Problem," *PhD Thesis*, 1980.

[4] T. Feagin and P. Nacozy, "Formulation and Evaluation of Parallel Algorithms for the Orbit Determination Problem," *Celestial Mechanics and Dynamical Astronomy*, Vol. 29, 1983, pp. 107–115.

[5] T. Fukushima, "Vector Integration of Dynamical Motions by the Picard-Chebyshev Method," *The Astronomical Journal*, Vol. 113, 1997, pp. 2325–2328.

[6] B. Macomber, R. Woollands, A. Probe, A. Bani Younes, X. Bai, and J. Junkins, "Modified Chebyshev Picard Iteration for Efficient Numerical Integration of Ordinary Differential Equations," *Proceedings of 2013 AAS/AIAA GNC Conference, Breckenridge, CO*, 2013.

[7] X. Bai and J. Junkins, "Solving Initial Value Problems by the Picard-Chebyshev Method with Nvidia GPUs," *Proceedings of the 20th Spaceflight Mechanics Meeting, San Diego, CA*, Vol. AAS 10-197, 2010.

[8] D. Koblick, M. Poole, and P. Shankar, "Parallel high-precision orbit propagation using the modified picard-chebyshev method," *ASME International Mechanical Engineering Congress and Exposition*, 2012, pp. 587–605.

[9] X. Bai and J. Junkins, "Modified Chebyshev-Picard Iteration Methods for Orbit Propagation," *Journal of the Astronautical Sciences*, Vol. 4, 2011, pp. 583–613.

[10] D. Kim and J. Junkins, "Multi-Segment Adaptive Modified Chebyshev Picard Iteration Method," *Proceedings of 2014 AAS/AIAA Space Flight Mechanics Conference, Santa Fe, NM*, 2014.

[11] B. Macomber, A. Probe, R. Woollands, and J. L. Junkins, "Parallel Modified-Chebyshev Picard Iteration for Orbit Catalog Propagation and Monte Carlo Analysis," *Proceedings of the 38th Annual AAS/AIAA Guidance and Control Conference, Breckenridge, CO*, 2015.

[12] J. J. Hyun, I. Kwan Park, N. Choe, and M. Choi, "The Comparison of the Classical Keplerian Orbit Elements, Non-Singular Orbital Elements (Equinoctial Elements), and the Cartesian State Variables in Lagrange Planetary Equations with J2 Perturbation: Part 1," *Journal of Astronomy and Space Sciences*, 2011.

[13] J. L. Read, A. B. Younes, and J. Junkins, "Efficient Orbit Propagation of Orbital Elements Using Modified Chebyshev Picard Iteration Method," *Proceedings of the 2015 ICCES Conference, Reno, NV*, 2015.

[14] J. L. Read, R. M. Woollands, A. B. Probe, T. A. Elgohary, A. B. Younes, and J. L. Junkins, "Parallel Computation of Orbital Element Continuous Thrust Maneuvers Using Modified Chebyshev Picard Iteration," *Abstract Submitted to 2016 AIAA SPACE Conference, Long Beach, CA*.

[15] R. Broucke and P. Cefola, "On the Equinoctial Orbit Elements," *Celestial Mechanics*, Vol. 5(3):303-310, 1972.

[16] D. Brouwer and G. Clemence, *Methods of Celestial Mechanics*. Academic Press, 1961.

[17] M. Walker, B. Ireland, and J. Owens, "A Set of Modified Equinoctial Orbit Elements," *Celestial Mechanics*, Vol. 36(4), 1985, pp. 409–419.

[18] M. Walker, B. Ireland, and J. Owens, "ERRATA: A Set of Modified Equinoctial Orbit Elements," *Celestial Mechanics*, Vol. 36(4), 1985, pp. 409–419.

[19] D. A. Danielson, C. Sagovac, B. Neta, and L. Early, *Semianalytic Satellite Theory*. Mathematics Department, Naval Postgraduate School Report. Monterey, CA, Feb 1995.

[20] J. H. Jo, I. K. Park, N. Choe, and M. Choi, "The Comparison of the Classical Keplerian Orbit Elements, Non-Singular Orbital Elements (Equinoctial Elements), and the Cartesian State Variables in Lagrange Planetary Equations with J2 Perturbation: Part 1," *Journal of Astronautical Space Sciences*, Vol. 28(1), 2011, pp. 37–54.

[21] P. Cefola and R. Broucke, "On the Formulation of the Gravitational Potential in Terms of Equinoctial Variables," *AIAA 13th Aerospace Sciences Meeting*, 1975.

[22] E. Roth, "The Gaussian Form of the Variation-of-Parameter Equations Formulated in Equinoctial Elements - Applications: Airdrag and Radiation Pressure," *Acta Astronautica*, Vol. 12, 1985, pp. 719–730.

[23] B. Macomber, A. Probe, R. Woollands, and J. L. Junkins, "Automated Tuning Parameter Selection for Orbit Propagation with Modified Chebyshev Picard Iteration," *Proceedings of AAS/AIAA Space Flight Mechanics Meeting, Williamsburg, VA*, Vol. AAS 15-417, 2015.

[24] A. Probe, B. Macomber, J. Read, R. Woollands, and J. L. Junkins, "Radially Adaptive Evaluation of the Spherical Harmonic Gravity Series for Numerical Orbital Propagation," *Proceedings of AAS/AIAA Space Flight Mechanics Meeting, Williamsburg, VA*, Vol. AAS 15-440, 2015.