Terminal Convergence Approximation Modified Chebyshev Picard Iteration for efficient numerical integration of orbital trajectories

Austin B. Probe
Texas A&M University
Brent Macomber, Donghoon Kim, Robyn M. Woollands, and John L. Junkins
Texas A&M University

ABSTRACT

Modified Chebyshev Picard Iteration (MCPI) is a numerical method for approximating solutions of Ordinary Differential Equations (ODEs) that uses Picard Iteration with orthogonal Chebyshev polynomial basis functions to obtain approximate time histories of system states. Unlike stepping numerical integrators, such as most Runge-Kutta methods, MCPI approximates large segments of the trajectory by evaluating the forcing function at multiple nodes along the current approximation of the trajectory during each iteration. The orthogonality of the basis functions and vector-matrix formulation allow for low overhead cost, efficient iterations, and parallel evaluation of the forcing function. Despite these advantages MCPI only achieves a geometric rate of convergence. This means it can require significantly more function evaluations than other integrators to generate an approximation over a given time span. As the computational complexity of the ODE forcing function increases, it decreases the relative speed of MCPI when compared to other integrators. On the other hand, there are many potential implicit avenues to alleviate these disadvantages.

To overcome the later iterations' geometric convergence, we introduce here the method of Terminal Convergence Approximation Modified Chebyshev Picard Iteration (TCA-MCPI). TCA-MCPI takes advantage of the property that once moderate accuracy has been achieved with the Picard Iteration or with a warm start of the iteration, the spatial deviation of nodes along the segment approaches zero (i.e., the nodes quickly approach fixed points in the force field). Applying well-justified local approximation methods to the forcing function at each node during terminal Picard iterations greatly reduces the number of full function evaluations required to achieve convergence. In many cases the full function evaluations per node necessary to achieve final convergence is reduced to a small single digit number.

One example of the potentially deleterious effect of a complex forcing function on MCPI is the high-order spherical-harmonic gravity models used for high accuracy orbital trajectory generation. When applied to orbital trajectory integration and combined with a starting approximation from the F&G Solution TCA-MCPI outperforms all current state-of-practice integration methods for astrodynamics. This paper presents the development of Terminal Convergence Approximation Modified Chebyshev Picard Iteration, as well as its implementation for orbital trajectory integration using multiple approximation methods. Examples comparing the output, timing, and performance from the TCA-MCPI to state-of-practice numerical integration methods, including Runge-Kutta 7-8, Runge-Kutta-Nystrom 12th-10th, and the 8th order Gauss-Jackson predictor-corrector algorithm, are presented as well.

1. INTRODUCTION

In order to effectively monitor the state of the orbital environment surrounding the Earth and to maintain awareness of potential threats to our space infrastructure, accurate methods for efficient catalog propagation and maintenance are invaluable. These methods are additionally useful in hypothesis-testing for various space situational awareness settings that require many high fidelity orbits to be iterated. Modified Chebyshev Picard Iteration (MCPI) has proven to be an effective method for solving the initial value problem for smooth and continuous Ordinary Differential Equations (ODEs), which is a large class of systems that includes orbital propagation. MCPI is a technique for numerical quadrature for ODEs that uses a trajectory approximation, generated from a set of high-order orthogonal Chebyshev polynomials, and recursively refines it using Picard iteration. The second order vector matrix implementation of the MCPI algorithm, shown in Figure 1, consists of two major stages: initialization and iteration. The initialization stage includes the determination of the time span and number of function evaluation nodes for the trajectory segment approximation, the creation of certain constant matrices required for iteration, the necessary time transformation, and the generation of an initial trajectory guess. The iteration stage evaluates the

forcing function at each of the nodes along the trajectory, and improves upon the trajectory approximation with an update of the velocity and subsequent update of the position. The algorithm then repeats the iteration phase until either the accuracy requirement or iteration limit is met. This formulation allows for low overhead and efficient iterations while also allowing for massive parallelization because all functions evaluations can be completed independently and simultaneously. [1][2][3]

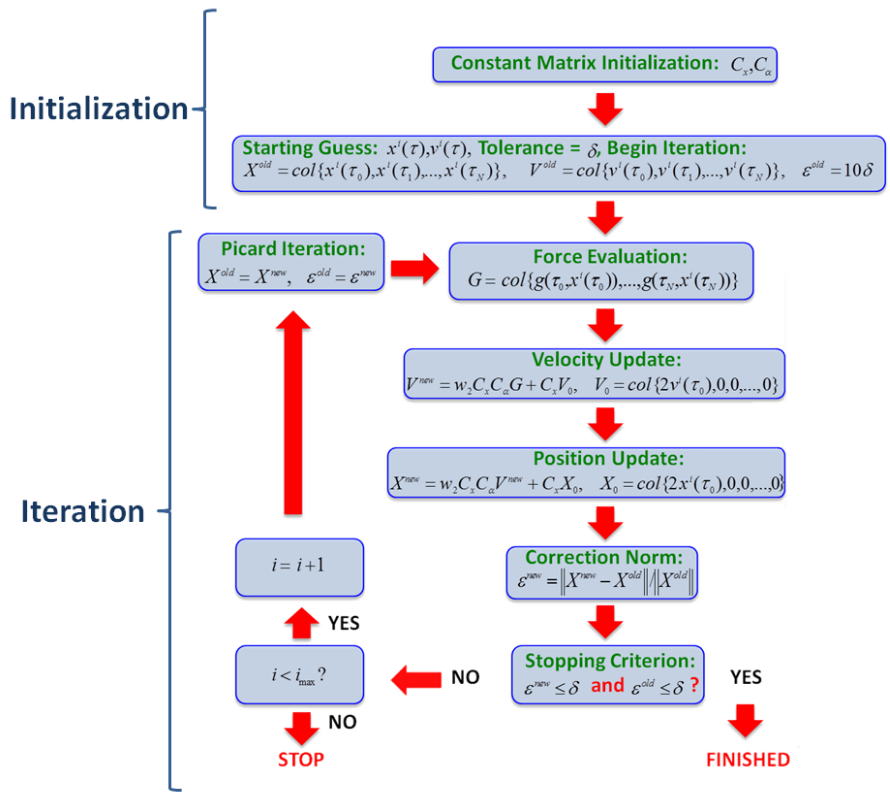

Figure 1: Second Order Cascade MCPI Algorithm for Solving Initial Value Problems

As mentioned, one limitation of MCPI is that it only achieves a geometric rate of convergence with typically up to one order of magnitude reduction in the solution approximation error achieved on each terminal iteration. As a result it can require a significant number of iterations to converge when compared to other methods. This can be a significant issue for high-accuracy orbital propagation because the spherical harmonic gravity function is computationally expensive. Terminal Convergence Approximation Modified Chebyshev Picard Iteration (TCA-MCPI) a modification to the original form of MCPI introduces a method of dramatically reducing the number of full force function evaluations to increase computational efficiency without adversely effecting final accuracy.

2. ORBITAL TRAJECTORY SEGMENTATION AND NODAL PLACEMENT

As part of the initialization of MCPI the time span for integration and the number of evaluation nodes must be selected. In contrast to stepping integrators or some implicit integrators that consider relatively short time segments, MCPI considers a large segment of a trajectory simultaneously; as a result the traditional methods for variable step size determination do not translate well for use with in MCPI. Additionally, traditional methods do not provide much insight into the number of evaluation nodes that should be used for each segment. While general methods for time span determination and nodal density selection are presently under development, for applications to a specific problem it is possible to use heuristic methods and physical insight to generate a segment setup (number of segments, node locations) that leads to efficient solutions to that class of problems.

A study of the accuracy performance of MCPI with varying time spans and nodal density was completed on a characteristic set of orbits with varying semi-major axis and eccentricity to establish a general approximate method

for segment setup for efficient and accurate orbital propagation. A periodic scheme that repeats for each orbit was adopted ("orbit completion" is defined as successive passages through perigee, the smallest radius on a generally perturbed orbit). The difficulty of the numerical quadrature for any method for non-maneuvering satellites is greatest surrounding the perigee of an orbit and simplest at apogee, therefore smaller timespans and/or greater nodal density are required there to achieve a constant level of accuracy throughout the orbit. An odd number of segments (generally three) with time spans and node counts that are symmetric with respect to perigee was selected for simplicity and to reflect evident physical truths approximated by Kepler's second law of motion. An example of this segmentation setup method for a highly eccentric (e~0.7) LEO orbit is shown in Figure 2. The different colors represent the three segments used to integrate the orbits and show the node distribution for each segment.

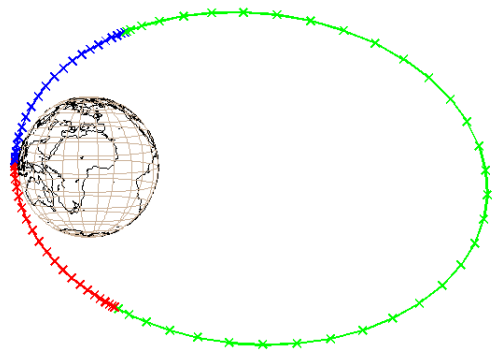

Figure 2: Example MCPI Segmentation Scheme

3. INITIAL ORBITAL TRAJECTORY APPROXIMATION

For successive iterations, MCPI uses a current approximation of a trajectory to generate an improved approximation. If a sufficiently accurate initial guess for the path being approximated can be provided analytically, it is possible to avoid the most slowly converging initial iterations that MCPI would require generating an approximation as accurate as the analytical initial guess. This process is known as providing MCPI with a "warm-start". To accelerate the computation of orbital trajectories, Battin's analytical two-body solution (exact solution for the F&G functions) to the two-body problem is used to generate the initial approximation. [4] Using the actual two-body trajectory as "warm-start" for the full perturbed trajectory puts the initial trajectory approximation in the general proximity of the true solution and allows for some of the iterations that would otherwise be required to be skipped altogether. It is also possible to invoke analytical perturbation theories to account for the zonal harmonics and approximately for drag. [5]

4. TERMINAL CONVERGENCE APPROXIMATION

Since the nodes are known to approach fixed points in the force field during terminal convergence, it is possible to avoid the high number of the function evaluations by judiciously replacing the full forcing function evaluations with a local force approximation. This is effective because MCPI is repeatedly evaluating the forcing function at nodes with the same spacing in τ. As the approximation of the trajectory approaches the true solution, the variation of the node locations in physical space approaches zero; this means that as the function accuracy requirements for the improving the approximation of the trajectory increase, so will the accuracy of any type of local force function quasi-linear approximation. It is possible to apply this principle in the case of a general forcing function using partial derivatives, Taylor Series techniques, or any other method of reliable approximation. Specifically for orbital trajectory propagation, it is possible to determine a computationally simple correction at each node of some a priori approximate gravity model that locally replaces the full-fidelity force model to high precision.

In order for the substitution of approximate function evaluations to be an effective, a metric for determining when the use of the approximation will accelerate convergence is needed. If an inaccurate local force function is employed, it can cause MCPI to converge to an erroneous trajectory, and if full function evaluations are used when

they are not required then it will be unnecessarily computationally expensive. To provide insight to these issues, we define the quantity level of approximate convergence as the ratio of the correction norm after the most recent full function evaluation to the current correction norm. We have found that comparing the reliable level of accuracy of the approximation used to the level of approximate convergence can be utilized as an effective switching metric. When the accuracy of the approximation is greater than the level of approximate convergence from the last full evaluation then it is safe to perform an approximate evaluation; otherwise, a full evaluation is required. Additionally, in the case that the chosen approximation can be used without a full function evaluation, initial approximate evaluations can be used until the current correction norm exceeds the accuracy of the approximation. These qualitative ideas apply to any local force approximation approach.

The key to utilizing terminal convergence approximation for orbital propagation is approximating the EGM2008 spherical harmonic gravity function using the analytical calculation for the two-body acceleration with only the J2-J6 perturbations to achieve approximate initial convergence. This force model can reliably predict at least 4 significant figures of the EGM2008 gravity model, so that was the level of accuracy used for comparison to level of approximate convergence. [6] While the correction norm is above that level of physical accuracy of the two-body plus J2-J6 approximation provides the entire force evaluation. Once that level of correction norm is exceeded a full EGM2008 gravity evaluation is performed. At this point, the "local offset" difference between the two-body plus J2-J6 approximation and the full evaluation is recorded (Equations 1-2). The local offsets are the local summation of all higher order gravity effects at that point. Since the shortest high frequency gravity wavelength is in 10s of km and since the local convergence errors are typically a fraction of 1 km, the offset is slowly varying spatially relative to convergence errors. This local offset, at each node, is treated as a constant perturbation imposed upon the approximation in subsequent approximate evaluations, as shown in Equation 3. A flow chart representation of the terminal convergence approximation algorithm that would replace the "Function Evaluation" step within the standard MCPI implementation (Figure 1) is shown in Figure 3. This method of function approximation is similar to one developed independently for use in the Bandlimited implicit Runge–Kutta method, with the key distinction being that the present approach utilizes accuracy based metrics for selection of whether or not to utilize the approximate model, and also to tailor adaptively the local convergence tolerances consistent with the physical accuracy of the force approximation. [7]

$$g_{full}(\tau, x(\tau)) = EGM2008(x(\tau)) \tag{1}$$

$$\Delta g(\tau) = g(\tau, x(\tau)) - J2J6(x(\tau)) \tag{2}$$

$$g(\tau, x(\tau)) = J2J6(x(\tau)) - \Delta g(\tau) \tag{3}$$

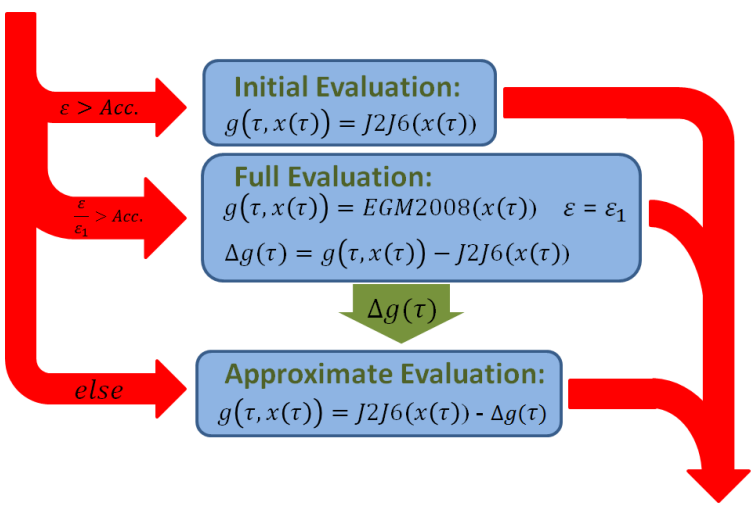

Figure 3: Gravity Function Evaluation Flow Chart

5. RESULTS

Comparing TCA-MCPI to the standard MCPI algorithm shows that introducing the approximation results in a major reduction in the required number of full function evaluations and in a minor penalty in terms of the number of iterations needed. This translates to major savings in terms of computational effort and run times, because extra iterations with the approximation are substantially less computationally expensive than a full evaluation.

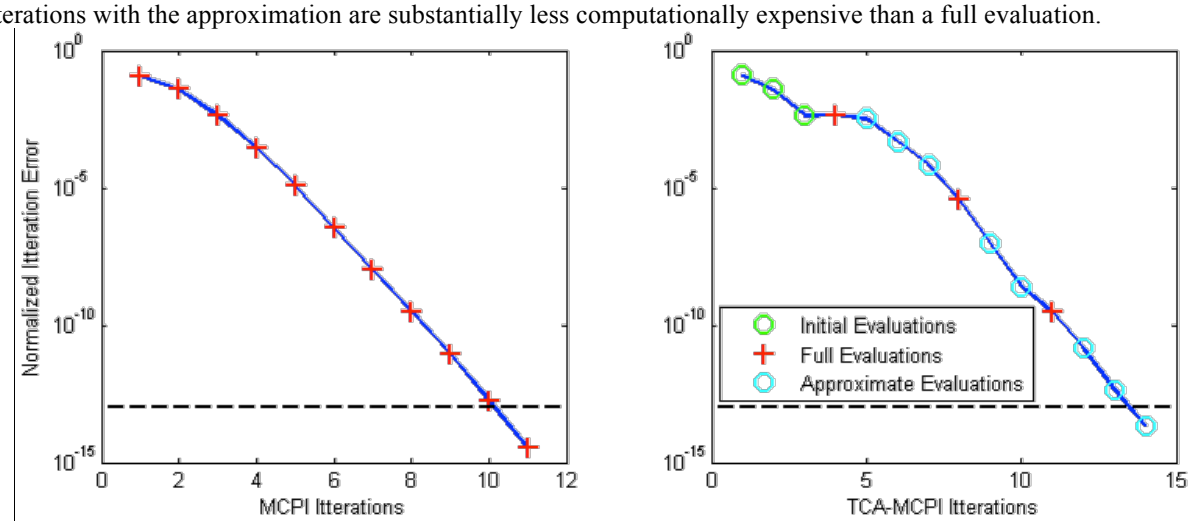

Figure 4: MCPI and TCA-MCPI Convergence Progress vs. Iteration Number

A study examining the performance of TCA-MCPI to other state of the practice integrators on various characteristic orbits was completed. The integrators considered were Runge-Kutta 4th-5th Order, Runge-Kutta 7th-8th Order, Runge-Kutta-Nystrom 12th-10th Order, Gauss-Jackson 8th Order, and finally the original version of MCPI without the implementation of terminal convergence approximation. Each integrator was used to propagate the same set of six orbits; a circular, moderately eccentric, and highly eccentric at perigee altitudes of both a Low Earth Orbit and Geosynchronous Orbit. Figure 5 illustrates the set of LEO altitude orbits.

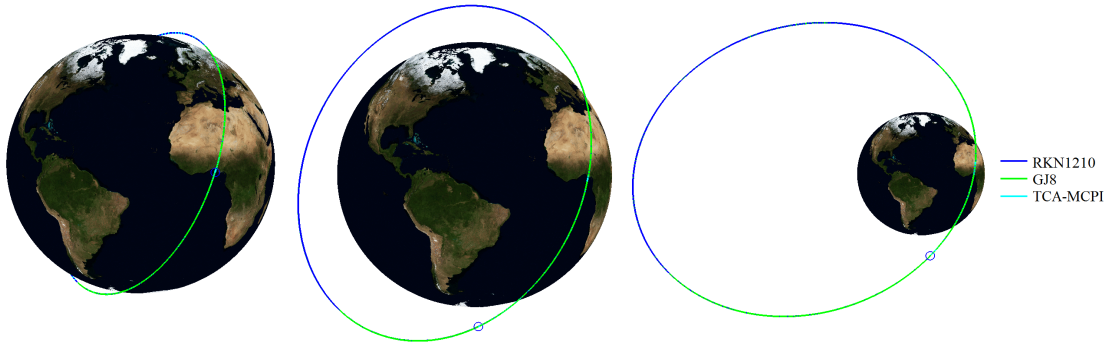

Figure 5: Circular, Moderately Eccentric and Highly Eccentric Orbital Trajectories with a LEO Perigee Altitude

Table 1 presents the orbital elements for each case, as well as the orbital period. Figures 6-17 provide a comparison of the computation time for one orbital period, the function evaluations required for each integrator, and how accurately each integrator preserves the relative Hamiltonian. In the case of Gauss-Jackson there are two function evaluations values reported, initial function evaluations, and the total function evaluations. The initial evaluations are required as part of the setup for the integrator, but are not required for evaluation after initialization; for the integration of subsequent orbits the number of required valuations would be the total number of evaluations minus the initial evaluations. [8] [9] [10][11] This study was run using Matlab™ 2013a with a custom EGM2008 Spherical Harmonic Gravity used as the only forcing function. [10]

Table 1: Test Case Orbit Elements and Period

	a	*e*	*i*	*M*	Ω	ω	*T (s)*
Case LC	6.7087 E6	0.0027	1.1866	1.5735	1.6057	4.8022	5.4905 E3
Case LME	8.4920 E6	0.21	1.1866	0.0154	1.6057	0.077	7.7883 E3
Case LHE	21.641 E6	0.69	1.1866	0.0066	1.6057	0.0859	31.683 E4
Case GC	42.164 E6	0	0	0	0	0	86.164 E3
Case GME	53.372 E6	0.21	0	0	0	0	122.271 E3
Case GHE	136.01 E9	0.69	0	0	0	0	499.21 E3

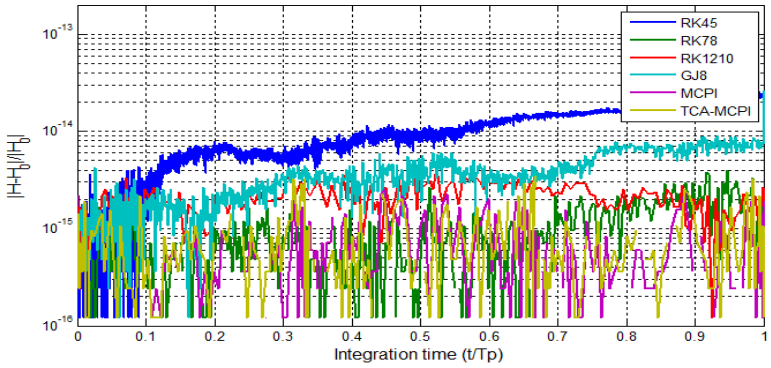

Figure 6: Case LC Relative Hamiltonian Preservation

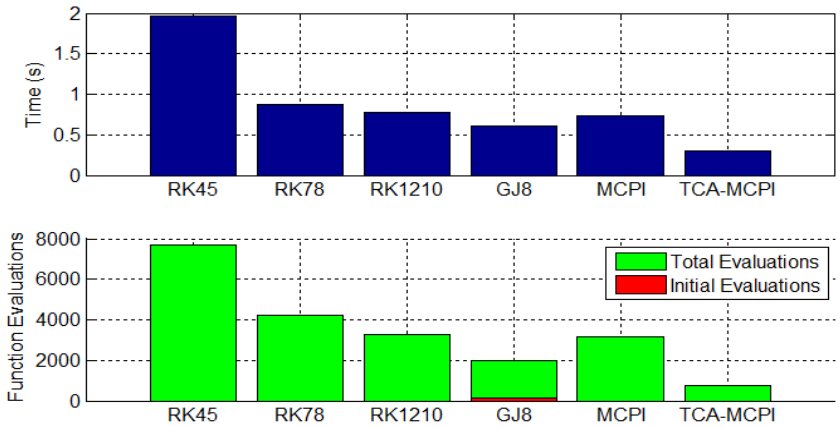

Figure 7: Case LC Timing and Function Evaluation Results

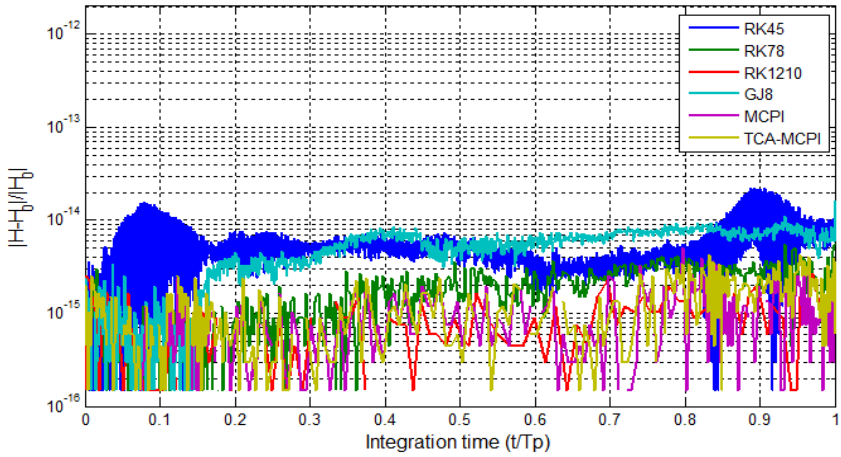

Figure 8: Case LME Relative Hamiltonian Preservation

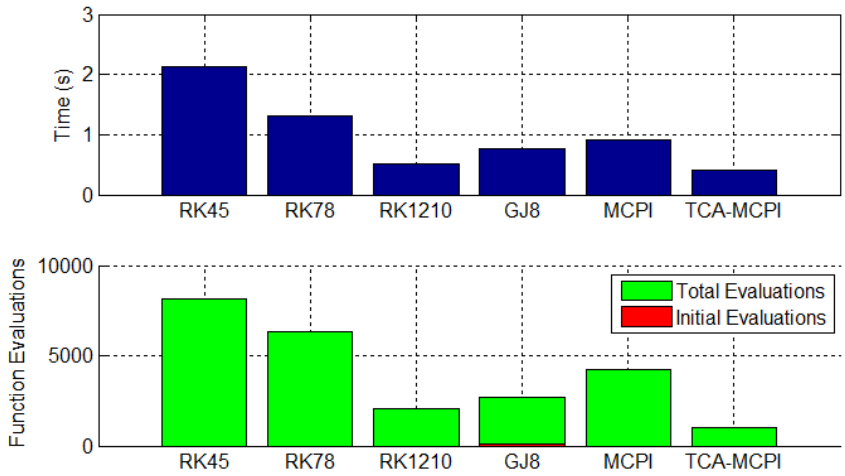

Figure 9: Case LME Timing and Function Evaluation Results

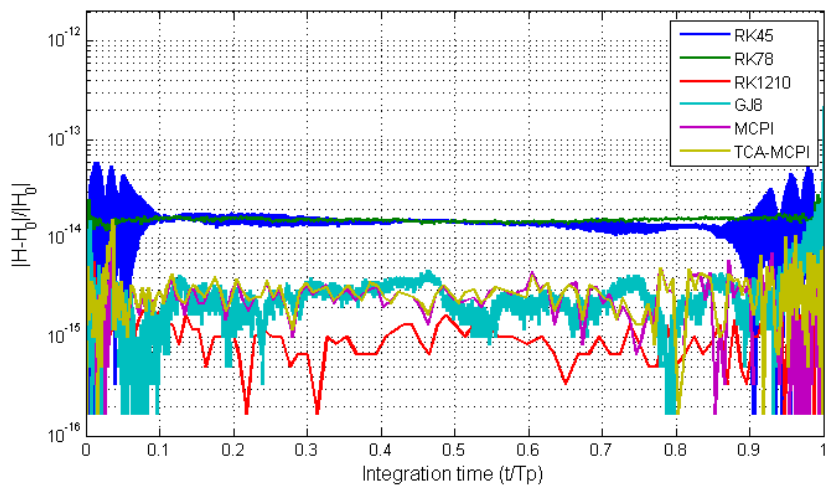

Figure 10: Case LHE Relative Hamiltonian Preservation

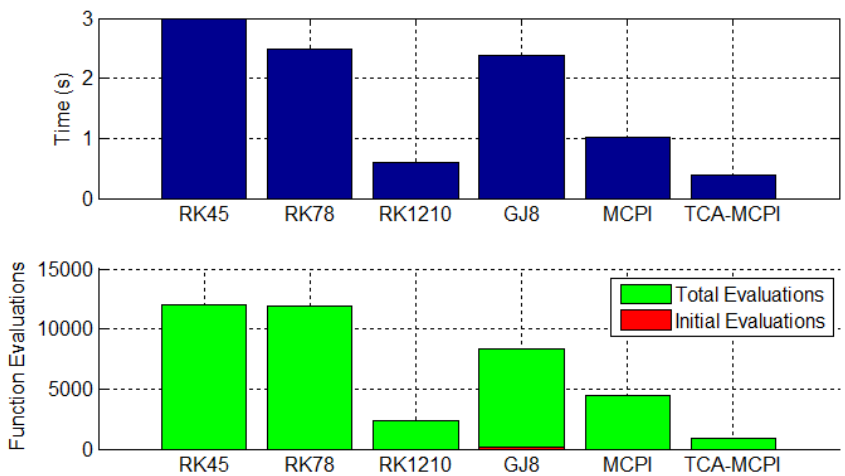

Figure 11: Case LHE Timing and Function Evaluation Results

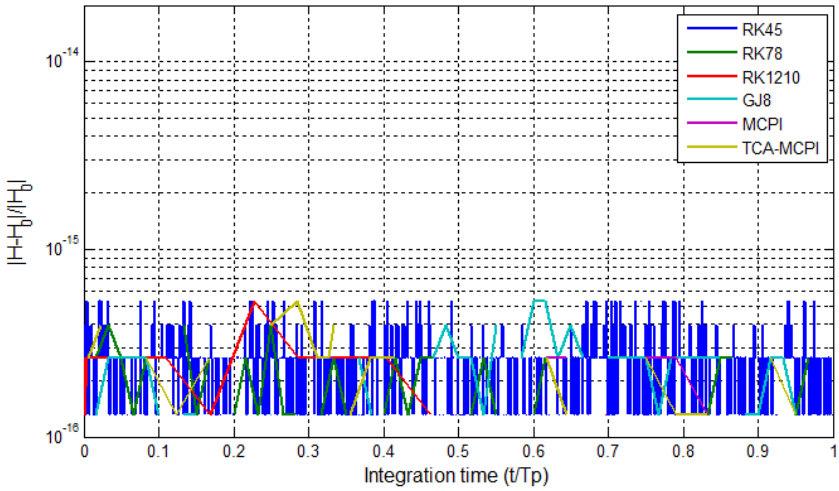

Figure 12: Case GC Relative Hamiltonian Preservation

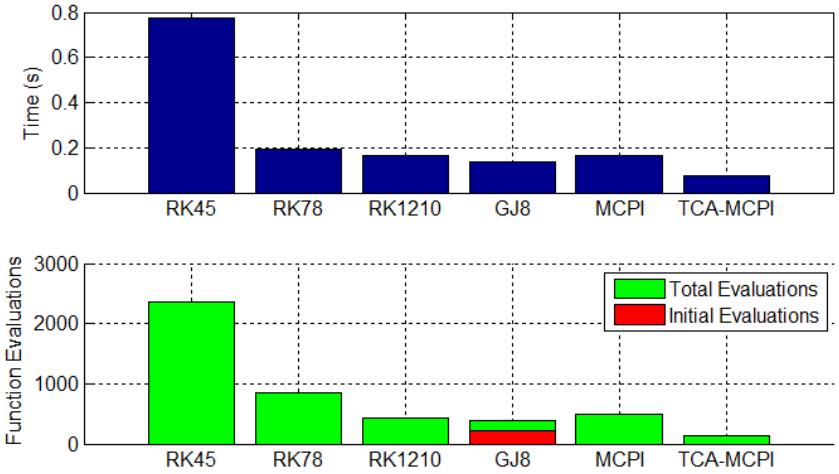

Figure 13: Case GC Timing and Function Evaluation Results

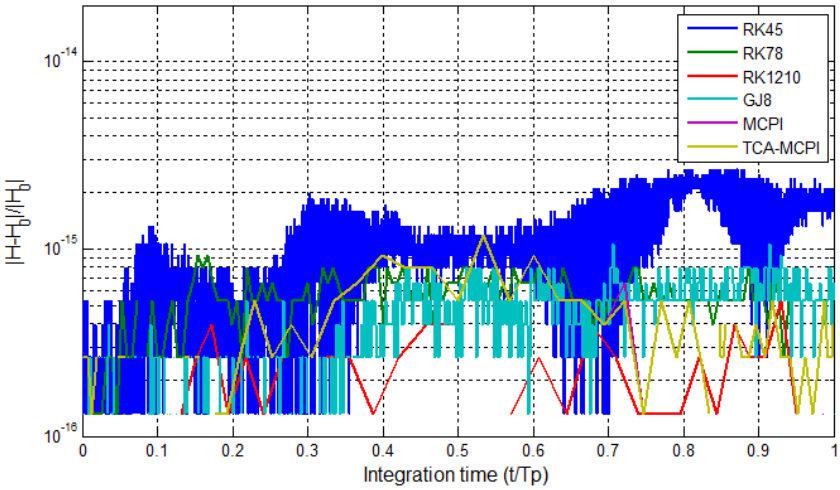

Figure 14: Case GME Relative Hamiltonian Preservation

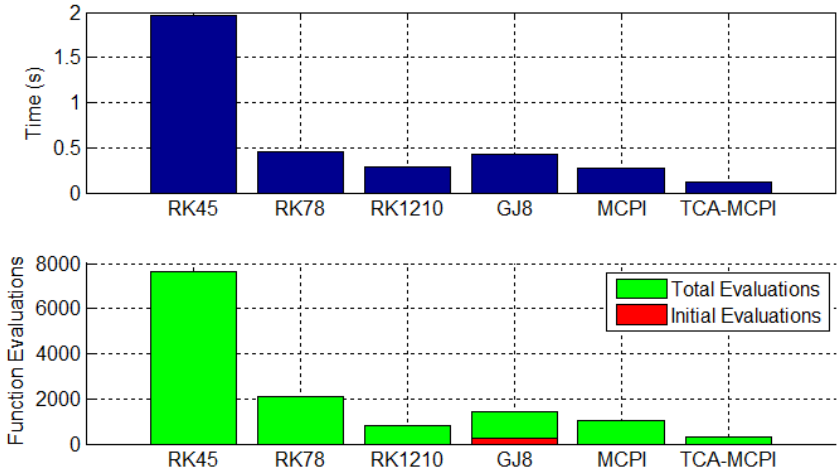

Figure 15: Case GME Timing and Function Evaluation Results

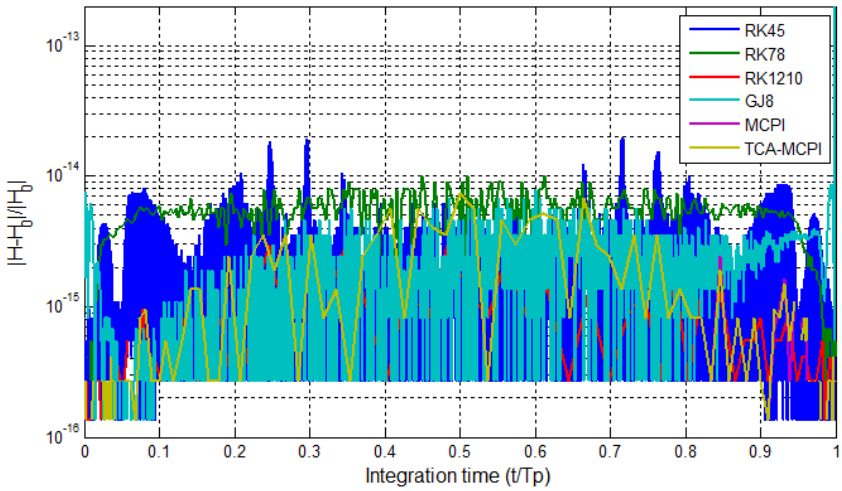

Figure 16: Case GHE Relative Hamiltonian Preservation

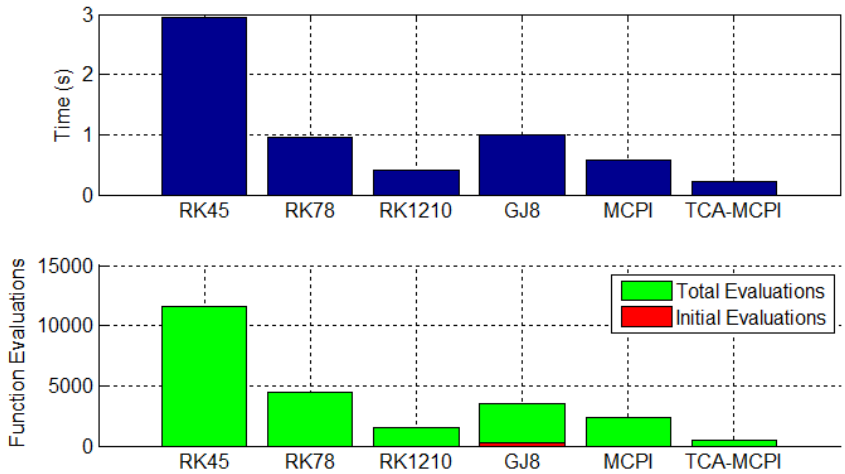

Figure 17: Case GHE Timing and Function Evaluation Results

6. CONCLUSIONS

Terminal Convergence Approximation Modified Chebyshev Picard Iteration shows the best performance in terms of timing and function evaluations in all cases, even requiring fewer evaluations than the repeating evaluations for the "industry-standard" Gauss-Jackson in circular LEO cases. In the circular orbit cases Gauss-Jackson has the second best performance, while the Runge-Kutta-Nystrom 12th-10th order and occasionally Runge-Kutta 7th-8th order perform better than Gauss-Jackson in the eccentric cases (as is well known). Runge-Kutta 4th-5th is generally the poorest performer in all cases. The single orbit case considered here illustrates the worst case for results for timing of Gauss-Jackson and the two MCPI variants because these methods require initialization to start performing integration. As the total number of orbits considered increases the fraction of the total time that these initializations represent will diminish. Of the methods tested in a serial mode in this paper, only MCPI parallelizes efficiently for each orbit, and this opens up substantial further speedups. Finally, TCA-MCPI provides a reliable method of using approximation methods to effectively circumvent many of the computationally expensive function evaluations required for standard MCPI and other methods based on Picard Iteration without negatively affecting the accuracy of the final approximation.

7. ACKNOWLEDGEMENTS

We thank our sponsors: AFOSR (Julie Moses), AFRL (Alok Das, et al), and Applied Defense Solutions (Matt Wilkins) for their support and collaborations under various contracts and grants. This work was completed at and with support from the Texas A&M University Aerospace Engineering Department.

8. REFERENCES

[1]. Bai, X., and Junkins, J.L., Modified Chebyshev-Picard Iteration Methods for Solution of Initial Value Problems, Advances in the Astronautical Sciences, vol. 139, pp. 345–362, 2011.

[2]. Bai, X. and Junkins, J.L., Modified Chebyshev Picard Iteration Methods for Station-Keeping of Translunar Halo Orbits, Mathematical Problems in Engineering, vol. 2012, Article ID 926158, 2012.

[3]. Bai, X. Modified chebyshev-picard iteration methods for solution of initial value and boundary value problems, Ph.D. dissertation, Texas A&M Univ, College Station, TX, 2010.

[4]. Battin, R. H., Mathematics and Methods of Astrodynamics, AIAA Education Series, 1999

[5]. Parks, A. D., A drag-augmented Brouwer-Lyddane artificial satellite theory and its application to long-term station alert predictions, Naval Surface Weapons Center Technical Report 83-107 (NSWC TR 83-107), April 1983

[6]. Schaub, H., and Junkins, J.L., Analytical Dynamics of Aerospace Systems, 3rd ed, AIAA Education Series, 2014

[7]. Bradley B.K., Jones, B.A., Beylkin, G., Sandberg, K., Axelrad**, P.,** Bandlimited implicit Runge–Kutta integration for Astrodynamics, Celestial Mechanics and Dynamical Astronomy **vol.**119, Issue 2 , pp 143-168

[8]. Berry, M. M., Healy, L. M., J of the Astronautical Sciences, Vol. 52, No. 3, July–September 2004, pp. 331 – 357

[9]. Compere M., ODE Solvers for Octave and Matlab, sites.google.com/site/comperem/home/ode_solvers, 9/1/2014

[10]. MATLAB 2013a, The MathWorks, Inc., Natick, Massachusetts, United States.

[11]. Oldenhuis, R., RKN1210 12th/10th order Runge-Kutta-Nyström integrator, http://www.mathworks.com/matlabcentral/fileexchange/25291- rkn1210-a-12th-10th-order-runge-kutta-nystr%C3%B6m-integrator, 9/1/2014