



AFRL-OSR-VA-TR-2015-0203

Picard Trajectory Approximation Iteration for Efficient Orbit Propagation

John Junkins
TEXAS ENGINEERING EXPERIMENT STATION COLLEGE STATION

07/21/2015
Final Report

DISTRIBUTION A: Distribution approved for public release.

Air Force Research Laboratory
AF Office Of Scientific Research (AFOSR)/ RTB
Arlington, Virginia 22203
Air Force Materiel Command

| REPORT DOCUMENTATION PAGE | | | Form Approved OMB No. 0704-0188 | | | | | | | | |
|--|-------------|--|--|---|--|---|---|---|---|----------------------------------|--|
| <p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Executive Services, Directorate (0704-0188). Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.</p> | | | | | | | | | | | |
| 1. REPORT DATE (DD-MM-YYYY) 29-07-2015 | | 2. REPORT TYPE Final Performance | | 3. DATES COVERED (From - To) 15-09-2011 to 14-03-2015 | | | | | | | |
| 4. TITLE AND SUBTITLE Picard Trajectory Approximation Iteration for Efficient Orbit Propagation | | | 5a. CONTRACT NUMBER | | | | | | | | |
| | | | 5b. GRANT NUMBER FA9550-11-1-0279 | | | | | | | | |
| | | | 5c. PROGRAM ELEMENT NUMBER | | | | | | | | |
| 6. AUTHOR(S) John Junkins | | | 5d. PROJECT NUMBER | | | | | | | | |
| | | | 5e. TASK NUMBER | | | | | | | | |
| | | | 5f. WORK UNIT NUMBER | | | | | | | | |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) TEXAS ENGINEERING EXPERIMENT STATION COLLEGE STATION 1470 WILLIAM D FITCH PKY COLLEGE STATION, TX 77843-3577 US | | | 8. PERFORMING ORGANIZATION REPORT NUMBER | | | | | | | | |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AF Office of Scientific Research 875 N. Randolph St. Room 3112 Arlington, VA 22203 | | | 10. SPONSOR/MONITOR'S ACRONYM(S) AFOSR | | | | | | | | |
| | | | 11. SPONSOR/MONITOR'S REPORT NUMBER(S) | | | | | | | | |
| 12. DISTRIBUTION/AVAILABILITY STATEMENT A DISTRIBUTION UNLIMITED: PB Public Release | | | | | | | | | | | |
| 13. SUPPLEMENTARY NOTES | | | | | | | | | | | |
| 14. ABSTRACT This report covers the third year of the contract AFOSR Contract FA9550-11-1-0279 which has involved the effort of the principal investigator, John L. Junkins, and 5 PhD students. One of these students completed his dissertation (Dong Hoon Kim) and another (Brent Macomber) is nearing completion. The other three will complete their work near the end of calendar year 2015. The project addresses a problem near the heart of space situational awareness (SSA), namely efficient and accurate propagation of orbits. This is a classical problem whose importance has been dramatically elevated by the growth of orbital debris population, and by several events involving deliberate and accidental collisions of spacecraft in low earth orbit, see Figure 1. There are numerous challenges: | | | | | | | | | | | |
| 15. SUBJECT TERMS orbit propagation | | | | | | | | | | | |
| 16. SECURITY CLASSIFICATION OF: <table border="1" style="width: 100%;"> <tr> <th>a. REPORT</th> <th>b. ABSTRACT</th> <th>c. THIS PAGE</th> </tr> <tr> <td>U</td> <td>U</td> <td>U</td> </tr> </table> | | | a. REPORT | b. ABSTRACT | c. THIS PAGE | U | U | U | 17. LIMITATION OF ABSTRACT UU | 18. NUMBER OF PAGES UU | 19a. NAME OF RESPONSIBLE PERSON John Junkins |
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | | | | | | | |
| U | U | U | | | | | | | | | |
| | | | | | 19b. TELEPHONE NUMBER (Include area code) 979-845-3912 | | | | | | |

Picard Path Approximation Methods for Orbit Propagation

John L. Junkins
Texas A&M University
October 6, 2014

This report covers the third year of the contract AFOSR Contract FA9550-11-1-0279 which has involved the effort of the principal investigator, John L. Junkins, and 5 PhD students. One of these students completed his dissertation (Dong Hoon Kim) and another (Brent Macomber) is nearing completion. The other three will complete their work near the end of calendar year 2015.

The project addresses a problem near the heart of space situational awareness (SSA), namely efficient and accurate propagation of orbits. This is a classical problem whose importance has been dramatically elevated by the growth of orbital debris population, and by several events involving deliberate and accidental collisions of spacecraft in low earth orbit, see Figure 1. There are numerous challenges:

- Space object catalog updates, requiring precision propagation of $\sim 20,000$ objects orbits,
- Conjunction analysis, probability of collision, and collision avoidance.
- Processing of nightly observables to identify and characterize existing and new objects, requires testing of $\sim 10^6$ orbit propagation hypothesis and hours of high performance CPU time for per day.

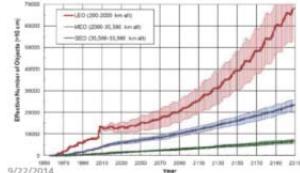
It is remarkable that this classical problem can be accelerated by over an order of magnitude in serial algorithms and over three orders of magnitude in parallel computation using methods we have developed.

Starting with my former PhD student Xiaoli Bai's dissertation, we have very significantly built on classical developments due to Picard by fusing approximation theory and a family of other advances to optimize the resulting algorithms for both serial and parallel computing environments. In contrast to classical step-by-step differential equation solvers in most common usage, the methods we are researching are path iteration methods where paths over time intervals spanning up to several orbits are approximated. In the course of this work, we have developed novel algorithms and compared them with the state of the art algorithms, and also other methods that have recently emerged in the research literature. We have also transitioned the results of this project to the GEO-Odyssey SSA project (a joint effort of AFRL, NRL and NRO, involving 30 investigators); in the course of this project we have confirmed that the accuracy and efficiency results of our research codes are realized when implementing the results in the production SSA codes at AFRL, NRL and NRO (POCs Alok Das, Paul Schumacher, and Shannon Coffey). We have organized this progress report in a "friendly" overview fashion, with discussion of fifteen figures from a recent briefing, and we have attached journal and conference papers as appendices in order to document the details.

Figure 1
Introduction

Accurate numerical orbit propagation is invaluable for protecting space infrastructure, tracking unknown objects, and mission planning

Modified Chebyshev Picard Iteration (MCPI) is a highly accurate ODE solver that can be effectively applied to orbit propagation



9/22/2014

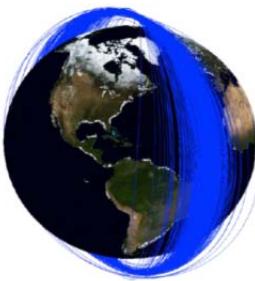


Figure 2
Picard Iteration

French Mathematician (Émile Picard), introduced a classical *successive path approximation method* for solving differential equations initial value problems:

$$\dot{x}(t) = f(t, x(t)), \quad x(t_0) = x_0, \quad t_0 \leq t \leq t_f$$

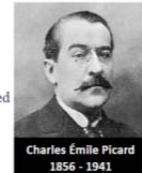
which can be rearranged without approximation to the following integral:

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

Picard, following an idea first introduced by Liouville, hypothesized a sequence of trajectory approximations (Picard Iteration) generated by:

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

Picard established the first convergence proof for this sequence of *path approximations*. While important theoretically, computationally, Picard Iteration remained a "museum piece" until the recent works of Clenshaw, Norton, Feagin, Shaver, Junkins, Bai, Bani-Younes, Macomber, Kim, Woollards, Probe, et al. The key to establishing efficient/accurate algorithms is fusion with approximation theory to obtain spectral approximations of the sequence of integrals. *The method also turns out to be ideally suited for parallel computation...*



Charles Émile Picard
1856 - 1941

The figures below introduce the basic orthogonal approximation approach that is fused with Picard iteration to represent the most fundamental step en route to developing the Modified Chebyshev Picard Iteration (MCPI) algorithms. Notice in Figs. 3, 4 that orthogonal approximation requires consistency among three coupled decisions:

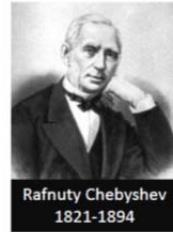
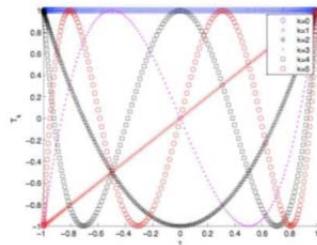
- (1) Choice of basis function (we choose the first N Chebyshev polynomials).
- (2) Choice of the cosine distribution of nodes (which correspond to the interior extrema of the Chebyshev polynomials plus the end points).
- (3) Choice of the a unit weight on each interior node and 0.5 weight on the two end points (this choice, together with the 1st 2 choices, ensure the orthogonality conditions are satisfied, & and

thereby ensure that the normal equations of least square approximation are diagonal; as a consequence, no matrix inversion is needed for arbitrary degree least square approximation).

Figure 3
Chebyshev Polynomial Approximation

- Orthogonal Chebyshev polynomials exist on the range from -1 to 1.
- Defined recursively.
- Integrals and derivatives also defined recursively.

$$\int T_k(x) dx = \frac{1}{2} \left(\frac{T_{k+1}}{k+1} - \frac{T_{k-1}}{k-1} \right)$$



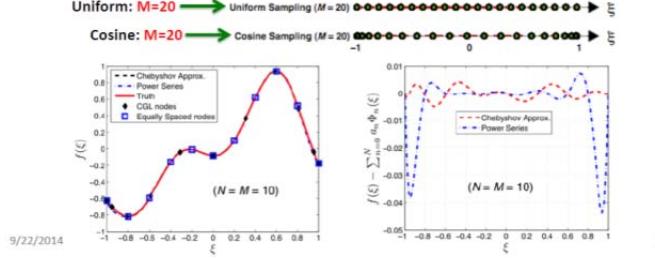
Rafnuty Chebyshev
1821-1894

Figure 4
Chebyshev Polynomial Approximation

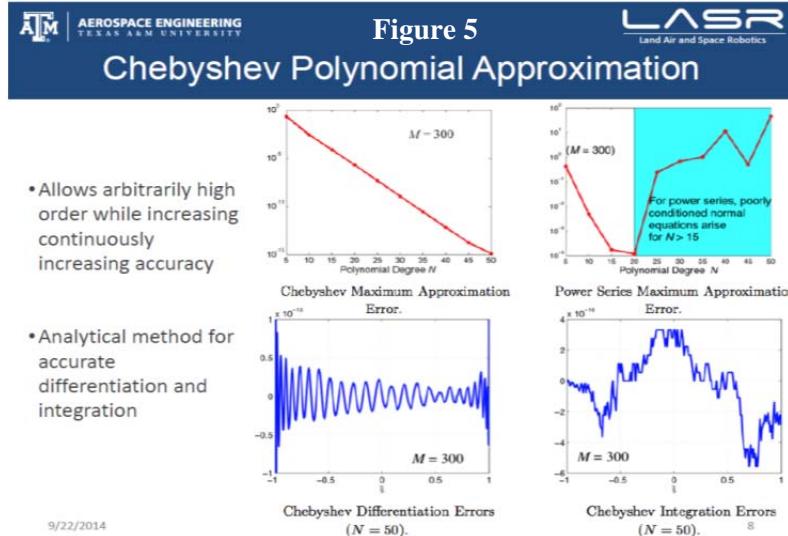
- Sample at Chebyshev-Gauss-Lobatto (CGL) Nodes:

$$\tau_j = -\cos(j\pi/N), \quad j = 0, 1, 2, \dots, N$$

- Higher nodal density near segment boundaries → higher fit accuracy near edges.
- Eliminates Runge phenomena.
- Allows exploitation of discrete orthogonality condition of Chebyshev polynomials.



It is easy to verify the important properties that an arbitrary continuous function can be approximated with spectral precision. Also, in Figure 4, compared to the usual power series approximation, notice the orthogonal Chebyshev approximation avoids the large error oscillations near the end of the interval (*Runge phenomena*), and also permits spectral accuracy (Figure 5) to be approached. The interval length and degree required to achieve spectral accuracy, essentially machine precision, depends on the spectral content of the function. Notice that any method that requires numerical inversion of a matrix can never approach spectral accuracy due to arithmetic errors and ultimately poorly conditioned matrices for large N . Furthermore, for ordinary power series and 64 bit arithmetic, it is not possible to solve the normal equations accurately for $N > 15$. Finally, note in Figure 5 that integration of the approximation leads to slightly smaller approximation errors of the function's integral, whereas differentiation of the approximation results in \sim one order of magnitude loss of accuracy in approximating the derivative of the function.



The conventional approach to collocation-based implicit integrators is to approximate the unknown trajectory as a linear combination of basis functions, then linearize the right hand side of the differential equations about trial values of the basis function coefficients ... then use collocation at $N+1$ nodes to

establish $N+1$ nonlinear algebraic functions to iterate for the unknown coefficients. We do not pursue this path, but rather use the Picard iteration without local linearization. We directly approximate the integrand of the Picard integral, and the path integral requires only term-by-term integration of the Chebyshev polynomials. The result is that linear combination of the coefficients of the integrand approximation, along with imposition of the boundary conditions permits direct update of the path approximation coefficients without a local linearization. The process to establish the integrand coefficient vectors along the $(i-1)^{\text{th}}$ path approximation is indicated in Figure 6.

Figure 6

Modified Chebyshev Picard Iteration

Approximate the integrand using N^{th} order Chebyshev Polynomials as basis functions:

$$\mathbf{g}(\tau, \mathbf{x}^{i-1}(\tau)) = \sum_{k=0}^{N-1} \mathbf{F}_k^{i-1} T_k(\tau) \equiv \mathbf{F}_0^{i-1} T_0(\tau) + \mathbf{F}_1^{i-1} T_1(\tau) + \mathbf{F}_2^{i-1} T_2(\tau) + \dots + \mathbf{F}_{N-1}^{i-1} T_{N-1}(\tau).$$

Transform generic independent variable $t [t_0, t_1]$ to τ , defined on the valid range [-1,1] of Chebyshev Polynomials – scale time and thus system dynamics

$$\text{Picard iteration becomes: } \mathbf{x}^i(\tau) = \mathbf{x}_0 + \int_{-1}^{\tau} \mathbf{g}(s, \mathbf{x}^{i-1}(s)) ds \quad i = 1, 2, \dots$$

$$\text{Calculation of coefficient vectors: } \mathbf{F}_n^{i-1} = \frac{1}{c_n} \sum_{k=0}^N w_k \mathbf{g}(\tau_k, \mathbf{x}^{i-1}(\tau_k)) T_n(\tau_k)$$

^ ^ Inner product ^ ^

$$c_0 = N; \{c_n = N/2; \text{ for } n = 1, 2, \dots, N\}; w_0 = w_N = \frac{1}{2}; \{w_k = 1; \text{ for } k = 1, 2, \dots, N\}$$

Given the integrand approximation coefficients, using the standard formula for integration of Chebyshev polynomials and imposing the intial boundary conditions, the vector coeffients for the i^{th} path approximation can be directly computed as shown in Figure 7.

Figure 7

Modified Chebyshev Picard Iteration

Picard Iteration provides:
$$\mathbf{x}^i(\tau) = \mathbf{x}_0 + \sum_{r=0}^N \mathbf{F}_r^{i-1} \int_{-1}^{\tau} T_r(s) ds \equiv \sum_{k=0}^N \beta_k^i T_k(\tau)$$

The β coefficients are:
$$\beta_r^i = \frac{1}{2r} (\mathbf{F}_{r-1}^{i-1} - \mathbf{F}_{r+1}^{i-1}), \quad r = 1, 2, \dots, N-1$$

Boundary Conditions manifest as constraints on initial/final/both coefficients.

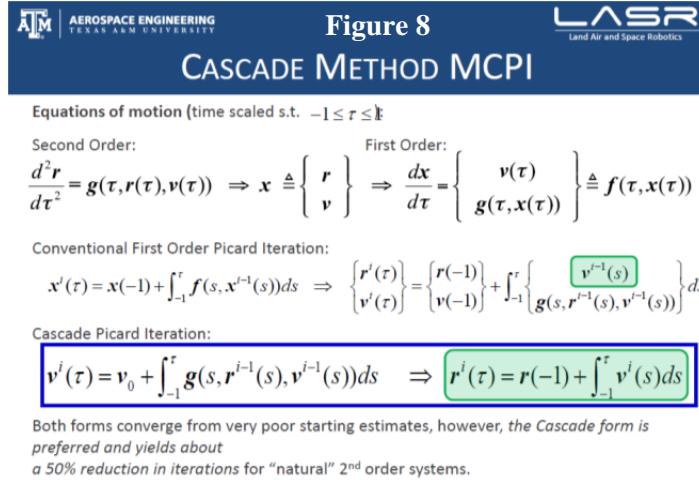
➤ IVPs / FVPs
$$\beta_0^i = \mathbf{x}_0 + \sum_{k=1}^N (-1)^{k+1} \beta_k^i$$

➤ BVPs
$$\beta_{N-1}^i = \frac{\mathbf{F}_{N-2}^{i-1}}{2(N-1)}$$

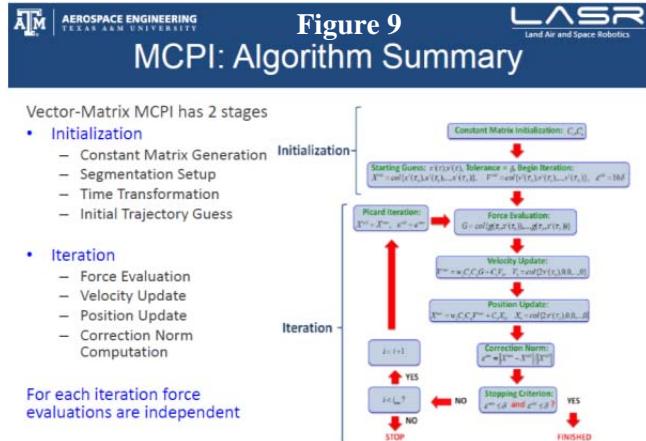
$$\beta_N^i = \frac{1}{2N} \mathbf{F}_{N-1}^{i-1}$$

The above developments represent the basic MCPI algorithm for first order state space dynamical systems' initial value problem. However, for natural systems that are fundamentally represented in a second order state space, we find that MCPI should be written in cascade form in order to realize maximum efficiency. The cascade form of MCPI reduces the dimension of the approximation space by 2 and, remarkably, we have shown that the cascade version of MCPI typically reduces the number of Picard iterations by 50% in comparison to the classical first order version of Picard iteration. The qualitative reason for the 50% reduction in the number of iterations is evident in Figure 8. Notice that when the

second order system of n differential equations is re-arranged as $2n$ first order equations, the first n differential equations are simply an identity (velocity is the derivative of position). When the Picard iteration is applied to this first order system, notice that the i^{th} position vector is the integral of the $(i-1)^{\text{th}}$ velocity approximation. Conversely, when the Picard iteration is applied in cascade fashion, the $(i-1)^{\text{th}}$ acceleration approximation vector is integrated to establish the i^{th} velocity approximation. Notice the result of this step is a Chebyshev series approximation of the i^{th} velocity, which can be integrated term by term to establish the i^{th} position. Thus, the 2nd order cascade approach allows the i^{th} position approximation to be based on the i^{th} velocity approximation instead of the $(i-1)^{\text{th}}$ velocity approximation when using the classical state space version of the Picard iteration. This simple truth has important consequences.

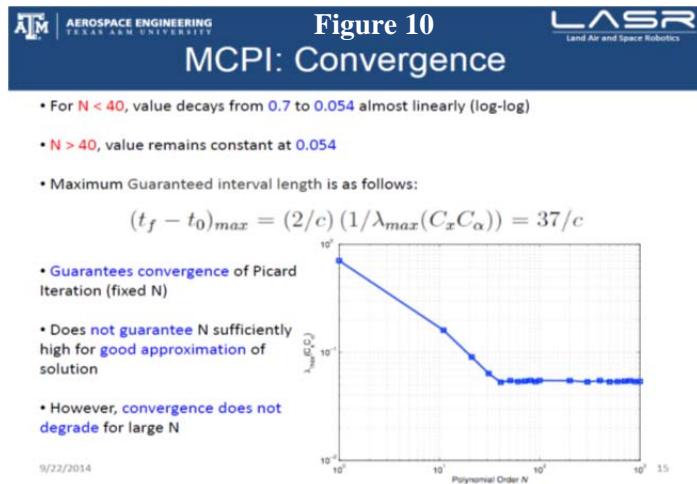


The classical developments above can be re-arranged in vector-matrix form where several matrices can be pre-computed to accelerate subsequent iterations' computations; the vector-matrix form is given in Fig 9.



For the case of a linear system, the relationship between the Chebyshev coefficients and the system states at the nodes are linearly related. This truth allows the Picard iteration to be re-arranged to recursively update nodal states on each iteration with a constant matrix operator $C_x C_\alpha$. The eigenvalues of this linear operator must lie in the unit circle to guarantee convergence. The matrix operator $C_x C_\alpha$ turns out to be scaled by $c = \text{Euclidian norm of the differential equation coefficient matrix}$, and the time interval $(t_f - t_0)$.

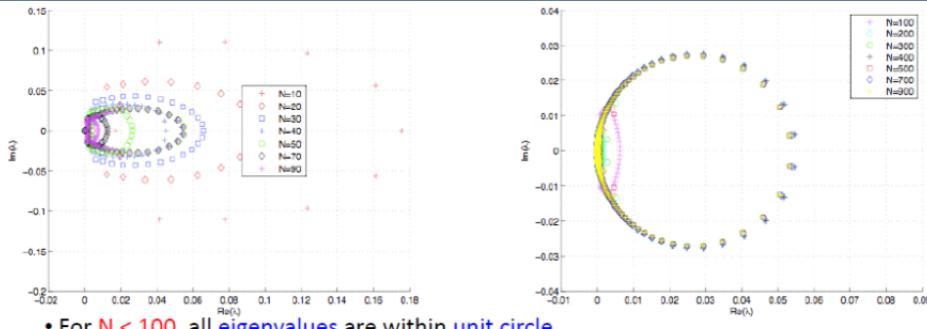
The maximum time interval to guarantee convergence is found to be $(t_f - t_0)_{\max} = (2/c)(1/\lambda_{\max}(C_x C_\alpha))$. As a consequence, we see that the maximum eigenvalue of $C_x C_\alpha$ dictates “everything” about convergence for the case of a linear system. The matrix $C_x C_\alpha$ is completely established by the choice of basis functions, degree N of approximation, and the nodal locations. Thus we can investigate $C_x C_\alpha$ eigenvalues “once and for all” to determine the conditions for convergence. Figure 10 shows the remarkable behavior of the maximum eigenvalue of $C_x C_\alpha$. We see for $N < \sim 45$, $\lambda_{\max}(C_x C_\alpha)$ decreases roughly linearly on a log plot, and thereafter, $\lambda_{\max}(C_x C_\alpha)$ asymptotically approaches a constant for increasing N. Thus the rate of convergence, remarkably, does not increase as N increases (however, increasing N increases accuracy, allowing spectral approximation without encountering a reduction in the rate of convergence).



The remarkable behavior of the maximum eigenvalue of $\lambda(C_x C_\alpha)$ in figure 10 can be further illuminated by looking at the locus of all of the eigenvalues as a function of increasing N. The locus is shown in Figure 11. The vertical axis is the imaginary value of the eigenvalues $\lambda(C_x C_\alpha)$, the horizontal axis is the real value of the eigenvalues of $\lambda(C_x C_\alpha)$. The eigenvalues occur in complex conjugate pairs. The right most pair have the same norm, and this norm is $\lambda(C_x C_\alpha)$. Remarkably, as N increases, the eigenvalues are attracted to fixed positions on a unit circle of diameter 0.054 that “kisses” the origin. The right-most eigenvalues being attracted to fixed positions on the circular locus explains the asymptotic behavior of $\lambda_{\max}(C_x C_\alpha)$ in Figure 10. While this eigenvalue analysis holds only for linear constant coefficient systems, the results provide qualitative insight for nonlinear systems Picard iteration, especially in terminal iterations where Picard iteration provides a near-identity mapping.

Figure 11

MCPI: Eigenanalysis



- For $N < 100$, all eigenvalues are within unit circle
- Distributed along **elliptical shaped loci** – decrease in size and eccentricity as N increases
- For each N , 2 sets \rightarrow 1 near origin and other loops to the right
- Eigenvalues are attracted to **distinct locations** along unit circle (radius **0.027**)
- For $N > 100$, eigenvalues cluster on **circular locus** near origin

While the Picard iteration can be guaranteed to be a contraction mapping for a large class of nonlinear systems, and the contraction mapping can be completely analyzed for the case of linear systems, guaranteeing efficiency requires further insights and care in implementation. Specifically, Picard iteration is known to converge geometrically, as compared to terminal quadratic convergence for most Newton-like algorithms based on local linearizations. Thus, for very good starting approximations, we can expect the number of Picard iterations to be greater than, for example a co-location algorithm based on local linearizations. However, the fixed point nature of Picard iterations allows several additional methods to be introduced that dramatically accelerate convergence. It should be noted that Picard path iterations are inherently parallelizable, so if the Picard iteration can be accelerated so that it “wins” with regard to computational time for a given accuracy in a serial implementation, then it is virtually certain to be superior in a parallel implementation.

The enhancements of the MCPI algorithm can be described with the following captions:

- Segmentation and order selection (take advantage of physics knowledge, if possible)
- Warm and hot starting approximate solutions
- Radially adaptive gravity (for the case of orbit solutions)
- Terminal Convergence Force Approximations for MCPI

The first issue is discussed briefly with reference to Figure 12. For the case of orbit problems, it is well known (Kepler’s Laws) that the nonlinearities are strongest at perigee where the gravitational perturbations and drag perturbations are largest and change more nonlinearly with position and velocity, whereas the slower motion and weaker nonlinearities are near apogee. Thus, we expect optimal accuracy (for a given number of nodes) will result with high nodal density near perigee and sparse nodes near apogee. These qualitative expectations, supported by numerical studies, result in an optimal pattern that has an odd number of segments {1,3,5,...} with the optimum starting point being at perigee, and the center of a segment being at Apogee. The optimal segment break point on the orbit has been found to be a function of three variables (perigee radius, eccentricity, and desired accuracy), for a given gravity field

model and atmospheric density model. Obviously perigee radius dictates how deeply the orbit dips into the more nonlinear part of the gravity field and the extent to which atmospheric drag affects the orbit. Eccentricity dictates the variability of the angular displacement with time, and the increasing apogee radius that decreases the strength and frequency content of acceleration perturbations. Figure 12 shows one such optimal pattern for the case of three segments. It is important to note that a particular pattern that characterizes the optimal solution must also show that the estimated accuracy of the converged solution be approximately uniform over the orbit. We also highlight useful methods that we have developed for estimating the solution accuracy.

There are essentially two sets of issues: (1) How to establish an initial conservative choice for the segmentation pattern and number N of nodes. (2) How to adapt the segmentation pattern and number of nodes to ensure both accuracy and efficiency. The first issue can be addressed fully by an a priori parametric study (Figure 13). If this process results in the ability to quickly establish conservative segment break times, along with a choice of choose N that ensure accuracy with reasonable sub-optimal efficiency, the second issue (adaptation) is a desirable luxury, but may not be required for many applications.



Figure 12
Segmentation Setup



- Traditional time-step determination does not work for MCPI
- MCPI setup requires nodal density selection
- A Study was performed to generate a segmentation scheme
 - Based on Physical Insight and Heuristics
 - Symmetric periodically repeating based on perigee
 - 3 segments with higher nodal density and smaller time spans near perigee

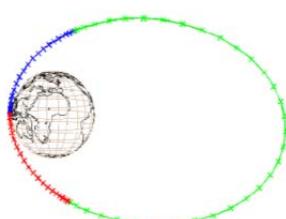


Figure 13
MCPI Parameter Study



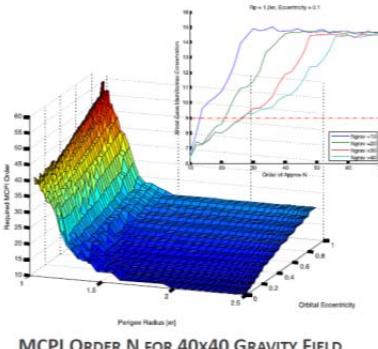
- For a given:
 - Perigee radius
 - Eccentricity
 - Required physical accuracy
- Need a scheme to choose:
 - Number of MCPI segments around the orbit
 - Order of approximation of each segment
 - Relative lengths of segments in True Anomaly
- Such that:
 - Minimize the number of function evaluations around the orbit
 - Place MCPI nodes optimally to get the most "bang for the computational buck"
 - Take advantage of radial adaptation gravity
 - Meet a given accuracy requirement – conserve energy evenly around the orbit

Shown below in Figure 14 is a response surface revealing the required number of approximation nodes, for a 3 segments, as a function of perigee radius and eccentricity. For near earth orbits in a gravity field with degree and order set to (40,40), this surface is optimized for nine digits accuracy (corresponding to centimeter level of position accuracy, which is more accurate than the physical accuracy of the gravity model if perigee is smaller than 1.5 earth radii). The tuning is designed to be conservative, and we have found that even without adaption, this tuning guarantees accuracy with good efficiency. The tuning of the algorithm (number of segments and number of nodes) as a function of cost (measured by the number of local force (gravity) evaluations) and the resulting accuracy achieved is shown in Figure 15. This optimization could be done a number of ways, for convenience, we used a generic algorithm and achieved good results. This is repeated over a family of neighboring orbits to establish the parametric relationship between {accuracy, efficiency, number of segments, number of nodes} for optimal approximation. This data base can be accessed to launch the solution process with conservative sub-optimal tuning.

Figure 14

MCPI Parameter Study: BRUTE FORCE METHOD

- For a given:
 - Perigee radius
 - Eccentricity
 - Required physical accuracy
 - Fixed number of segments around the orbit
- Look at the first segment of the orbit (starting from perigee). Vary the MCPI order of approximation and determine the necessary order to preserve the Hamiltonian to the given accuracy. Do this for various order gravity fields.
- Loop over feasible parameter space and calculate required order of approximation as a function of parameters.
- Gives a conservative estimate of the required MCPI order of approximation

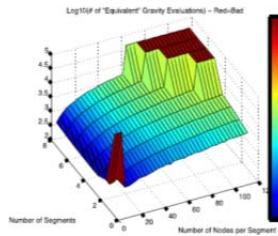


MCPI ORDER N FOR 40X40 GRAVITY FIELD

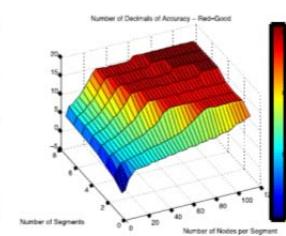
Figure 15

GENETIC ALGORITHM

GRAVITY EVALUATIONS



ACHIEVABLE ACCURACY



LEO ORBIT

Given a good choice on the tuning parameters, a very important issue that governs efficient convergence is the closeness of the starting approximation. This issue is relatively easy to address for problems in celestial mechanics because the two-body problem is often accurate to a few significant figures over one orbit. However, over longer time intervals, this solution breaks down. In addition, for the computation of multiple orbits, successive orbits experience highly correlated non-two-body perturbations, if reference to the previously completed orbit's osculating perigee state. Figure 16 illustrates qualitatively the *warm start* (two body analytical approximation), typically, although we can also include the first few zonal harmonic perturbations, and a hot start correction, based upon the displacement of the final converged orbit from the warm start orbit at the nodes on the previous nodes. We have found that the warm start typically gives 3 correct digits, and the hot start correction typically has errors in the 5th significant figure, therefore very significantly accelerating the process by eliminating the early iterations to get into the terminal convergence regime. The consequences are dramatic. Figure 17 shows the convergence that ensues with a "cold start" (linear extrapolation of initial state), a "warm start" (two body approximation based on initial conditions), and a "hot start", an Encke-like correction where the displacement from the previous warm start to the final converged solution is added to the current warm start to initiate the next orbit's approximation. As is evident, better than one order of magnitude improvement at each iteration is made by using a warm start, and another order of magnitude by using a hot start.

Figure 16

MULTI-ORBIT "HOT START"

- MCPI node times are known
- Initialize node positions & velocities with analytical 2-body solution \Rightarrow *warm start*
- Iterate to MCPI convergence on ($n-1$)th orbit segment
- Form difference between *warm start* and final converged value at each node \Rightarrow the *nodal displacement* from *warm start* after each converged orbit... Essentially, this is the "Encke Displacement" and is the net departure motion due to all non-two-body perturbations.
- \Rightarrow Apply the $n-1$ th nodal displacements to the n^{th} orbit's osculating *warm start* to make it a *hot start*

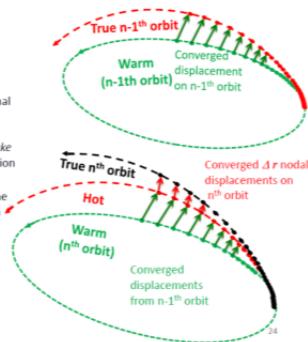
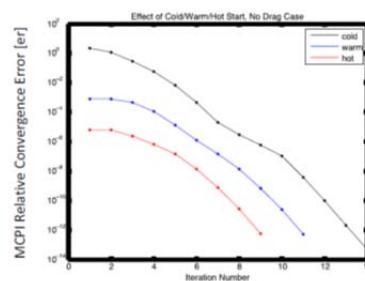


Figure 17

MULTI-ORBIT "HOT START"



In this case (GEO/HEO orbit with no drag), hot start provides \sim 6 orders of magnitude better initial estimate than cold start, and reduces the required number of iterations from 14 to 9.

Certain forces exhibit a particular behavior that allows an additional state-dependent acceleration. An excellent and frequently occurring example is the earth's gravity field which decays in strength and spatial granularity as the distance from the center of the earth increases. There are several ways to think

about this truth. Among the simplest is to ask the question: For a prescribed gravity model, how many terms in that model {specified by choosing the degree and order} are required as a function of the earth's radius and the desired number of significant figures in the gravity model. The right graph of Figure 18 shows the surface giving N as a function of radius and accuracy. The radius is in units of the Earth's radius. As is evident, except inside 2.5 earth radii, it is virtually never required to have a degree and order greater than (20,20) and at the GEO orbit, (6.623 Earth radii), we never require a degree and order above (6,6). Only for lower altitudes inside about 1.75 Earth radii do we need to consider models with degree and order above (100,100). The nature of this surface is known a priori, so any call to compute gravitational acceleration should be accompanied by a prescribed accuracy, in addition to the maximum degree, order. As is evident in the left curve of Figure 18, the computational cost to compute the higher order gravity is significant in a serial processor, although this cost can be negated to a significant degree in a parallel computing implementation.

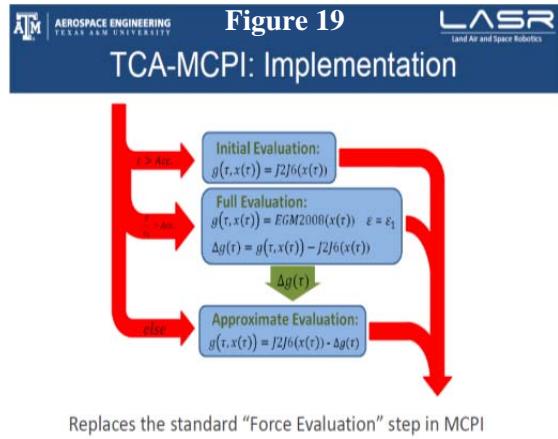
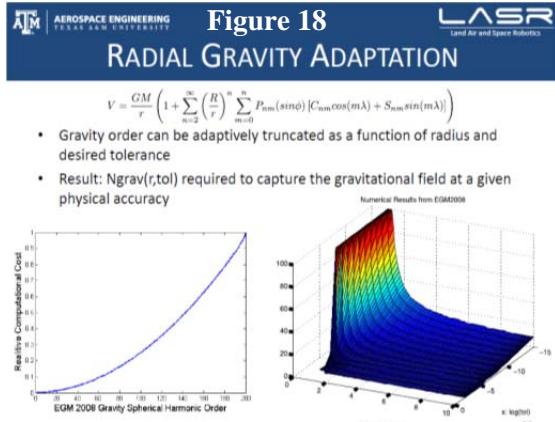
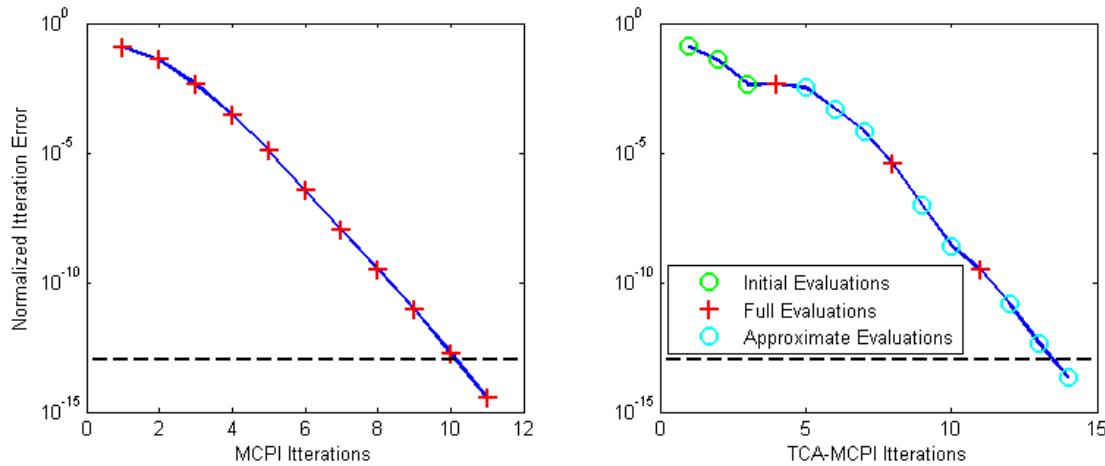


Figure 19 briefly addresses another issue. The fact that convergent MCPI process is a contraction algorithm means that the nodes are attracted to fixed points in the gravity field. We have developed metrics that enable the algorithm to approximately determine the number of significant digits in the solution associated with the current Picard iterations, as well as the final desired accuracy. This means that we can utilize a variable fidelity gravity model, whereby an occasional accurate force computation is followed by local approximations consistent with the progress toward final convergence. We have found one way to do this is to utilize a zonal gravity approximation to capture the generic shape of the gravity field in the vicinity of a node. The departure (offset) from this local inexpensive gravity model can be added based on an occasional update from the full force model. Remarkably, as is evident in Figure 20, we can use this approach, following warm/hot starts and most often only two full force models evaluations in order to achieve final precision tolerances of 9 digit accuracy, and 14 digit accuracy with only two full fidelity force model evaluations. The initial evaluations (green symbol) of this figure began with a warm start and the first six zonal harmonics.

The consequence of these several insights (warm/hot starts, radially adaptive gravity, and local force approximation) is a substantial speedup of MCPI, and in fact, based on the extensive studies to date, we can claim it represents the present state of the art for efficient/accurate orbit propagation (as evidenced below) for serial implementations. Moreover, since almost all of the competing algorithms, in particular the Gauss-Jackson industry standard, are known to be very difficult to parallelize and the MCPI is inherently parallelizable, we conclude that MCPI provides dramatic advantages (especially as we move to exploit the emerging massively parallel architectures)

Figure 20



In order to illustrate the implications of the above developments, we present a study involving six orbits over a range of orbit elements (see the table in Figure 21). These orbits include low eccentricity low earth orbits, highly eccentric orbits, near circular orbits near GEO, and intermediate orbits. This variety of orbits is sufficient to exercise the current implementation of MCPI and compare to five competing algorithms that represent both the state of the art and the state of the practice. All of the solution algorithms make use of the same intermediate degree and order gravity model (40, 40) and included no other perturbation effects. This force model has an exact integral (the Hamiltonian) that is theoretically constant, and this allows one immediate metric to be computed as a necessary condition measure of the accuracy of the solution obtained for any of the orbits by any of the competing methods. Two other methods provide stronger validation of closure between the approximate solutions and the true solutions, these are detailed in the appendices (the method of manufactured solutions (MMS) and the round trip closure (RTC) method). These two methods are only employed selectively due to the higher cost, the preservation of the Hamiltonian is demonstrated to behave consistently with MMS and RTC for conservative systems, however, both MMS and RTC are applicable to general non-conservative systems.

Figure 21
Results: Integrator Performance Study

| | 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 |

For each of these orbits, we summarize the five competing algorithms accuracy metrics and relative efficiency in the figures below; we used the unperturbed orbit period to establish the final time. It is clearly evident that MCPI produces the most efficient solution for the prescribed accuracy for all six orbits. These are all serial results on a conventional personal computer (specifications in the appendix), and we are presently implementing the same algorithms on a representative parallel computing environment. For the present discussion, Figs 22-25 show results from each of the six integrators. The lower right sub-figure in each of Figs 22-25 shows the relative error in preserving the Hamiltonian. In each case the MCPI algorithm was tuned to give a slightly better solution (smaller error) than the 5 competing algorithms, and for each of the competing algorithms, the tuning was optimized for a common

accuracy tolerance. For each case, as is evident, the speedup relative to the best competing algorithms varies from $\sim 2x$ to $\sim 10x$. The use of parallel computation will result in an additional two orders of magnitude speedup.

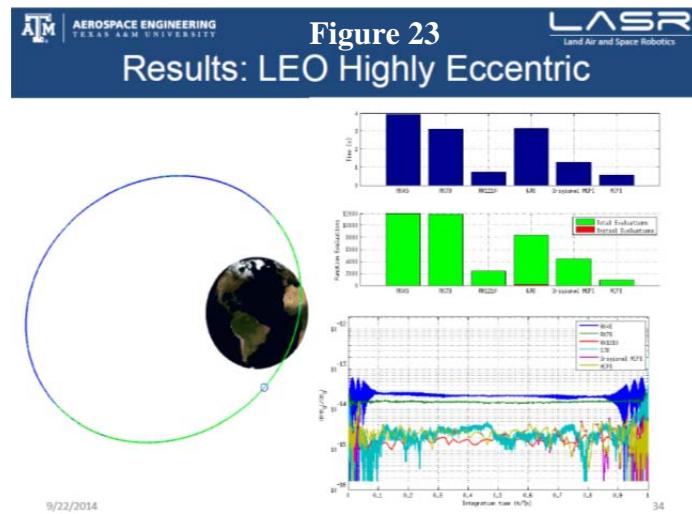
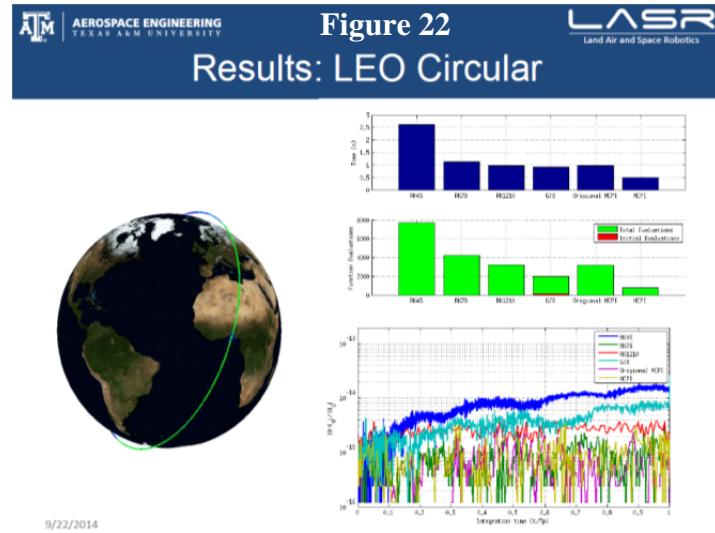
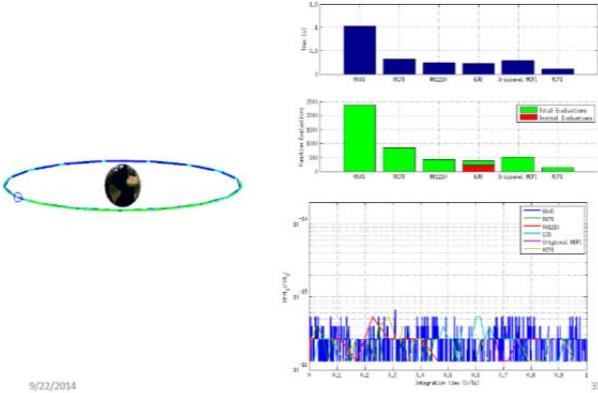
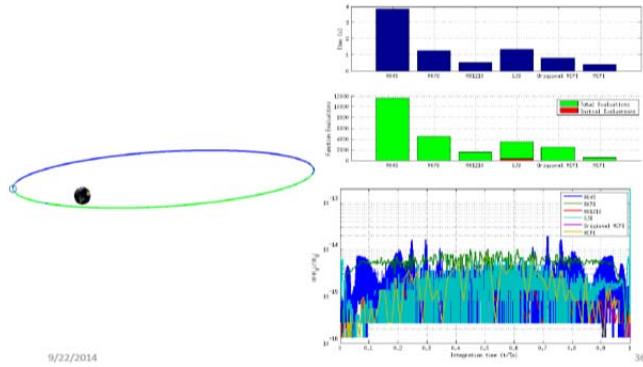


Figure 24
Results: GEO Circular



9/22/2014

Figure 25
Results: GEO Highly Eccentric



9/22/2014

Figure 26
Future Work

1. Propagation Applications: SSA and Mission Planning
2. Boundary Value Problem work (Lambert's Problem)
3. Develop methods that take advantage of parallel structure
4. Look at optimization for other problems; e.g. rotating body
5. Generate general integration methods such as adaptive time segmentation and node density determination

Appendices

The following attached publications containing details from this research project are appended to this report:

1. Macomber, B., Woollards, R., Probe, A., Bani Younes, A., Junkins, J., “Modified Chebyshev Picard Iteration for Efficient Numerical Integration of Ordinary Differential Equations”, Advanced Maui Optical and Space Surveillance Technologies Conference, Maui, Sep 2013.
2. Kim, D., Junkins, J., Turner, J., Bani Younes, A., “Multi-Segment Adaptive Modified Chebyshev Picard Iteration Method”, 24th AAS/AIAA Space Flight Mechanics Meeting, Santa Fe, MN, Jan 2014.
3. Kim, D., Junkins, J., Turner, J., “Multisegment Scheme Application to Modified Chebyshev Picard Iteration Method for Highly Elliptical Orbits”, Mathematical Problems in Engineering, 2014.
4. Woollards, R., Junkins, J., Bani Younes, A., “A New Solution for the Generalized Lambert’s Problem”, 37th Annual AAS Guidance & Control Conference, Breckenridge, Feb 2014.
5. Woollards, R., Bani Younes, A., Junkins, J., “New Solutions for Lambert’s Problem Utilizing Regularization and Picard Iteration”, Journal of Guidance Dynamics and Controls, Richard H. Battin Special Edition, submitted Sep, 2014.
6. Woollards, R., Bani Younes, A., Macomber B., Probe, A., Kim, D., Junkins, J., “Validation of Accuracy and Efficiency of Long-Arc Orbit Propagation using the Method of Manufactured Solutions and the Round-Trip-Closure Method”, Advanced Maui Optical and Space Surveillance Technologies Conference, Maui, Sep, 2014.
7. Probe, A., Macomber, B., Kim, D., Woollards, R., Junkins, J., “Terminal Convergence Approximation Modified Chebyshev Picard Iteration for Efficient Numerical Integration of Orbital Trajectories”, Advanced Maui Optical and Space Surveillance Technologies Conference, Maui, Sep, 2014.

Modified Chebyshev Picard Iteration for Efficient Numerical Integration of Ordinary Differential Equations

Brent Macomber, Robyn M. Woollards, Austin Probe, Ahmad Bani Younes, John L. Junkins

Texas A & M University, Aerospace Engineering Dept, H.R. Bright, 3141 TAMU, College Station, TX, 77843-3141

Xiaoli Bai

Optimal Synthesis, Inc, 95 1st St. Suite 240, Los Altos, CA 94022

ABSTRACT

Modified Chebyshev Picard Iteration (MCPI) is an iterative numerical method for approximating solutions of linear or non-linear Ordinary Differential Equations (ODEs) to obtain time histories of system state trajectories. Unlike other step-by-step differential equation solvers, like the Runge-Kutta family of numerical integrators, MCPI approximates long arcs of the state trajectory with an iterative path approximation approach, and is ideally suited to parallel computation. Orthogonal Chebyshev Polynomials are used as basis functions during each path iteration, and the integrations of the Picard iteration are then done analytically. The orthogonality of the Chebyshev basis functions mean that the least square approximations can be computed without a matrix inversion; the coefficients are conveniently computed robustly from discrete inner products. As a consequence of discrete sampling and weighting adopted for the inner product definition, the Runge phenomena errors that usually occur near the ends of the approximation intervals are significantly minimized. The MCPI algorithm utilizes a vector-matrix framework for computational efficiency. Additionally, all Chebyshev coefficients and integrand function evaluations are independent, meaning they can be simultaneously computed in parallel for further decreased computational cost. Over an order of magnitude speedup from traditional methods is achieved in serial processing, and an additional order of magnitude is achievable in parallel architectures.

This paper presents a new MCPI library, a modular toolset designed to allow MCPI to be easily applied to a wide variety of ODE systems. Library users will not have to concern themselves with the underlying mathematics behind the MCPI method. Inputs are the boundary conditions of the dynamical system, the integrand function governing system behavior, and the desired time interval of integration, and the output is a time history of the system states over the interval of interest.

Examples from the field of astrodynamics are presented to compare the output from the MCPI library to current state-of-practice numerical integration methods. It is shown that MCPI is capable of out-performing the state-of-practice in terms of computational cost and accuracy.

1. INTRODUCTION

Modified Chebyshev Picard Iteration (MCPI) is an iterative numerical method for solving linear or non-linear ordinary differential equations. It combines the discoveries of two great mathematicians: Émile Picard (Picard Iteration) and Rafnuty Chebyshev (Chebyshev Polynomials). The decision to make use of these techniques in a simultaneous manner was first proposed by Clenshaw and Norton in 1963 [1].

Picard stated that any first order differential equation

$$\dot{x}(t) = f(t, x(t)), \quad x(t_0), \quad (1.1)$$

with an initial condition $x(t_0) = x_0$, may be rearranged without approximation to obtain the integral equation shown in Eq. (1.2).

$$x(t) = x(t_0) + \int_{t_0}^t f(\tau, x(\tau)) d\tau. \quad (1.2)$$

A sequence of approximate solutions, $x^i(t)$, ($i = 1, 2, 3, \dots, \infty$), to this differential equation may be obtained through Picard iteration using the following formula:

$$x^i(t) = x(t_0) + \int_{t_0}^t f(\tau, x^{i-1}(\tau)) d\tau, \quad i = 1, 2, \dots \quad (1.3)$$

In the MCPI method, orthogonal Chebyshev polynomials are used as basis functions to approximate the integrand in the Picard integral. Chebyshev polynomials reside in the domain $\tau = [-1, 1]$, and can be defined recursively as:

$$T_0(\tau) = 1, \quad (1.4)$$

$$T_1(\tau) = \tau, \quad (1.5)$$

$$T_{k+1}(\tau) = 2\tau T_k(\tau) - T_{k-1}(\tau). \quad (1.6)$$

Unlike traditional step-by-step integrators, for example the Runge-Kutta methods, MCPI is unique in that long state trajectory arcs are approximated during the Picard iteration. The system dynamics are normalized such that the timespan of integration is projected onto the domain of the Chebyshev polynomials, thus the system states can be approximated using the Chebyshev polynomial basis functions. The orthogonal nature of the basis functions means that the coefficients that linearly scale the basis functions can be computed independently as simple ratios of inner products with no matrix inversion.

As a consequence of the independence of the basis functions, the coefficients multiplying the Chebyshev basis functions may be computed in parallel by separate processor threads. This is the first of two available layers of parallelization in the MCPI method. The second layer of parallelization is enabled by the fact that the entire state trajectory over the time interval of interest is estimated at once. Thus the calculation of the integrand function (which is a function of the system states) can be performed all at once on parallel processor threads. Using MCPI, over an order of magnitude speedup from traditional methods is achieved in serial processing, and an additional order of magnitude is achieved in parallel architectures.

A key feature of MCPI is a non-uniform cosine density sampling of the domain of the Chebyshev basis functions called Chebyshev-Gauss-Lobatto (CGL) nodes, defined in Eq. (1.7).

$$\tau_j = \cos(j\pi / N), j = 0, 1, 2, \dots, N \quad (1.7)$$

This sampling scheme has much higher density towards the edges, which enables a higher accuracy solution near the boundaries of the state trajectory. This scheme eliminates the Runge phenomena, a common issue in function approximation whereby noisy estimates are returned near the edges due to lack of knowledge of the states on the other sides of the boundaries. The coefficients multiplying the Chebyshev basis functions are approximated by the method of least squares, which generally requires a matrix inversion. A wonderful side effect of the cosine sampling scheme is that the matrix required to be inverted in the Normal Equations of least squares is diagonal, thus the inverse is trivial.

In 2010, Bai's dissertation [2] laid the groundwork of MCPI and proved the capability of the method to outperform the state of the practice for numerical integration of ODEs. Bai and Junkins applied MCPI to non-linear IVPs and orbit propagation in [3], and showed that MCPI can outperform other higher order integrators such as Runge-Kutta-

Nystrom 12(10). In [4] Bai and Junkins applied MCPI to efficiently solving Lambert's transfer problem, and to solving an optimal control trajectory design problem more accurately and efficiently than the Chebyshev pseudospectral method. In [5] Bai and Junkins use MCPI in a complex three-body station-keeping control problem formulated as a BVP. Subsequent publications by Junkins et al. [6], [7], and [8] further clarify the concept and derivation of MPCI and orthogonal approximation in general, and apply the method to problems in the field of astrodynamics.

A full derivation of MCPI is beyond the scope of this short paper. Instead we present a flow chart in Fig. 1 briefly summarizing the mathematics underlying the MCPI method for solution of an Initial Value Problem (IVP). Fig. 2 is the same mathematics represented in the more elegant vector/matrix formulation, which is computationally the most efficient way to implement the method. Any of the above references provide more detailed derivations, as well as examples and results that demonstrate the power of the MCPI algorithm with regard to speed and accuracy. Additionally, those references contain comparisons to other well-known integrators including high-order Runge-Kutta methods and the Gauss-Jackson method.

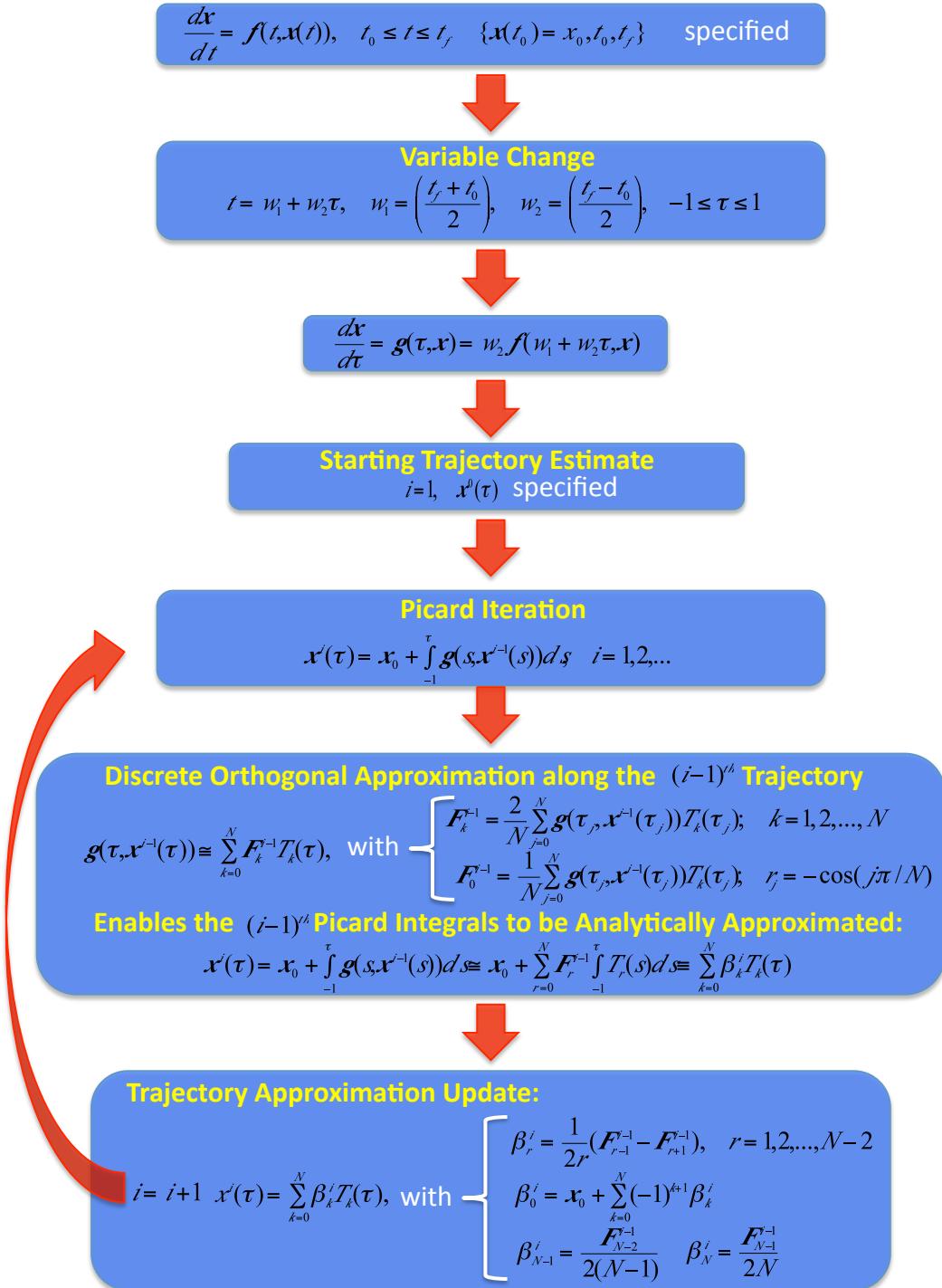


Fig. 1. Flow diagram of MCPI Initial Value Problem implementation.

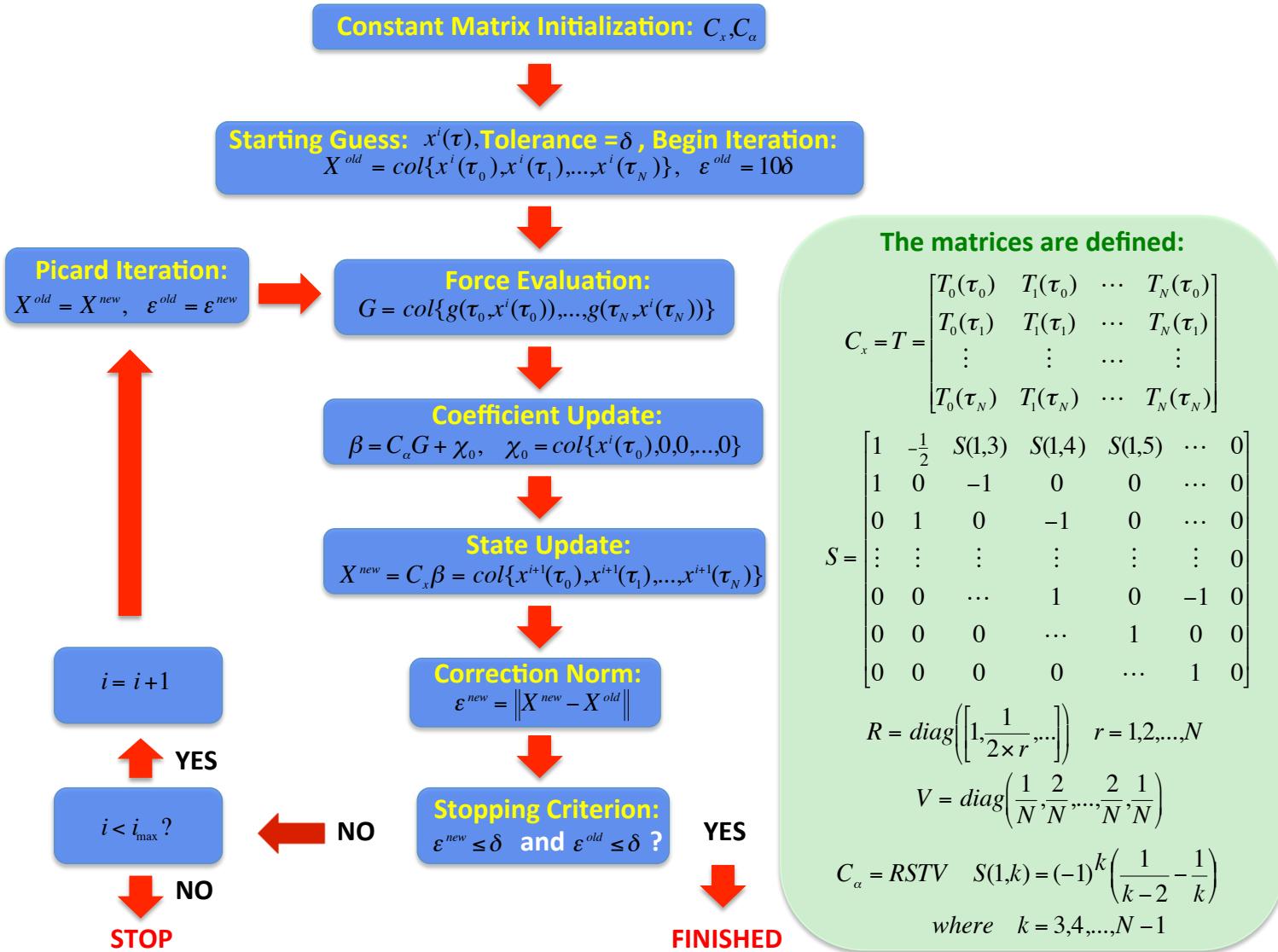


Fig. 2. Flow diagram of MCPI algorithm in vector-matrix form.

2. TAMU MCPI LIBRARY

This paper introduces the Texas A&M University MCPI libraries (TAMU MCPI), which have been created to encourage widespread use of the MCPI method for solution of Ordinary Differential Equations. The goal of the project is to create an easy to use toolset that effectively eliminates the learning curve of using MCPI methods, but at the same time is versatile and powerful enough for application to a variety of projects. The user is not required to have a thorough understanding of the inner-workings of MCPI in order to implement it in their own projects. TAMU MCPI is a set of efficient and lightweight classes for solution of Initial Value Problems (IVPs) and Boundary Value Problems (BVPs). Solvable ODEs can be linear or non-linear, autonomous or non-autonomous, and first-order or second-order. Higher order systems are solvable by decomposition to a first-order or second-order system by the inclusion of additional states that are the time derivatives of lower order states.

Fig. 3 shows a high-level overview of the TAMU MCPI structure from an implementation point of view. The user provides a handle to an integrand function for the problem at hand, that is, the update function that describes how the time derivatives of the system states behave. Additionally, the user provides the relevant boundary conditions for the system states, defined at the initial time, the final time, or both, depending upon the problem to be solved. If the system has time-varying parameters, or other numerical data is required in the integrand function, these may be inputted as well. Given these inputs, TAMU MCPI will iteratively attempt to numerically solve the state-space trajectories of the system over the desired time interval. If a solution is found, the time history of the system states over the interval of interest is returned.

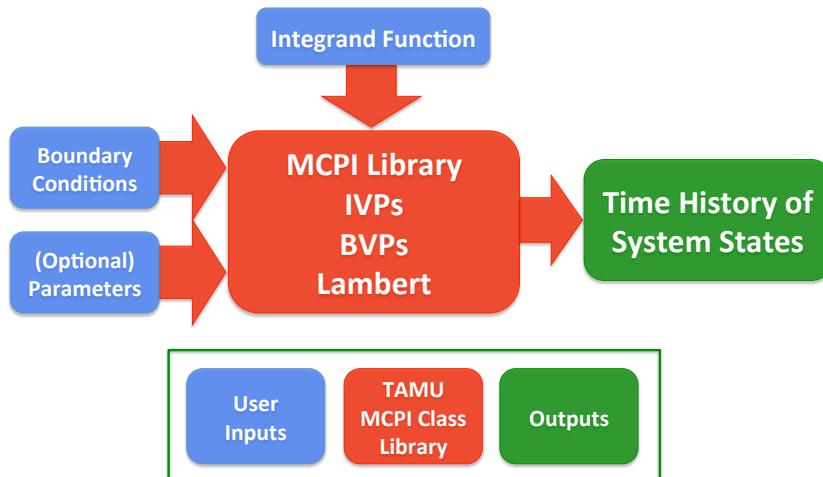


Fig. 3. High-level overview of TAMU MCPI library.

The TAMU MCPI library is available in Matlab, C++, and as Matlab wrapper functions to the CUDA parallel computation environment. CUDA stands for Compute Unified Device Architecture, and is a parallel computing language developed by NVIDIA for use upon their Graphics Processing Units (GPUs); effectively it allows lightweight parallel computation at a desktop workstation. TAMU MCPI is fully cross-platform, and has been tested on Windows, Linux, and Apple computers. The structure of the libraries is hierarchical, with an abstract parent class and derived child classes tailored to the solution of various problem types. This modular approach is to allow for future expansion, or application-specific customization and optimization. Control parameters can be set from a configuration file or interactively by the user.

The C++ libraries can be distributed as source code with minimal external dependencies (the only dependencies are headers from the Boost cross-platform library¹), or as pre-compiled binaries and header files for many widely used operating systems. Compiling the libraries from source is possible with any reasonable C++ compiler, and include

¹ Boost is a set of cross-platform C++ tools to accomplish common tasks. TAMU MCPI uses header-only Boost libraries to avoid inclusion of large binary files. See <http://www.boost.org/> for more information.

files and linking are managed with CMake². The CUDA libraries utilize the Matlab Parallel Computation Toolbox, and require Matlab 2010 or newer (2011 or newer recommended), and an NVIDIA GPU with compute capability of 1.3 or greater.

3. EXAMPLE: ORBITAL PROPAGATION OF DEBRIS CLOUD

In this example, we forward propagate the orbital motion of a cloud of 1000 simulated debris objects in Low Earth Orbit. Initially the cloud is a three-dimensional Gaussian distribution with mean initial position and velocity and distribution parameters as shown in Table 1. The mean particle orbital eccentricity is $e = 0.0099$, and the mean orbital period is $P = 5.3905 \times 10^3$ seconds. The motion of each object is propagated forward by one (mean) orbital period using a simple inverse square gravity model. The initial and final distributions are shown in Fig. 4 (note that the Earth is shown solely to provide scale, the coordinate system is arbitrary).

This numerical integration is performed using the TAMU MCPI Initial Value Problem library running in Matlab 2013, and benchmarked against the native Matlab Runge-Kutta 4(5) variable step size numerical solver ODE45. The comparison is carried out on a laptop computer with an Intel Core i7 2.3GHz processor, and 16GB of RAM. The accuracy of the numerical solution is verified against the analytic F and G solution, and both algorithms are tuned to have similar accuracy as shown in Fig. 5, in which the motion of a single particle is propagated forward by several orbits. In this arrangement, the Matlab implementation of TAMU MCPI forward propagates the particle cloud motion five times faster than ODE45, and with comparable accuracy.

Table 1: Parameters required for the IVP solution.

| Orbit Parameters | |
|--|---------------------------------|
| Propagation Time (s) | 5.3905×10^3 |
| Mean Particle Initial Position Vector (km) | [-464.856, 6667.880, 574.231] |
| Mean Particle Initial Velocity Vector (km/s) | [-2.8381, -0.7872, 7.0830] |
| Standard Deviation Particle Position (km) | 0.1 |
| Standard Deviation Particle Velocity (km/s) | 0.1 |

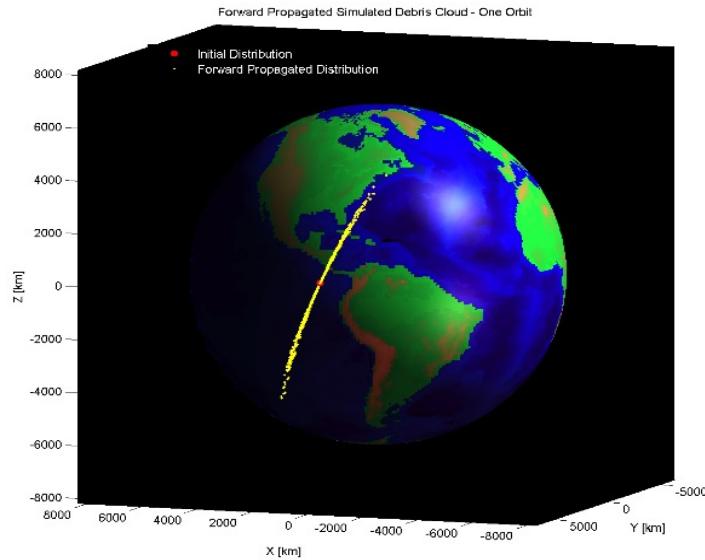


Fig. 4: Simulated debris cloud, initial Gaussian distribution and final distribution after forward propagation by one mean orbital period.

² Cmake is a cross-platform build tool that creates projects such that the native compiler can build applications from source code. See <http://www.cmake.org/> for details.

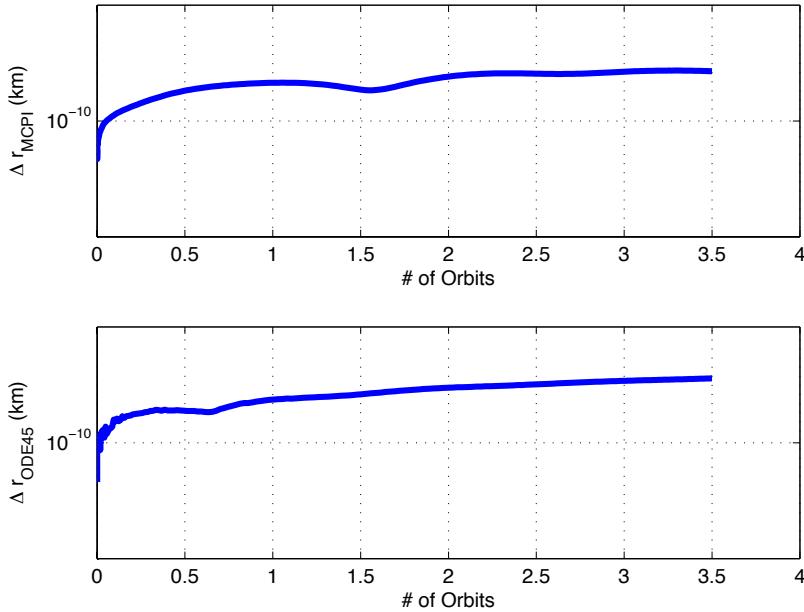


Fig. 5. Position errors of the MCPI algorithm (top panel) and the ODE45 (bottom panel) compared with the analytic F and G solution.

4. EXAMPLE: LAMBERT'S TRANSFER PROBLEM

We solve the orbital motion for a section of a Low Earth Orbit given boundary conditions on the initial and terminal position as well as the time taken for the motion, a formulation called Lambert's Problem. These input parameters are shown in Table 2. The period of the chosen orbit is $P = 5.3905 \times 10^3$ seconds, and the eccentricity is $e = 0.0099$.

This problem is solved with the TAMU MCPI Second Order Boundary Value (Lambert-Style) library running in Matlab 2013, and benchmarked against the Shooting Method using fsolve and ODE45. The comparison is carried out on a laptop computer with an Intel Core i7 2.3GHz processor, and 16GB of RAM. The output from the two solvers are verified against the analytic F and G solution, and the parameters of both algorithms are tuned until the accuracy is comparable, as shown in Fig. 7. Depending upon the desired arc-length of solution, the Matlab implementation of TAMU MCPI is able to solve the Lambert Problem 20-60 times faster than the shooting method with fsolve and ODE45, and with comparable accuracy.

For this given orbit, the MCPI BVP algorithm maximum arc length over which convergences occurs is 38% of an orbital period. We are currently investigated promising new methods to increase this arc length, and these will appear in subsequent publications.

Table 2: Parameters required for the BVP solution.

| Orbit Parameters | |
|------------------------------|------------------------------------|
| Propagation Time (s) | $0.38 * 5.3905 \times 10^3$ |
| Initial Position Vector (km) | [-464.856, 6667.880, 574.231] |
| Final Position Vector (km) | [-1386.506, -5174.986, 3873.216] |

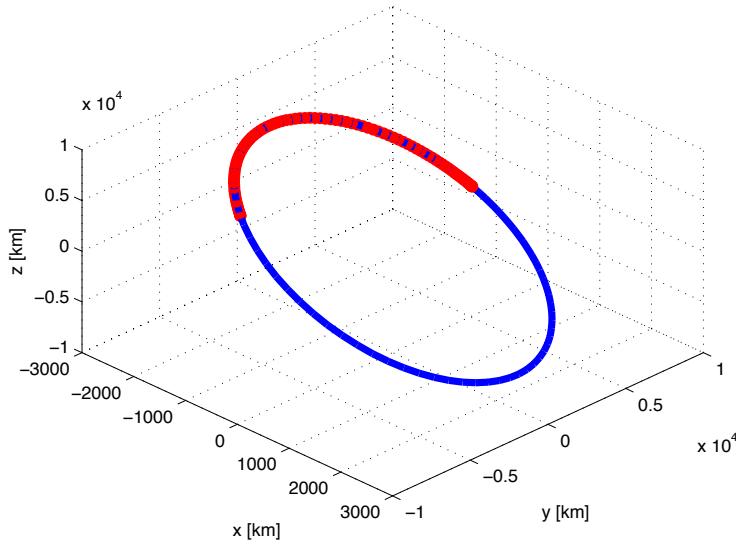


Fig. 6: The reference orbit generated from an F and G solution (blue), and the 38% time period arc (red) propagated with the BVP algorithm.

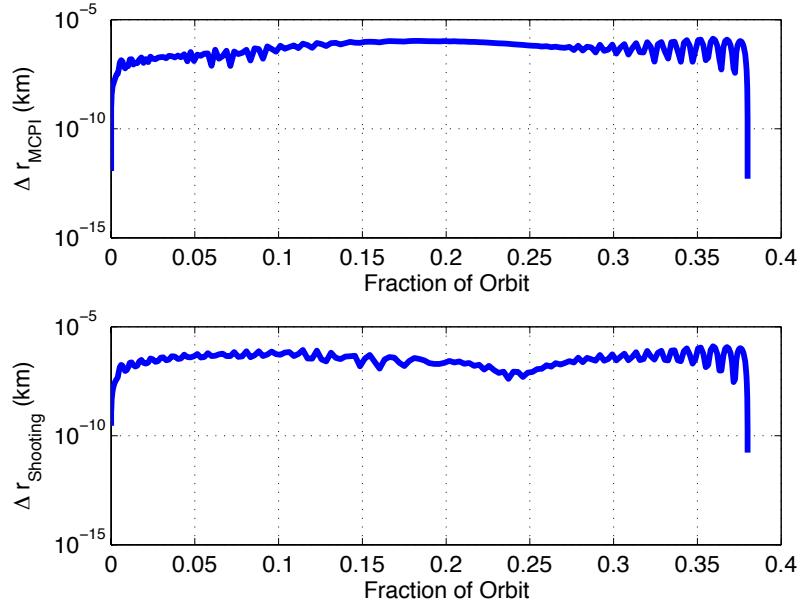


Fig. 7. Position errors of the MCPI algorithm (top panel) and the shooting method (bottom panel) compared with the analytic F and G solution.

5. CONCLUSION

Modified Chebyshev Picard Iteration (MCPI) is an iterative numerical method for approximating solutions of linear or non-linear Ordinary Differential Equations (ODEs). Unlike other step-by-step differential equation solvers, like the Runge-Kutta family, MCPI approximates long arcs of the state trajectory with an iterative path approximation approach, and is ideally suited to parallel computation. Orthogonal Chebyshev Polynomials are used as basis functions during each path iteration, and the integrations of the Picard iteration are then carried out analytically. The orthogonality of the Chebyshev basis functions allows the least square approximations to be computed without

matrix inversion. Instead the coefficients are computed robustly from discrete inner products. The discrete sampling and weighting that is adopted to satisfy the inner product definition creates that added benefit that the approximation errors are minimized near the ends of the interval.

The MCPI algorithm utilizes a vector-matrix framework for computational efficiency. All Chebyshev coefficients and integrand function evaluations are independent, meaning they can be simultaneously computed in parallel for further decreased computational cost. Over an order of magnitude speedup from traditional methods is achieved in serial processing, and an additional order of magnitude is achievable in parallel architectures.

In this paper we have presented the new TAMU MCPI library that allows the user to easily apply the MCPI method to their own ODE systems. The TAMU MCPI library is available in Matlab, C++, and as Matlab wrappers for CUDA parallel computation. It is fully cross-platform for Windows, Linux, and Apple, and can be compiled from source by the user, or distributed as a binary library for many common operating systems. The idea is that the user does not need to concern themselves with the underlying mathematics behind the MCPI algorithm, but simply inputs the boundary conditions of the dynamical system, the integrand function governing system behavior, and the desired time interval of integration. The algorithm outputs the time history of the system states over the interval of interest.

Two astrodynamical examples are presented to demonstrate the capability of the algorithm for the initial value and boundary value problems respectively. For the first example (IVP) we forward propagate a simulated cloud of debris particles in a low earth orbit. Compared to a native Matlab ODE45 integrator, we are able to forward propagate the motion five times faster with the same accuracy. For the second example (BVP) we consider Lambert's problem and present a convergence arc length of 38% of the orbit. Depending upon the arc-length of the orbit in the Lambert's problem, MCPI is able to obtain a solution 20-60 times faster than the shooting method. We have demonstrated the power of our MCPI algorithm in numerous publications, and we are excited at the prospect of sharing this new library to afford other researchers the opportunity to benefit from these tools.

6. REFERENCES

1. C. W. Clenshaw and H. J. Norton, *The solution of Nonlinear Ordinary Differential Equations in Chebyshev Series*, The Computer Journal, 6(1):88-92, 1963.
2. X. Bai, *Modified Chebyshev-Picard Iteration Methods for Solution of Initial Value and Boundary Value Problems*, Ph.D. dissertation, Texas A&M University, College Station, Tex, USA, 2010.
3. X. Bai and J. L. Junkins, *Modified Chebyshev-Picard Iteration Methods for Solution of Initial Value Problems*, Advances in the Astronautical Sciences, vol. 139, pp. 345–362, 2011.
4. X. Bai and J. L. Junkins, *Modified Chebyshev-Picard Iteration Methods for Solution of Boundary Value Problems*, Advances in the Astronautical Sciences, vol. 140, pp. 381–400, 2011.
5. X. Bai and J. L. Junkins, *Modified Chebyshev Picard Iteration Methods for Station-Keeping of Translunar Halo Orbits*, Mathematical Problems in Engineering, vol. 2012, Article ID 926158, 2012.
6. John L. Junkins, Ahmad Bani Younes, Robyn M. Woollards and Xiaoli Bai, *Orthogonal Approximation in Higher Dimensions: Applications in Astrodynamics*, ASS 12-634, Jer-nan Juang Astrodynamics Symposium, College Station, TX, June 14-26, 2012.
7. John L. Junkins, Ahmad Bani Younes, Robyn M. Woollards and Xiaoli Bai, *Orthogonal Approximation in Higher Dimensions: Applications in Astrodynamics*, submitted to The Journal of the Astronautical Sciences, June, 2013.
8. John L. Junkins, Ahmad Bani Younes, Robyn M. Woollards and Xiaoli Bai, *Picard Iteration, Chebyshev Polynomial and Chebyshev Picard Methods: Application in Astrodynamics*, accepted in The Journal of the Astronautical Sciences, July, 2013.

MULTI-SEGMENT ADAPTIVE MODIFIED CHEBYSHEV PICARD ITERATION METHOD

Donghoon Kim,* John L. Junkins,[†] James D. Turner,[‡] and Ahmad Bani-Younes[§]

A modified Cheyshev Picard iteration method is proposed for solving orbit propagation initial value problems. Cosine sampling, known as Chebyshev-Gauss-Labatto (CGL) node, is used to reduce the Runge's phenomenon that plagues many series approximations. The key benefit of using the CGL data sampling is that the nodal points are distributed non-uniformly, with dense sampling at the beginning and end times. This problem can be addressed by a nonlinear time transformation and/or by utilizing multiple time segments over an orbit. This paper suggests a method, called a multi-segment method, to obtain accurate solutions overall regardless of initial positions and eccentricity by dividing the given orbit into two or more segments.

INTRODUCTION

Modified Chebyshev Picard Iteration (MCPI) is an iterative numerical method for approximating solutions of linear or nonlinear ordinary differential equations to obtain time histories of system state trajectories. In contrast to many step-by-step integrators, the MCPI algorithm approximates long arcs of the state trajectory with an iterative path approximation approach and is ideally suited to parallel computation.¹ It is well known that Picard iteration has theoretical guarantees for converging to the desired solution. The rate of convergence of Picard iteration is geometric rather than quadratic for Jacobian based methods, however, the case for parallelization provides a significant advantage.²

Orthogonal Chebyshev Polynomials are used as basis functions during each path iteration in our approach, and the integrations of Picard iteration are then performed analytically. The orthogonality of the Chebyshev basis functions implies that the least square approximations can be computed to arbitrary precision without a matrix inversion; the coefficients are conveniently and robustly computed from discrete inner products.³ Similar approximation approaches that use Legendre polynomials are not successful because the starting and ending points of the fits are not sampled as densely as the MCPI algorithm. The MCPI algorithm utilizes a vector-matrix framework for computation

*Postdoctoral Research Associate, Aerospace Engineering, Texas A&M University, 741D H.R. Bright Bldg., College Station, TX 77843-3141, E-mail: AEROSPACE38@GMAIL.COM

[†]Distinguished Professor, Aerospace Engineering, Texas A&M University, 722B H.R. Bright Bldg., College Station, TX 77843-3141, AAS Fellow, Honorary Fellow AIAA. E-mail: JUNKINS@TAMU.EDU

[‡]Research Professor, Aerospace Engineering, Texas A&M University, 745 H.R. Bright Bldg., College Station, TX 77843-3141, E-mail: JDTURNER@TAMU.EDU

[§]Assistant Professor, Aerospace Engineering, Khalifa University, Abu Dhabi, UAE, E-mail: AH-MAD.YOUNES@KUSTAR.AC.AE

efficiency. Additionally, all Chebyshev coefficients and integrand function evaluations are independent, meaning they can be simultaneously computed in parallel for further decreased computational cost.¹

For the MCPI algorithm, the cosine sampling, which is known as Chebyshev-Gauss-Lobatto (CGL) node, is utilized to reduce the Runge's phenomenon. The Runge's phenomenon is a problem of oscillation at the edges of an interval that occurs when using polynomial interpolation with polynomials of high degree. Since dense sample points are distributed at the beginning and ending locations, less accurate solutions can be obtained where sparse points are distributed.

For example, let us consider an unperturbed two-body problem where the initial position is not located near the periapsis (See Fig. 1). Obviously, large errors can be observed near the periapsis where dense points are required but sparse points are distributed.

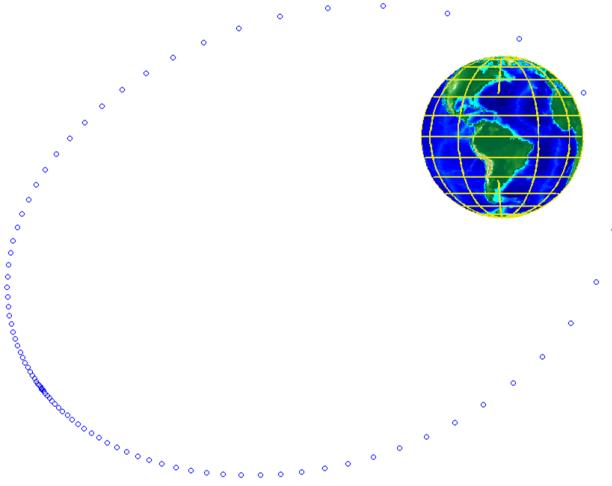


Figure 1. Sparse Point Distribution Description at Perigee

This problem is overcome by introducing a multi-segment method. Two and three segmented orbits are considered and compared with the general MCPI algorithm. The performance of the proposed approach is described by the numerical examples through a solution of the two-body problem.

MODIFIED CHEBYSHEV PICARD ITERATION

The MCPI algorithm combines the discoveries of two great mathematicians: Émile Picard (Picard iteration) and Rafnuty Chebyshev (Chebyshev polynomials). Combing these techniques was first proposed by Clenshaw and Norton in 1963.⁴

Picard stated that any first order differential equation

$$\frac{dx}{dt} = f(t, x) \quad (1)$$

with an initial condition $x(t_0) = x_0$ can be rearranged without approximation to obtain the following:

$$x(t) = x_0 + \int_{t_0}^t f(\tau, x(\tau)) d\tau \quad (2)$$

In the MCPI algorithm, orthogonal Chebyshev polynomials are used as basis functions to approximate the integrand in the Picard integral. Chebyshev polynomials reside in the domain $\tau = [-1, 1]$, and are defined recursively as:

$$\mathbf{x}^i(t) = \mathbf{x}_0 + \int_{t_0}^t \mathbf{f}(\tau, \mathbf{x}^{i-1}(\tau)) d\tau \quad (3)$$

The system dynamics are normalized such that the time span of integration is projected onto the domain of the Chebyshev polynomials, thus the system states are approximated using the Chebyshev polynomial basis functions. The orthogonal nature of the basis functions means that the coefficients that linearly scale the basis functions are computed independently as simple ratios of inner products with no matrix inversion.

A key feature of the MCPI algorithm is a nonuniform cosine density sampling of the domain of the Chebyshev basis functions called CGL nodes defined as follows:

$$T_0(\tau) = 1, \quad T_1(\tau) = \tau, \quad T_{k+1}(\tau) = 2\tau T_k(\tau) - T_{k-1}(\tau) \quad (4)$$

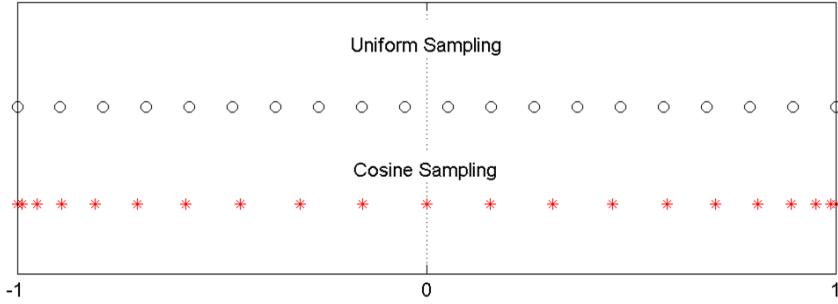


Figure 2. Uniform and Cosine Sampling Description in τ -Domain

As shown in Fig. 2, this sampling scheme has much higher density towards the edges, which enables a higher accuracy solution near the boundaries of the state trajectory. This scheme eliminates the Runge's phenomenon, a common issue in function approximation whereby noisy estimates are returned near the edges due to lack of knowledge of the states on the other sides of the boundaries. The coefficients multiplying the Chebyshev basis functions are approximated by the method of least squares, which generally requires a matrix inversion but the inverse is trivial.

A full derivation of the MCPI algorithm is not included in this work (Refer to Bai and Junkins¹). Instead, the authors present a flowchart in Fig. 3 briefly summarizing the mathematics underlying the MCPI algorithm for solution of initial value problems.

MULTI-SEGMENT APPROACH FOR MCPI ALGORITHM

For the MCPI algorithm, the CGL node is utilized to reduce the Runge's phenomenon, which is a problem of oscillation at the edges of an interval that occurs when using polynomial interpolation with polynomials of high degree. Since dense sample points are distributed at the beginning and ending locations, less accurate solutions are obtained where sparse points are distributed.

Known $\mathbf{x}(t_0) \equiv \mathbf{x}_0$, the 1st order ODE: $\frac{d\mathbf{x}}{dt} = \mathbf{f}(t, \mathbf{x}(t)), \quad t = [t_0, t_f]$



Variable change from t domain to τ domain: $t = \underline{a} + \underline{b}\tau$,
 $\tau = [-1, 1]$ where $\underline{a} = (t_f + t_0)/2$, $\underline{b} = (t_f - t_0)/2$



Transformed-domain 1st order ODE: $\frac{d\mathbf{x}}{d\tau} = \underline{b}\mathbf{f}(\underline{a} + \underline{b}\tau, \mathbf{x}(\underline{a} + \underline{b}\tau)) \equiv \mathbf{g}(\tau, \mathbf{x})$



Picard iteration:

$$\mathbf{x}^i(\tau) = \mathbf{x}_0 + \int_{-1}^{\tau} \mathbf{g}(s, \mathbf{x}^{i-1}(s)) ds, \quad i = 1, 2, \dots$$

Starting guess:

$$i = 1, \quad \mathbf{x}^0(\tau)$$



Discrete orthogonal approximation along the $(i-1)^{\text{th}}$ trajectory:

$$\begin{aligned} \mathbf{g}(\tau, \mathbf{x}^{i-1}(\tau)) &\cong \sum_{k=0}^{N-1} \mathbf{F}_k^{i-1} T_k(\tau) \quad \text{where } \begin{cases} \mathbf{F}_k^{i-1} = \frac{1}{c_k} \sum_{j=0}^N w_j \mathbf{g}(\tau_j, \mathbf{x}^{i-1}(\tau_j)) T_k(\tau_j) \\ \tau_j = -\cos(j\pi/N) \\ c_0 = N; c_k = N/2 \text{ for } k = 1, 2, \dots, N \\ w_0 = w_N = 1/2; w_j = 1 \text{ for } j = 1, 2, \dots, N-1 \end{cases} \end{aligned}$$

Enables the $(i-1)^{\text{th}}$ Picard integrals to be analytically approximated:

$$\mathbf{x}^i(\tau) = \mathbf{x}_0 + \int_{-1}^{\tau} \mathbf{g}(s, \mathbf{x}^{i-1}(s)) ds \cong \mathbf{x}_0 + \sum_{r=0}^{N-1} \mathbf{F}_r^{i-1} \int_{-1}^{\tau} T_r(s) ds \equiv \sum_{k=0}^N \boldsymbol{\beta}_k^i T_k(\tau)$$



Trajectory approximation update: $i = i + 1$,

$$\begin{aligned} \boldsymbol{\beta}_0^i &= \mathbf{x}_0 + \sum_{k=0}^N (-1)^{k+1} \boldsymbol{\beta}_k^i \\ \mathbf{x}^i(\tau) &= \sum_{k=0}^N \boldsymbol{\beta}_k^i T_k(\tau) \quad \text{where } \begin{cases} \boldsymbol{\beta}_r^i = \frac{1}{2r} (\mathbf{F}_{r-1}^{i-1} - \mathbf{F}_{r+1}^{i-1}), \quad r = 1, 2, \dots, N-2 \\ \boldsymbol{\beta}_{N-1}^i = \frac{\mathbf{F}_{N-2}^{i-1}}{2(N-1)}, \quad \boldsymbol{\beta}_N^i = \frac{\mathbf{F}_{N-1}^{i-1}}{2N} \end{cases} \end{aligned}$$

Figure 3. Flowchart for MCPI Algorithms for Solution of Initial Value Problems

For example, let us consider an unperturbed two-body problem where the initial position is not located near the periapsis. Obviously, large errors are observed near the periapsis where dense points are required but sparse points are distributed. Additionally, even though the initial position is located near the periapsis, accurate solutions can't be obtained for highly elliptical orbits. To obtain accurate solutions for the above cases using the MCPI algorithm, a multi-segment approach is proposed.

Given the initial true anomaly (f_0), two and three segmented orbits are considered as shown in Fig. 4. These two cases requires patch times to link the divided segments. To distribute dense points near the periapsis, the following strategies are suggested.

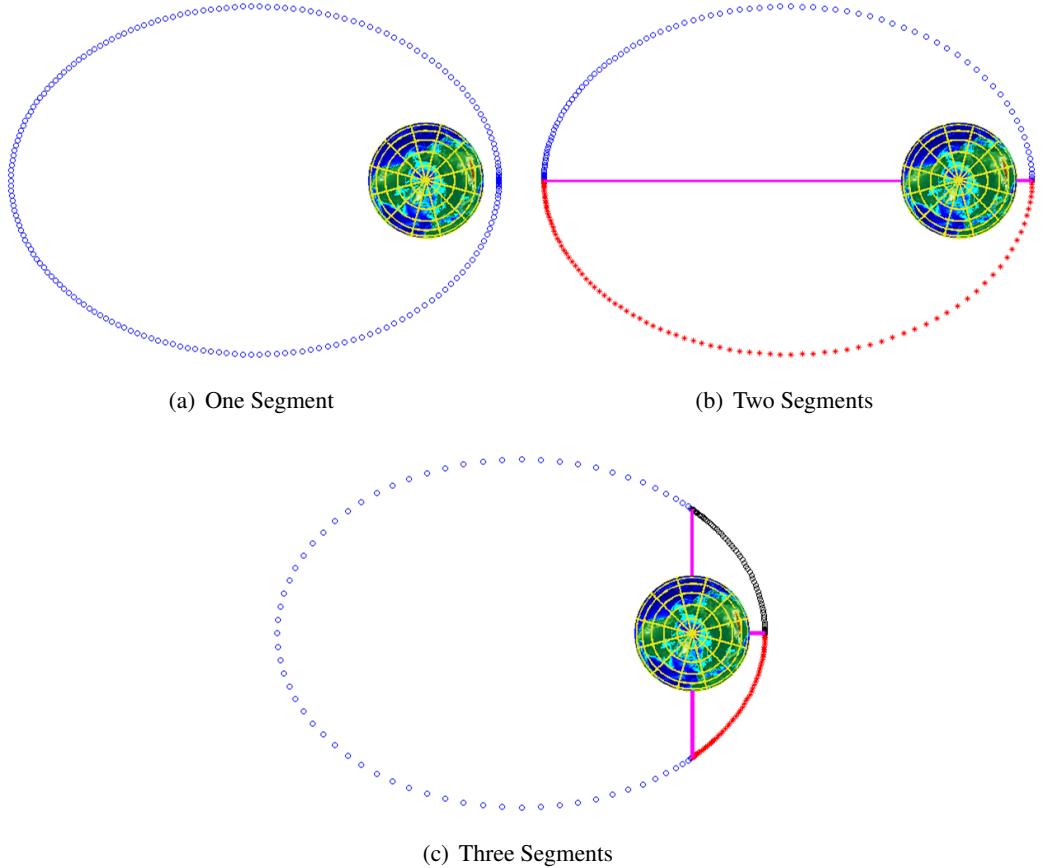


Figure 4. Segmented Orbit Descriptions

For two segmented orbits, the time for the patch point is selected at the time at the perigee, where the true anomaly (f) is 0 degree. For three segmented orbits, the time for the first patch point is selected where $f = -f_0$ degree for symmetry and the time for the second patch point is selected where $f = 0$ degree. To find propagation time for each segment, the following calculation needs to be performed. First, given the initial position and velocity vectors, calculate the initial true anomaly and one orbit period time (T_P) as follows:

$$T_P = 2\pi \sqrt{\frac{a^3}{\mu}} \quad (5)$$

where a is the semi-major and μ is the Earth gravitational constant.

Second, calculate the initial mean anomaly (M_0) as follows:

$$M_0 = E_0 - e \sin E_0 \quad (6)$$

where e is the eccentricity and E_0 is the initial eccentric anomaly described as

$$E_0 = 2 \tan^{-1} \left[\sqrt{\frac{1-e}{1+e}} \tan \left(\frac{f_0}{2} \right) \right] \quad (7)$$

Finally, calculate the propagation time for each segment as follows:

$$T_{P_1} = T_P - (S - 1)T_{P_2}, \quad T_{P_2} = M_0 \sqrt{\frac{a^3}{\mu}} \quad (8)$$

where S is the number of segment for one orbit propagation.

For the considered cases, two sets of propagation time are determined as follows:

$$\begin{cases} \text{Two Segments: } & T_P = [T_{P_1}, T_{P_2}] \\ \text{Three Segments: } & T_P = [T_{P_1}, T_{P_2}, T_{P_2}] \end{cases} \quad (9)$$

NUMERICAL EXAMPLES

A satellite motion integration problem, where only the gravitational force from the Earth, is considered. The three-dimensional dynamical equations are

$$\ddot{x} = -\frac{\mu}{r^3}x, \quad \ddot{y} = -\frac{\mu}{r^3}y, \quad \ddot{z} = -\frac{\mu}{r^3}z \quad (10)$$

where x , y , and z are the three coordinates in Earth-centered inertial reference frame; r is the distance of the satellite from the Earth; and the Earth gravitational constant μ is chosen as $3.98600433 \times 10^{14} \text{ m}^3/\text{s}^2$.

To verify the results, the following normalized energy error check is utilized:

$$\mathcal{E}_{\text{error}} = \frac{|\mathcal{E} - \mathcal{E}_0|}{|\mathcal{E}_0|} \quad (11)$$

where \mathcal{E}_0 is the initial energy and the energy is calculated as follows:

$$\mathcal{E} = \frac{1}{2} (\dot{x}^2 + \dot{y}^2 + \dot{z}^2) - \frac{\mu}{r} \quad (12)$$

Note that the goal is to obtain solutions where $\mathcal{E}_{\text{err}} < 10^{-13}$.

Two sets of initial position and velocity vectors

$$\begin{cases} \mathbf{r}(t_0) = [-0.9085, -0.0652, 1.0328]^T \times 10^7 \text{m} \\ \mathbf{v}(t_0) = [-4.8283, -4.4242, 0.4949]^T \times 10^3 \text{m/s} \end{cases} \quad (13)$$

$$\begin{cases} \mathbf{r}(t_0) = [-1.9994, -3.6222, -1.9875]^T \times 10^7 \text{m} \\ \mathbf{v}(t_0) = [1.0649, 0.0765, -1.2106]^T \times 10^3 \text{m/s} \end{cases} \quad (14)$$

Table 1. Classical Orbital Elements

| Parameter | Symbol | Value | Unit |
|---------------------------------------|----------|----------------------|--------|
| Semi-major axis | a | 2.7×10^7 | m |
| Eccentricity | e | 0.7 | - |
| Inclination | i | 60 | degree |
| Right ascension of the ascending node | Ω | 45 | degree |
| Argument of periapsis | w | 30 | degree |
| Orbit period | T_P | 4.4153×10^4 | s |

which lead to initial true anomalies $f_0 = 90$ and 180 degrees, respectively, and the classical orbital elements listed in Table 1.

For the MCPI algorithm implementation to solve this problem, various factors need to be determined in prior calculation: 1) maximum iteration number (I_M), 2) error tolerance (T_E), 3) degree of polynomial (N), and 4) number of sample points (M). In this work, the authors focus on finding a methodology to improve MCPI accuracy and reduce computational burden given the factors listed in Table 2 and the initial conditions.

Table 2. Tuning Parameters

| Parameter | Symbol | Value | Unit |
|--|-----------------|------------|------|
| Maximum iteration number | I_M | 200 | - |
| Error tolerance | T_E | 10^{-13} | - |
| Degree of polynomial | N | 200 | - |
| Number of sample points (One segment) | M | 200 | - |
| Number of sample points (Two segments) | M_1, M_2 | 100, 100 | - |
| Number of sample points (Three segments) | M_1, M_2, M_3 | 68, 66, 66 | - |

Numerical simulations are performed and the normalized energy error results are shown in Figs. 5-8. Figure 5 shows that the normalized energy errors are much larger than the requirement ($\mathcal{E}_{\text{err}} < 10^{-13}$). Obviously, largest error is observed at periapsis when $f_0 = 180$ degree because of sparse point distribution at the periapsis.

Figure 6 shows that the solution satisfies the requirement when $f_0 = 180$ degree. Same number of sample points are distributed for each segment and the total number of the sample points are equal to the number of sample points for the basic (one segment) MCPI algorithm. *Note that only two segmented orbit approach is applicable when the initial position is not located at periapsis.*

Figure 7 shows that the solution satisfies the requirement when $f_0 = 90$ degree. Same number of sample points are distributed for the second and third segments and the total number of the sample points are equal to the number of sample point for the basic MCPI algorithm.

For the case where $f_0 = 90$ degree, both approaches, two and three segmented orbits, are applicable. As shown in Fig. 8, both approaches satisfy the requirement but the three segmented orbit approach outperforms most other method. The number of distributions for each approach is determined by try and error and a methodology to select best number of distribution is under development.

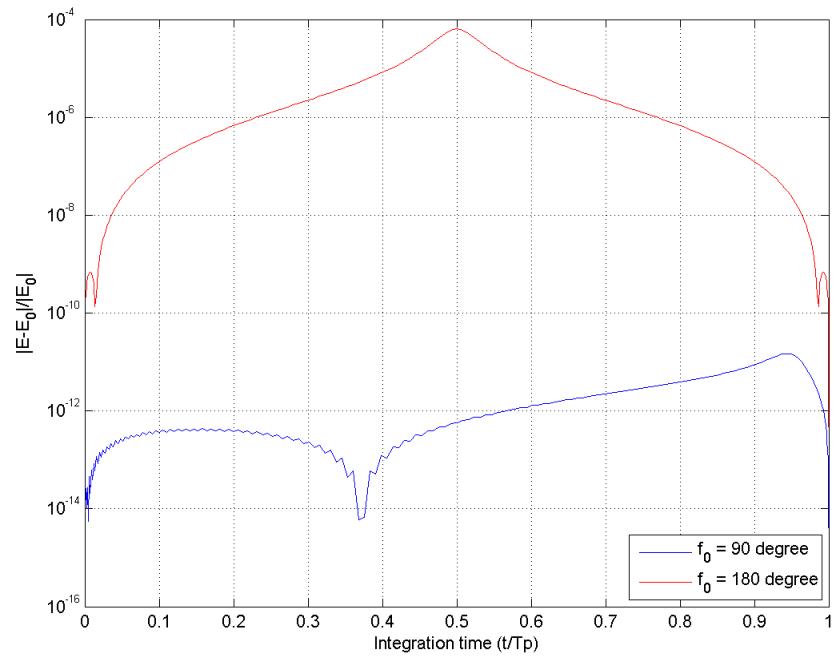


Figure 5. Energy Error (One Segment)

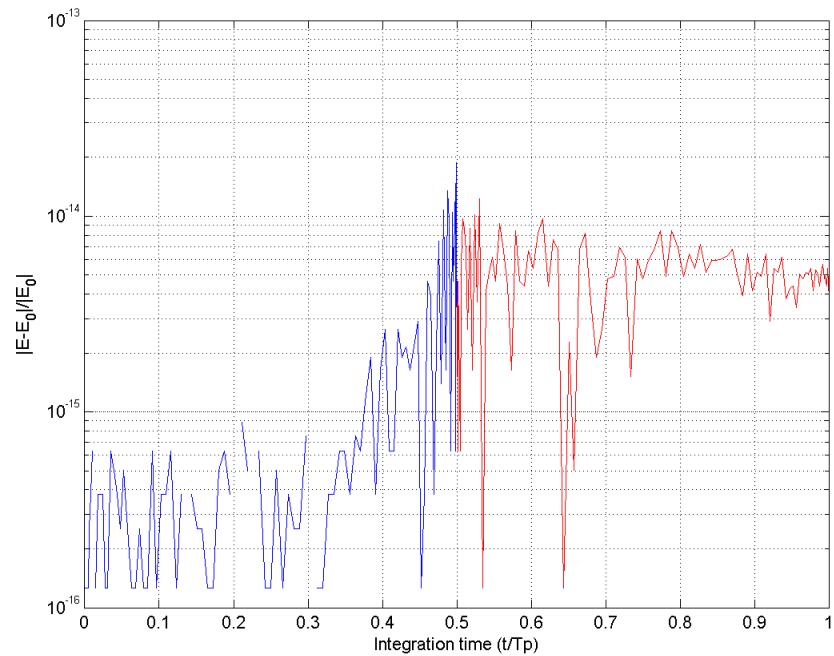


Figure 6. Energy Error (Two Segments, $f_0 = 180$ degree)

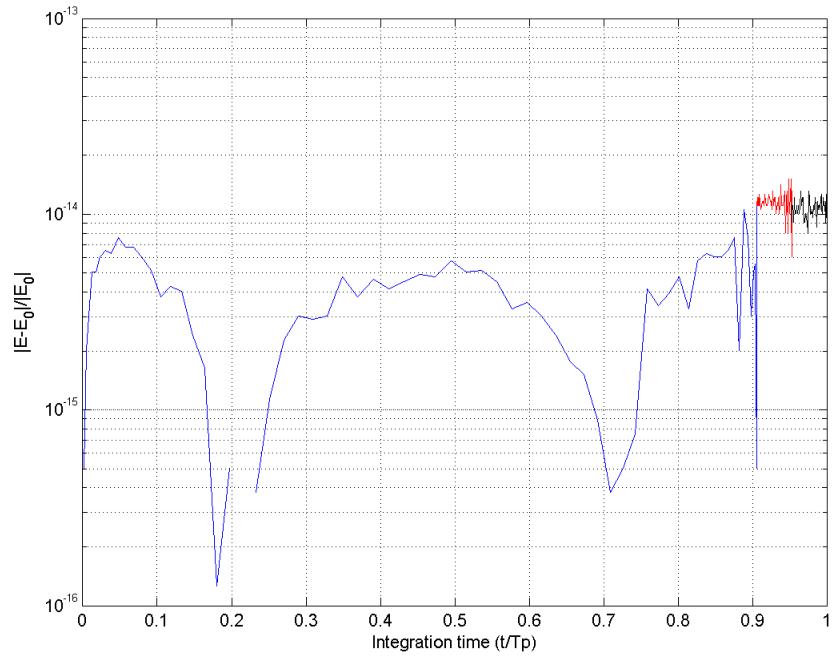


Figure 7. Energy Error (Three Segments, $f_0 = 90$ degree)

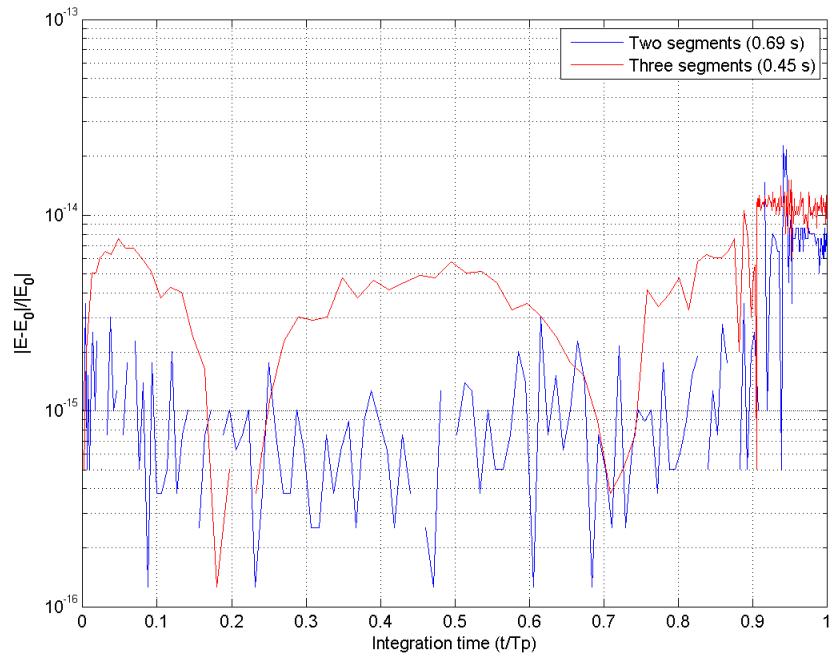


Figure 8. Energy Error Comparison between Two and Three Segments ($f_0 = 90$ degree)

CONCLUSION

The modified Chebyshev Picard iteration (MCPI) algorithm uses Chebyshev-Gauss-Lobatto (CGL) node to reduce the Runge's phenomenon. However, by using the CGL node, less accurate solutions are obtained where sparse points are distributed. For the unperturbed two-body problem, the multi-segment approach is utilized to obtain accurate solution. As a result, the multi-segment approach provides much more accurate solutions comparing to the basic MCPI solution. Moreover, the authors show that the three segmented approach outperforms other method in terms of computational burdens. This approach will be very useful when the initial position is not located near the periapsis for the MCPI algorithm.

REFERENCES

- [1] X. Bai, *Modified Chebyshev-Picard Iteration Methods for Solution of Initial Value and Boundary Value Problems*. PhD thesis, Department of Aerospace Engineering, Texas A&M University, College Station, TX, 2010.
- [2] X. Bai and J. L. Junkins, "Modified Chebyshev-Picard Iteration Methods for Solution of Initial Value Problems," *Journal of the Astronautical Sciences*, Vol. 59, January-June 2012, pp. 335–359, 10.1007
- [3] A. Bani-Younes, *Orthogonal Polynomial Approximation in Higher Dimensions: Applications in Astrodynamics*. PhD thesis, Department of Aerospace Engineering, Texas A&M University, College Station, TX, 2013.
- [4] C. W. Clenshaw and H. J. Norton, "The Solution of Nonlinear Ordinary Differential Equations in Chebyshev Series," *The Computer Journal*, Vol. 6, No. 1, 1963, pp. 88–92.

Research Article

Multisegment Scheme Applications to Modified Chebyshev Picard Iteration Method for Highly Elliptical Orbits

Donghoon Kim,¹ John L. Junkins,² and James D. Turner¹

¹ Aerospace Engineering, Texas A&M University, College Station, TX 77843-3141, USA

² Royce Wisenbaker Chair in Engineering, Aerospace Engineering, Texas A&M University, College Station, TX 77843-3141, USA

Correspondence should be addressed to Donghoon Kim; aerospace38@gmail.com

Received 10 April 2014; Accepted 22 July 2014

Academic Editor: Ker-Wei Yu

Copyright © 2014 Donghoon Kim et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A modified Chebyshev Picard iteration method is proposed for solving orbit propagation initial/boundary value problems. Cosine sampling techniques, known as Chebyshev-Gauss-Lobatto (CGL) nodes, are used to reduce Runge's phenomenon that plagues many series approximations. The key benefit of using the CGL data sampling is that the nodal points are distributed nonuniformly, with dense sampling at the beginning and ending times. This problem can be addressed by a nonlinear time transformation and/or by utilizing multiple time segments over an orbit. This paper suggests a method, called a multisegment method, to obtain accurate solutions overall regardless of initial states and albeit eccentricity by dividing the given orbit into two or more segments based on the true anomaly.

1. Introduction

A modified Chebyshev Picard iteration (MCPI) is an iterative numerical method for approximating solutions of linear or nonlinear ordinary differential equations to obtain time histories of system state trajectories [1, 2]. In contrast to many step-by-step integrators, the MCPI algorithm approximates long arcs of the state trajectory with an iterative path approximation approach and is ideally suited to parallel computation [3]. It is well known that Picard iteration has theoretical guarantees for converging to the solution assuming the forces are continuous, once differentiable, and the solution of the differential equation is unique [4]. The rate of convergence of Picard iteration is geometric rather than quadratic for Jacobian based methods. However, given a good starting approximation, excellent efficiency is possible, and the ease for parallelization provides a significant advantage [5, 6].

Orthogonal Chebyshev polynomials are used as basis functions during each path iteration, and the integrations of Picard iteration are then performed analytically. The orthogonality of the Chebyshev basis functions implies that the least-square approximations can be computed to arbitrary precision without a matrix inversion; the coefficients are

conveniently and robustly computed from discrete inner products [7]. Similar approximation approaches that use Legendre polynomials can be utilized, but the authors obtain slightly better results because the starting and ending points of the fits are not sampled as densely as the MCPI algorithm, and importantly the location of the nodes for the Chebyshev basis functions is computed exactly without iterations. The MCPI algorithm utilizes a vector-matrix framework for computational efficiency. Additionally, all Chebyshev coefficients and integrand function evaluations are independent, meaning that they can be simultaneously computed in parallel for further decreased computational costs [3].

For the MCPI algorithm, the cosine sampling techniques, known as Chebyshev-Gauss-Lobatto (CGL) nodes [8], are utilized to reduce Runge's phenomenon. The Runge phenomenon is a problem of oscillation at the edges of an interval that occurs when using polynomial interpolation with polynomials of high degree [9]. Since dense sample points are distributed at the beginning and ending locations, less accurate solutions are usually obtained where sample points are more uniformly distributed [10].

For the most extreme counterexample, let us consider an unperturbed two-body problem, where the initial position is

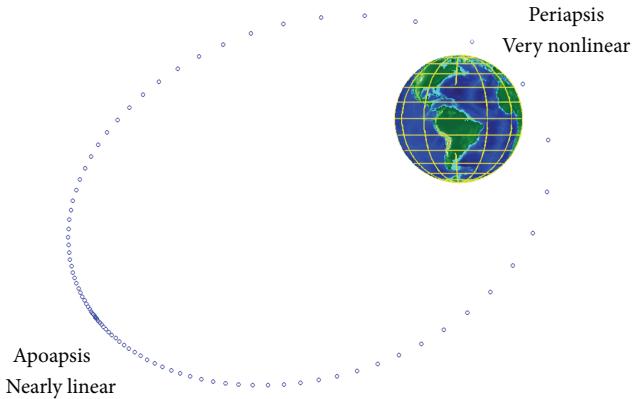


FIGURE 1: Sparse sample point distribution description at periapsis.

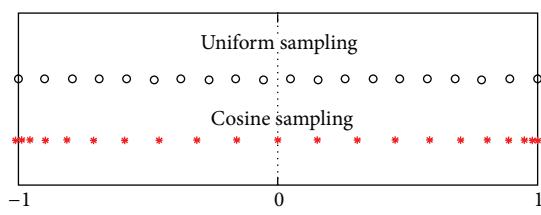


FIGURE 2: Uniform and cosine sampling descriptions.

TABLE 1: Classical orbital elements.

| Parameter | Symbol | Value | Unit |
|---------------------------------------|----------|----------------------|--------|
| Semimajor axis | a | 2.7×10^7 | m |
| Eccentricity | e | 0.7 | — |
| Inclination | i | 60 | Degree |
| Right ascension of the ascending node | Ω | 45 | Degree |
| Argument of periapsis | w | 30 | Degree |
| Orbit period | T_p | 4.4153×10^4 | s |

not located near the periapsis. Obviously, large errors can be observed near the periapsis due to sparse sample points where the dynamics are most nonlinear, yet we waste the dense sample points at apoapsis when the problem is most linear as shown in Figure 1.

This problem is overcome by introducing a multisegment method and the results are compared with the basic MCPI algorithm. This paper only considers two and three segments per one orbit. The performance of the proposed approach is established by numerical examples of the two-body problem.

2. Modified Chebyshev Picard Iteration

The MCPI algorithm combines the discoveries of two great mathematicians: Émile Picard (Picard iteration) and Rafnuty Chebyshev (Chebyshev polynomials). Combing these techniques was first proposed by Clenshaw and Norton in 1963 [11].

TABLE 2: Tuning parameters.

| Parameter | Symbol | Value | Unit |
|--|-----------------|------------|------|
| Maximum iteration number | I_M | 50 | — |
| Error tolerance | T_E | 10^{-13} | — |
| Degree of polynomial | N | 200 | — |
| Number of sample points (one segment) | M | 200 | — |
| Number of sample points (two segments) | M_1, M_2 | 100, 100 | — |
| Number of sample points (three segments) | M_1, M_2, M_3 | 68, 66, 66 | — |

Picard stated that any first-order differential equation

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(t, \mathbf{x}(t)) \quad (1)$$

with an initial condition $\mathbf{x}(t_0) = \mathbf{x}_0$ can be rearranged without approximation as follows:

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

In the MCPI algorithm, orthogonal Chebyshev polynomials are used as basis functions to approximate the integrand in the Picard integral. Chebyshev polynomials reside in the domain $\tau = [-1, 1]$ and are defined recursively as

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

The system dynamics are normalized such that the time span of integration is projected onto the domain of the Chebyshev polynomials. Thus, the system states are approximated using the Chebyshev polynomial basis functions. The orthogonal nature of the basis functions means the coefficients that linearly scale the basis functions are computed independently as simple ratios of inner products without requiring matrix inversions.

A key feature of the MCPI algorithm is a nonuniform cosine density sampling of the domain of the Chebyshev basis functions, called CGL nodes, defined as follows:

$$\begin{aligned} T_0(\tau) &= 1, \\ T_1(\tau) &= \tau, \\ T_{k+1}(\tau) &= 2\tau T_k(\tau) - T_{k-1}(\tau). \end{aligned} \quad (4)$$

This sampling scheme provides much higher density towards the edges (beginning and ending points), which enables high accuracy solutions near the boundaries of the state trajectory. This scheme eliminates the Runge phenomenon, a common issue in function approximations, whereby noisy estimates are returned near the edges due to lack of knowledge of the states on the other sides of the boundaries (see Figure 2). The coefficients multiplying the Chebyshev basis functions are approximated by the method

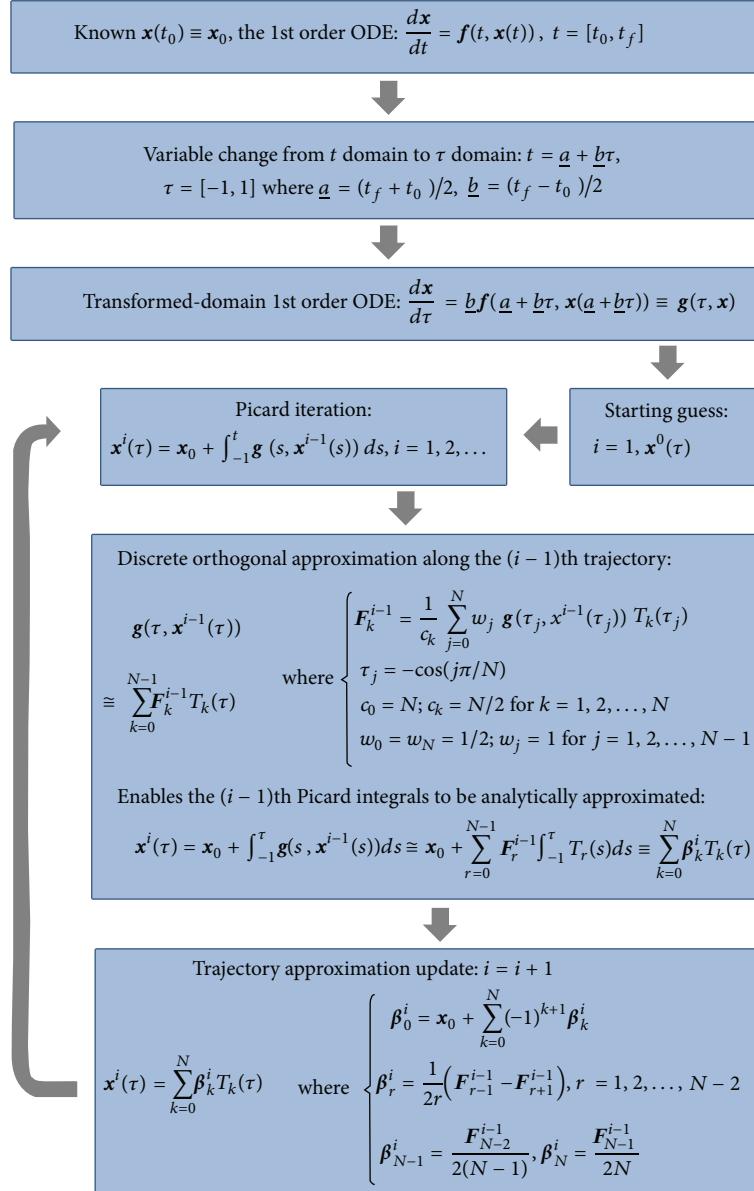


FIGURE 3: Flowchart for the MCPI algorithm for solution of initial value problems.

of least squares, which generally requires a matrix inversion. A wonderful side effect of the cosine sampling scheme is that the matrix required to be inverted in the normal equations of least squares is diagonal; thus the inverse is trivial.

A full derivation of the MCPI algorithm is not included in this work (refer to [Bai \[3\]](#)). Instead, the authors present a flowchart in Figure 3 briefly summarizing the mathematics underlying the MCPI algorithm for solution of initial value problems.

3. Multisegment Approach for MCPI Algorithm

This work considers an unperturbed two-body problem, where the initial position is not located near the periapsis. As

expected, large errors are observed near the periapsis where dense sample points are required, but sparse sample points are distributed. In addition, even though initial positions are located near the periapsis, accurate solutions cannot be obtained for highly elliptical orbits. To obtain accurate solutions for the above cases using the MCPI algorithm, the multisegment approach is proposed.

Given the initial true anomaly (f_0), two and three segmented orbits are considered as shown in Figure 4. These two cases require patch times to link the divided segments. To distribute dense sample points near the periapsis, several strategies are presented.

For two segmented orbits, the time for the patch point is selected as the time at the perigee, where the true anomaly (f) is 0 degrees. For three segmented orbits, the time for the first

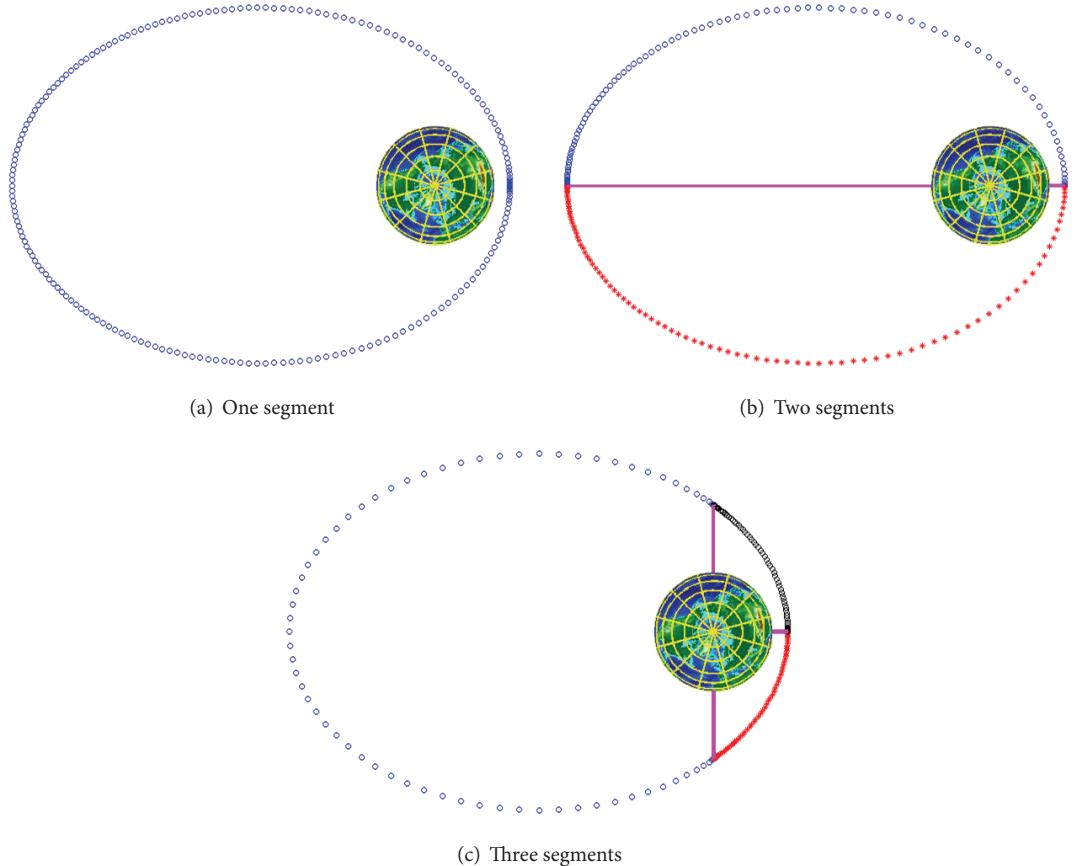


FIGURE 4: Segmented orbit descriptions.

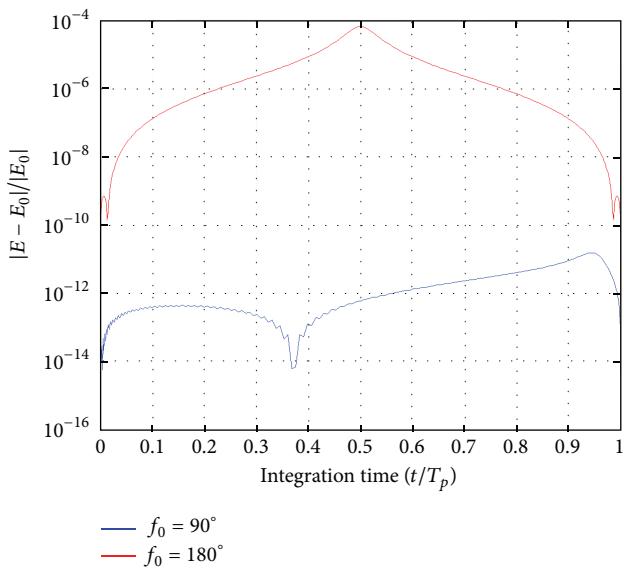


FIGURE 5: Time trajectories of the energy error; one segment.

patch point is selected where $f = -f_0$ degree for symmetry, and the time for the second patch point is selected where $f = 0$ degrees. To find propagation times for each segment, the

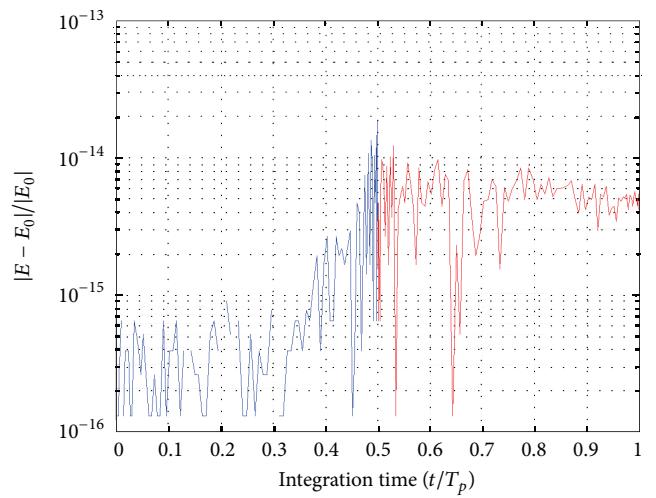


FIGURE 6: Time trajectories of the energy error and its comparison; two segments ($f_0 = 180$ degrees).

following calculation needs to be performed. First, given the initial position and velocity vectors, prescribe the break point f_0 and one orbit period time (T_p) as follows [12]:

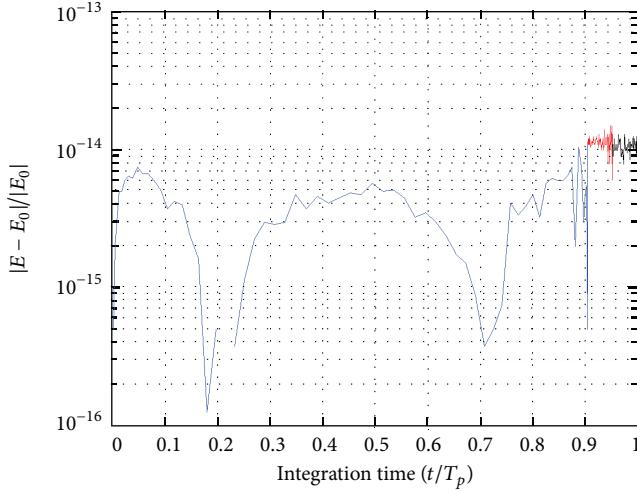


FIGURE 7: Time trajectories of the energy error and its comparison; three segments ($f_0 = 90$ degrees).

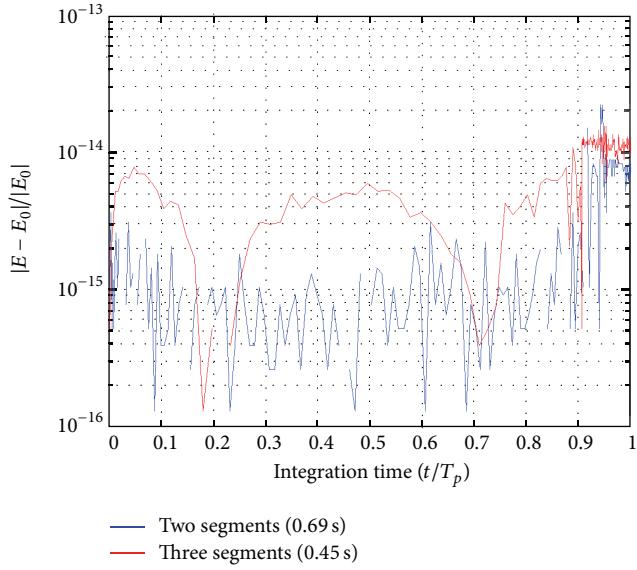


FIGURE 8: Time trajectories of the energy error and its comparison; two and three segments ($f_0 = 90$ degrees).

Finally, the propagation times for each segment are calculated as follows:

$$\begin{aligned} T_{P_1} &= T_P - (S - 1) T_{P_2}, \\ T_{P_2} &= M_0 \sqrt{\frac{a^3}{\mu}}, \end{aligned} \quad (8)$$

where $S > 1$ is the number of segments for the orbit.

The sets of propagation times are determined as follows:

$$\begin{aligned} \text{two segments : } \mathbb{T}_P &= [T_{P_1}, T_{P_2}], \\ \text{three segments : } \mathbb{T}_P &= [T_{P_1}, T_{P_2}, T_{P_3}]. \end{aligned} \quad (9)$$

For more than three segments and the associated break points, the above logic is readily extended. The optimization of the break points to achieve efficiency and accuracy is not addressed in this paper but is an easy-to-pose optimization problem research for a future study.

4. Numerical Examples

A satellite motion integration problem, idealized for the case with only the inverse square gravitational force from the Earth, is considered. The three-dimensional dynamical equations are given by [12]

$$\begin{aligned} \ddot{x} &= -\frac{\mu}{r^3} x, \\ \ddot{y} &= -\frac{\mu}{r^3} y, \\ \ddot{z} &= -\frac{\mu}{r^3} z, \end{aligned} \quad (10)$$

where x , y , and z are the three coordinates in Earth-centered inertial reference frame; r is the distance of the satellite from the Earth; μ is the Earth gravitational constant and is chosen as $3.98600433 \times 10^{14} \text{ m}^3/\text{s}^2$.

To verify the results, the following normalized energy error check is utilized:

$$\mathcal{E}_{\text{error}} = \frac{|\mathcal{E} - \mathcal{E}_0|}{|\mathcal{E}_0|}, \quad (11)$$

where \mathcal{E}_0 is the initial energy and the energy is calculated as follows:

$$\mathcal{E} = \frac{1}{2} (\dot{x}^2 + \dot{y}^2 + \dot{z}^2) - \frac{\mu}{r}. \quad (12)$$

Note that the goal for the demonstration example in this paper is to obtain solutions where $\mathcal{E}_{\text{err}} < 10^{-13}$. Moreover, for the unperturbed two-body problem, the analytical solution [12] for the F&G function can also be used to confirm the accuracy of the solution.

Second, calculate the initial mean anomaly (M_0) as follows [13]:

$$M_0 = E_0 - e \sin E_0, \quad (6)$$

where e is the eccentricity and the eccentric anomaly (E_0) is defined as [13]

$$E_0 = 2 \tan^{-1} \left[\sqrt{\frac{1-e}{1+e}} \tan \left(\frac{f_0}{2} \right) \right]. \quad (7)$$

Two sets of initial position and velocity vectors are given as follows:

$$\begin{aligned}\mathbf{r}(t_0) &= [-0.9085, -0.0652, 1.0328]^T \times 10^7 \text{ m}, \\ \mathbf{v}(t_0) &= [-4.8283, -4.4242, 0.4949]^T \times 10^3 \text{ m/s}, \\ \mathbf{r}(t_0) &= [-1.9994, -3.6222, -1.9875]^T \times 10^7 \text{ m}, \\ \mathbf{v}(t_0) &= [1.0649, 0.0765, -1.2106]^T \times 10^3 \text{ m/s}.\end{aligned}\quad (13)$$

The given initial states lead to $f_0 = 90$ and 180 degrees, respectively, and the classical orbital elements are listed in Table 1.

For the MCPI algorithm implementation to solve this problem, various tuning parameters are determined in a prior calculation: (1) maximum iteration number (I_M), (2) error tolerance (T_E), (3) degree of polynomial (N), and (4) number of sample points (M). This work focuses on finding a methodology to improve MCPI accuracy and reduce computational burden given the described factors in Table 2 and initial conditions listed in Table 1.

Numerical simulations are performed, and the normalized energy error results are shown in Figures 5–8. Figure 5 shows that the normalized energy errors are much larger than the requirement ($\mathcal{E}_{\text{err}} < 10^{-13}$). Obviously, the largest error is observed at the periapsis when $f_0 = 180$ degrees because of sparse sample point distributions at the periapsis.

Figure 6 shows that the solution satisfies the requirement when $f_0 = 180$ degrees using the two-segment scheme. The same number of sample points is distributed for each segment, and the total number of the sample points is equal to the number of sample points for the basic (one-segment) MCPI algorithm. Note that only the two-segment orbit approach is used when the initial position is located at apoapsis for symmetry.

Figure 7 shows that the solution satisfies the requirement when $f_0 = 90$ degrees using the three-segmentation scheme. The same number of sample points is distributed for the second and third segments, and the total number of the sample points is equal to the number of sample points for the basic MCPI algorithm.

For the case where $f_0 = 90$ degrees, both approaches such as the two- and three-segment schemes are applicable. As shown in Figure 8, both approaches satisfy the requirement, but the three-segment orbit approach outperforms the other methods. The number of nodes for each approach is determined by a heuristic method for this paper (and tuned numerically); and a methodology to select optimal number of nodes is under development.

5. Conclusion

The modified Chebyshev Picard iteration (MCPI) algorithm uses Chebyshev-Gauss-Lobatto (CGL) nodes to reduce the Runge phenomenon. By using the CGL nodes, however, less accurate solutions may be obtained where sparse sample points are distributed. Physical insights indicate that the

dense nodes should be located where the orbit is most nonlinear. However, the **stating epoch** state can be at a random point in the orbit. For the unperturbed two-body problem, where the initial state is not located near the periapsis and the eccentricity is high, the multisegment approach is utilized to obtain an accurate solution. The final perigee passage can be used to make all subsequent segment breaks symmetrical about the major axis. As a result, the multisegment approach provides much more accurate solutions when compared to the solution from the basic MCPI algorithm with random user-specified segmentation logic. Moreover, it is shown that the three-segment orbit approach outperforms **others** in terms of computational efficiency. To improve the performance of the MCPI algorithm, this approach will be very useful, especially when the initial position is not located near the periapsis and high eccentric orbits are given.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work is supported by the Air Force Office of Scientific Research, USA.

References

- [1] X. Bai and J. L. Junkins, “Modified chebyshev-picard iteration methods for orbit propagation,” *Journal of the Astronautical Sciences*, vol. 58, no. 4, pp. 583–613, 2011.
- [2] X. Bai and J. L. Junkins, “Modified Chebyshev-Picard iteration methods for solution of boundary value problems,” *Journal of the Astronautical Sciences*, vol. 58, no. 4, pp. 615–642, 2011.
- [3] X. Bai, *Modified Chebyshev-Picard iteration methods for solution of initial value and boundary value problems* [Ph.D. thesis], Department of Aerospace Engineering, Texas A&M University, College Station, Tex, USA, 2010.
- [4] L. Fox and I. B. Parker, *Chebyshev Polynomials in Numerical Analysis*, Oxford University Press, 1st edition, 1968.
- [5] X. Bai and J. L. Junkins, “Modified Chebyshev-Picard iteration methods for solution of initial value problems,” *Journal of the Astronautical Sciences*, vol. 59, no. 1-2, pp. 335–359, 2012.
- [6] X. Bai and J. L. Junkins, “Modified Chebyshev-Picard iteration methods for station-keeping of translunar halo orbits,” *Mathematical Problems in Engineering*, vol. 2012, Article ID 926158, 18 pages, 2012.
- [7] A. Bani-Younes, *Orthogonal polynomial approximation in higher dimensions: applications in astrodynamics* [Ph.D. thesis], Department of Aerospace Engineering, Texas A&M University, College Station, Richardson, Tex, USA, 2013.
- [8] S. A. Sarra, “Chebyshev interpolation: an interactive tour,” *Journal of Online Mathematics and its Applications*, pp. 1–13, 2006.
- [9] P. O. Runck and K. Mahler, “Über Konvergenzfragen bei Polynomialinterpolation mitäquidistanten Knoten. I,” *Journal für die reine und angewandte Mathematik (Crelles Journal)*, vol. 208, pp. 51–69, 1961.

- [10] D. Kim, J. L. Junkins, J. D. Turner, and A. Bani-Younes, "Multi-segment adaptive modified chebyshev picard iteration method," in *Proceedings of the 24th AAS/AIAA Space Flight Mechanics Meeting*, Santa Fe, NM, USA, January 2014, AAS Paper 14-232.
- [11] C. W. Clenshaw and H. J. Norton, "The solution of nonlinear ordinary differential equations in Chebyshev series," *The Computer Journal*, vol. 6, no. 1, pp. 88–92, 1963.
- [12] H. Schaub and J. L. Junkins, *Analytical Mechanics of Space Systems*, AIAA, Reston, Va, USA, 2nd edition, 2009.
- [13] R. H. Battin, *An Introduction to the Mathematics and Methods of Astrodynamics*, AIAA Education Series, AIAA: American Institute of Aeronautics and Astronautics, 1999.

Composition Comments

1. Please check and confirm the author(s) first and last names and their order which exist in the last page.
2. We redrew Figures 2 and 3. Please check.
3. We made the highlighted change according to the list of references. Please check.
4. We redrew some parts in Figures 5, 6, 7, and 8. Please check.
5. We made the highlighted change. Please check.
6. Please check the correctness of the highlighted part(s).
7. We added the highlighted part(s) for the sake of correctness. Please check.

Author(s) Name(s)

It is very important to confirm the author(s) last and first names in order to be displayed correctly on our website as well as in the indexing databases:

Author 1

Given Names: Donghoon
Last Name: Kim

Author 2

Given Names: John L.

Last Name: Junkins

Author 3

Given Names: James D.
Last Name: Turner

It is also very important for each author to provide an ORCID (Open Researcher and Contributor ID). ORCID aims to solve the name ambiguity problem in scholarly communications by creating a registry of persistent unique identifiers for individual researchers.

To register an ORCID, please go to the Account Update page (<http://mts.hindawi.com/update/>) in our Manuscript Tracking System and after you have logged in click on the ORCID link at the top of the page. This link will take you to the ORCID website where you will be able to create an account for yourself. Once you have done so, your new ORCID will be saved in our Manuscript Tracking System automatically.

A NEW SOLUTION FOR THE GENERALIZED LAMBERT'S PROBLEM

Robyn M. Woollards,^{*} John L. Junkins,[†] and Ahmad Bani Younes[‡]

A method is presented for solving boundary and initial value problems in celestial mechanics. In particular we consider the well-known Lambert TPBVP. The approach is quite general, however certain details in the transformed space boundary conditions pose challenges. We have been able to resolve these difficulties fully for the planar classical two-body problem, and we are engaged in a study to extend our numerical algorithm to the generally perturbed case. This method fuses three sets of ideas: (i) Picard Iteration, (ii) Orthogonal approximation, and notably, regularizing transformation of the equations of motion. Curiously, we find that a local-linearization-based shooting is not required, and we also illustrate that the method is not highly sensitive to the starting approximation. Two variants of the approach are considered, with the first model utilizing a Picard Iteration operating on the general differential equations in rectangular coordinates, which are approximated by Chebyshev polynomials. The second variant makes use of the KS transformation to render the unperturbed motion rigorously linear. These techniques combined improve the time interval over which the Picard Iteration converges, and increases the speed of convergence over all time intervals. A numerical study demonstrates excellent execution time efficiency, and shows that these algorithms are also attractive for parallelization if needed for further computational speedup. These new algorithms address improvements in the solutions of a fundamental problem in astrodynamics and should find widespread use in contemporary and future applications.

INTRODUCTION AND MOTIVATION

Lambert's problem is *the* classical two-point boundary value problem in celestial mechanics, which was first developed and solved by Johann Heinrich Lambert in 1761. Solving the problem requires determining the orbital arc between a prescribed initial and final position in a specified flight time. In the modern literature, Battin [1] developed the most widely used and general algorithm for solving the unperturbed Lambert's Problem, for the case of Keplerian motion. For general perturbations, the most common approach is to utilize the state transition matrix sensitivity of the final state with respect to the initial velocity, and iterate via Newton's method on the three

^{*} Graduate Research Assistant, Department of Aerospace Engineering, Texas A&M University, TAMU-3141, College Station, Texas 77843-3141.

[†] Regents Professor, Distinguished Professor of Aerospace Engineering, Holder of the Royce E. Wisenbaker '39 Chair in Engineering, Department of Aerospace Engineering, Texas A&M University, TAMU-3141, College Station, Texas 77843-3141. Fellow AAS.

[‡] Assistant Professor, Department of Aerospace Engineering, Khalifa University, Abu Dhabi, UAE 127788.

components of initial velocity to “hit” the final desired position at the prescribed final time. The unperturbed solution (e.g., Battin’s algorithm) can be used to start the perturbed case iterations.

There are various challenges in space situational awareness with a difficult “data association” problem, wherein short tracks of many newly observed objects must be processed to determine orbits and correlate the tracked objects, if possible, with existing space object data bases. In the current state of the practice, hundreds of thousands of hypotheses must frequently be tested to find feasible preliminary orbits connecting time-displaced short tracks of unknown space objects, and these preliminary orbits and the underlying data associations are taken as the starting estimates for further correlation and final hypotheses and orbit determination. Solutions of Lambert’s problem are presently used extensively to address this challenge and the computational cost can exceed many CPU days per month on high performance computers. So the issue of accuracy and efficiency of the solution of two-point boundary value problems in orbital mechanics lie near the heart of a critically important computational grand challenge of vital national interest. These considerations provided the motivation for the current paper.

KS REGULARIZING TRANSFORMATION

The Kustaanheimo-Stiefel (KS) transformation [2] is a method for linearizing, without approximation, the two-body problem through a judicious coordinate transformation.

We begin by writing the classical differential equations of orbital motion in the most familiar rectangular coordinates:

$$\frac{d^2\mathbf{r}}{dt^2} = -\frac{\mu}{r^3} \mathbf{r} + \mathbf{F}, \quad (1)$$

where $\mathbf{r} = [x \ y \ z]^T$, $r = |\mathbf{r}|$. The KS transformation involves transforming both the position coordinates and the time variable. The position transformation can be written compactly in matrix form as

$$\begin{Bmatrix} x \\ y \\ z \\ 0 \end{Bmatrix} = L(\mathbf{u})\mathbf{u}, \text{ where } L(\mathbf{u}) = \begin{bmatrix} u_1 & -u_2 & -u_3 & -u_4 \\ u_2 & u_1 & -u_4 & -u_3 \\ u_3 & u_4 & u_1 & u_2 \\ u_4 & -u_3 & u_2 & -u_1 \end{bmatrix}, \text{ Inverse: } \mathbf{u} = \begin{Bmatrix} \left(\frac{r+x}{2}\right)^{1/2} \\ \frac{y}{2u_1} \\ \frac{z}{2u_1} \\ 0 \end{Bmatrix}. \quad (2)$$

The operator $L(\mathbf{u})$ has many interesting properties including

$$L^{-1}(\mathbf{u}) = \frac{1}{r} L^T(\mathbf{u}), \text{ and } L(\mathbf{u})\mathbf{v} = L(\mathbf{v})\mathbf{u}. \quad (3)$$

Also, the \mathbf{u} vector has the nice property that $r = \mathbf{u}^T \mathbf{u}$. Obviously, quadratic combinations of the elements of \mathbf{u} produce the rectangular coordinates and the radial distance. The mapping from \mathbf{u} to (x, y, z) is unique for all \mathbf{u} . The inverse from (x, y, z) to \mathbf{u} (the right most of Eqs (2)) is not unique, the given inverse transformation is the most popular of the finite set of inverse mappings (more on this point below); and any of these will establish valid initial conditions in \mathbf{u} and permit

valid motion computation from specified initial conditions. We mention that the finite set of feasible initial conditions in the generally complex (\mathbf{u}) space generate a finite set of geodesic curves on the surface of the variable radius four dimensional sphere, and *each of the ensuing \mathbf{u} -trajectories corresponds to exactly the same true physical motion in Cartesian space* $\{x(t), y(t), z(t)\}$. It is well known that any new time coordinate that is linearly proportional to r , together with Eqs (2) maps the nonlinear differential equations (1) into 4 oscillators in \mathbf{u} -space where the nonlinearities vanish identically as $\mathbf{F} \rightarrow 0$. For a particular choice of time coordinate (change in eccentric anomaly, implicitly, we restrict attention in the present discussion to the case of perturbed elliptic orbits for which the instantaneous Keplerian energy ($\alpha = 1/a$) is positive):

$$\frac{dt}{dE} = \left(\frac{1}{\sqrt{\mu\alpha}} \right) r, \quad \alpha = \frac{2}{r} - \frac{\dot{\mathbf{r}}^T \dot{\mathbf{r}}}{\mu} \quad (4)$$

Then the resulting rigorously linear differential equations can be shown to have the form

$$\frac{d^2 \mathbf{u}}{dE^2} + \frac{1}{4} \mathbf{u} = \frac{1}{2\mu\alpha} [\mathbf{u}^T \mathbf{u} I + 4 \frac{d\mathbf{u}}{dE} \frac{d\mathbf{u}^T}{dE}] L^T(\mathbf{u}) \mathbf{F}. \quad (5)$$

And the time is a function of the change in eccentric anomaly from Eq. (4)

$$t = t_0 + \int_0^E \left(\frac{r(\phi)}{\sqrt{\mu\alpha(\phi)}} \right) d\phi. \quad (6)$$

And finally, for $\mathbf{F} \neq 0$ case, α satisfies the variation of parameters differential equation

$$\frac{d\alpha}{dE} = -\frac{4}{\mu} \frac{d\mathbf{u}^T}{dE} L^T(\mathbf{u}) \mathbf{F} \quad \text{and} \quad \alpha = \frac{1}{2} (\mathbf{u}^T \mathbf{u} + 4 \frac{d\mathbf{u}^T}{dE} \frac{d\mathbf{u}}{dE}) \quad (7)$$

The second expression for α is the KS transformed Keplerian energy equation which holds instantaneously due to the osculation constraint in the variation of parameters. Therefore α in equation (5) does not have to be solved by a differential equation, it is a known function of the KS state variables.

For the $\mathbf{F} = 0$ case, of course the integral in (6) can be done analytically, and also it is evident that solving the four uncoupled harmonic oscillators of Eq. (5) is simply

$$\mathbf{u} = \mathbf{u}_0 \cos \frac{E}{2} + 2 \frac{d\mathbf{u}}{dE} \Big|_0 \sin \frac{E}{2}, \quad \frac{d\mathbf{u}}{dE} = -\frac{1}{2} \mathbf{u}_0 \sin \frac{E}{2} + \frac{d\mathbf{u}}{dE} \Big|_0 \cos \frac{E}{2} \quad (8)$$

Or, in state transition matrix form:

$$\begin{Bmatrix} \mathbf{u} \\ \frac{d\mathbf{u}}{dE} \end{Bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_0 \\ \frac{d\mathbf{u}}{dE} \Big|_0 \end{Bmatrix}, \quad \text{where the } 4 \times 4 \text{ submatrices are simply} \quad (9)$$

$$\Phi_{11} = \cos \frac{E}{2} I, \quad \Phi_{12} = 2 \sin \frac{E}{2} I, \quad \Phi_{21} = -\frac{1}{2} \sin \frac{E}{2} I, \quad \Phi_{22} = \cos \frac{E}{2} I$$

More generally, for an arbitrary force, the differential equations (5)-(7) must be solved numerically, however for small forces, they represent weak perturbations of the Keplerian motion and these equations are attractive from several points of view.

From the work of Bai and Junkins we know that the convergence of the Picard method is a function of the “strength” of the dominant terms of the differential equation. Therefore, we can anticipate the $\frac{1}{4}$ coefficient of Eq (5) suggests a basis for optimism that significant advantages will be achieved in these transformed equations, compared to Eqs (1), for reducing the number of Picard iterations and also increasing the maximum time interval over which the Picard contraction mapping iterations will converge. We anticipate these advantages for both the initial value problem and for the two point boundary value problem. As will be evident below, these heuristic expectations are consistent with numerical reality and represent a significant computational advantage for both initial and boundary value problems.

Before looking at the general three dimensional Lambert problem in detail, it is useful to consider the planar, Keplerian special case. The upper 2×2 of $L(\mathbf{u})$ is all that is needed of the position transformation and the resulting equations turn out to be the classical Levi-Civita transformation [3] discovered in 1920, some forty years prior to the more general KS result.

Restricting the motion to the plane ($z(t) = 0$), then the general KS transformation simplifies as follows:

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = L(\mathbf{u})\mathbf{u}, \text{ where } L(\mathbf{u}) = \begin{bmatrix} u_1 & -u_2 \\ u_2 & u_1 \end{bmatrix}, \text{ Inverse: } \mathbf{u} = \begin{Bmatrix} \pm \left(\frac{r+x}{2} \right)^{1/2} \\ \frac{y}{2u_1} \end{Bmatrix} \text{ or } \mathbf{u} = \begin{Bmatrix} \frac{y}{2u_2} \\ \pm \left(\frac{r-x}{2} \right)^{1/2} \end{Bmatrix}, \quad (10)$$

As mentioned above, the mapping from \mathbf{u} -space to Cartesian space is not unique. In fact for the planar case ($z(t) = 0$), each point in \mathbf{u} -space has eight possible corresponding positions in Cartesian space. Four of these are imaginary and can be immediately eliminated, but of the remaining four, if the *wrong* \mathbf{u} -position can easily be specified (i.e., on the opposite side of the \mathbf{u} -sphere for a fractional physical orbit transfer desired) is selected, the solution may be mathematically feasible, but be physically impossible for the prescribe t_f . Careful attention is required to avoid such circumstances.

Based on studying the analytical solution (notice two “revolutions” in \mathbf{u} -space correspond to one “revolution” in Cartesian coordinates), we find that after selecting the sign on the real boundary condition in \mathbf{u} space at initial time, there are only two real boundary conditions possible at any subsequent time, and these differ only in sign and along a \mathbf{u} trajectory sign switches occur when the change E in eccentric anomaly passes through odd multiples of π .

MODIFIED CHEBYSHEV PICARD ITERATION

Modified Chebyshev Picard Iteration (MCPI) is an attractive numerical method for solving linear or non-linear differential equations. It combines the discoveries of two great mathematicians: Emile Picard (Picard Iteration) and Rafnuty Chebyshev (Chebyshev Polynomials). The original fusion of orthogonal approximation and Picard iteration was apparently first proposed by Clenshaw and Norton in 1963 [4].

Picard observed that any first order differential equation

$$\dot{x}(t) = f(t, x(t)), \quad (11)$$

with an initial condition $x(t_0) = x_0$, and an integrable right hand side may be rearranged, without approximation, to obtain the following integral equation:

$$x(t) = x(t_0) + \int_{t_0}^t f(\tau, x(\tau)) d\tau. \quad (12)$$

This re-arrangement, at first glance, does not appear to have made any progress, since the unknown trajectory $x(t)$ is contained in the integrand on the right hand side. A sequence of approximate solutions $x^i(t)$, ($i = 1, 2, 3, \dots, \infty$), for the path $x(t)$ that satisfies this differential equation may be obtained through Picard iteration using the following formula:

$$x^i(t) = x(t_0) + \int_{t_0}^t f(\tau, x^{i-1}(\tau)) d\tau, \quad i = 1, 2, \dots \quad (13)$$

In first step toward the MCPI method, orthogonal Chebyshev polynomials are used as basis functions to approximate the integrand in the Picard integral. Chebyshev polynomials reside in the domain $\tau = [-1, 1]$, and can be defined recursively as:

$$T_0(\tau) = 1, \quad T_1(\tau) = \tau, \quad T_{k+1}(\tau) = 2\tau T_k(\tau) - T_{k-1}(\tau). \quad (14)$$

Unlike traditional step-by-step integrators, for example the Runge-Kutta methods, MCPI is unique in that long state trajectory arcs are approximated during the Picard iteration, and under known theoretical circumstances, we can show [4] that the Picard sequence is a contraction mapping guaranteed to converge to the solution of Eq. (11). The system dynamics are normalized such that the timespan of integration is projected onto the domain $\{-1 \leq \tau \leq +1\}$ of the Chebyshev polynomials, thus the system states can be approximated using the Chebyshev polynomial basis functions. The orthogonal nature of the basis function means that the coefficients that linearly scale the basis functions can be computed independently as simple ratios of inner products with no matrix inversion.

As a consequence of the independence and orthogonality of the basis functions, the coefficients multiplying the Chebyshev basis functions may be computed, as an inner product of the basis functions with the integrand, in parallel by separate processor threads with no matrix inversion required. Since the orthogonal polynomials, for large N constitute a complete set, and no matrix inversion is required, a smooth integrand can be approximated to machine precision. This independently computable integrand approximation coefficients is the first of two available layers of parallelization in the MCPI method. The second layer of parallelization is more important and is enabled by the fact that the entire state trajectory over the time interval of interest is estimated at once. Thus the calculation of the integrand functions (which must be computed as a function of the system states along the current approximate trajectory, at the nodes, as required for the discrete inner products leading to the approximation coefficients) can be performed at all nodes simultaneously in parallel processor threads. Using MCPI, over an order of magnitude speedup from traditional methods is achieved in serial processing, and an additional order of magnitude, or more, is achieved in parallel architectures, depending on the specifics of the parallel implementation.

A key feature of MCPI is a non-uniform cosine sampling domain of the Chebyshev basis functions called Chebyshev-Gauss-Lobatto (CGL) nodes, defined in the following equation.

$$\tau_j = \cos(j\pi / N), \quad j = 0, 1, 2, \dots, N \quad (31)$$

This sampling scheme has much higher nodal density towards the edges, which enables a higher accuracy solution near the boundaries of the state trajectory to compensate for the Runge phenomena (a common concern in function approximation whereby larger oscillatory approximation errors are returned near the edges of the domain due to lack of support for the approximation outside the boundaries of the approximation domain). The coefficients multiplying the Chebyshev basis functions are approximated by the method of least squares, which generally requires a matrix inversion. As a consequence of a consistent choice of choice of basis functions, weights, and discrete node locations is that the matrix required to be inverted in the Normal Equations of least squares is rigorously diagonal, thus the inverse is trivial and the coefficient computation is independent.

In 2010, Bai’s dissertation [5] extending the classical work of Clenshaw and Norton [4], and the more recent and related works of Feagin [14] Fukushima [15] and Shaver [16]. She established new convergence insights and optimized the solution of initial value problems utilizing vector-matrix formulations. She also proved the capability of the method to outperform the most commonly used methods representing the state of the practice for numerical integration of ODEs on several representative benchmark problems. Bai and Junkins applied MCPI to non-linear IVPs and orbit propagation in [6], and then showed that MCPI can outperform other higher order integrators such as Runge-Kutta-Nystrom 12(10). In [7] Bai and Junkins applied MCPI to efficiently solve Lambert’s transfer problem, and to solving an optimal control trajectory design problem more accurately and efficiently than the Chebyshev pseudospectral method. Notably, over intervals where the Picard iteration converges, there is no need to use a shooting method to solve Lambert problems and similar two point boundary value problems (TPBVPs). In [8] Bai and Junkins use MCPI in a complex three-body station-keeping control problem formulated as a sequence of TPBVPs. Subsequent publications by Junkins et al. [9], [10], and [11] further clarify the concept and derivation of MCPI and orthogonal approximation in general, and apply the method to problems in the field of astrodynamics. The most recent publication [12] discusses how the MCPI algorithm for the IVP has been made into an easily accessible library.

A full derivation of MCPI is beyond the scope of this short paper. Instead we present flow charts in Figure 1 that briefly summarize the mathematics represented in the more elegant vector/matrix formulation, which is computationally the most efficient way to implement the method. Any of the above references provide more detailed derivations, as well as examples and results that demonstrate the power of the MCPI algorithm with regard to speed and accuracy. Additionally, those references contain comparisons to other well-known integrators including high-order Runge-Kutta methods and the Gauss-Jackson method.

We digress briefly to discuss metrics for accuracy of the numerical solutions presented. We note that it is relatively straightforward to enforce *symplectic* constraints on the above developments, however, we are attracted to the theoretically stronger fact the classical Picard iteration of Eq. (28) is a contraction mapping converging to the solution of Eq (26); this rigorous theoretical guarantee can be lost in the process of trying to enforce a constraint that is already theoretically satisfied. In enforcing underdetermined symplectic constraints in any step-by-step numerical method to solve a differential equation, there is no guarantee that the true solution is obtained (because an infinity of neighboring trajectories have the same energy and removing small residuals can result small iso-energy errors by such ad hoc imposition of a scalar constraint). On the other hand, the use of exact integrals to check the integrity of the solution process is the preferred approach and the one taken in this paper. We mention, as a matter of course, we also examine the acceleration errors (how well the trajectory and its derivatives satisfy the differential equations) at mid-points between the inner product nodes as an additional numerical check to confirm the fidelity and accuracy of the solutions we discuss below.

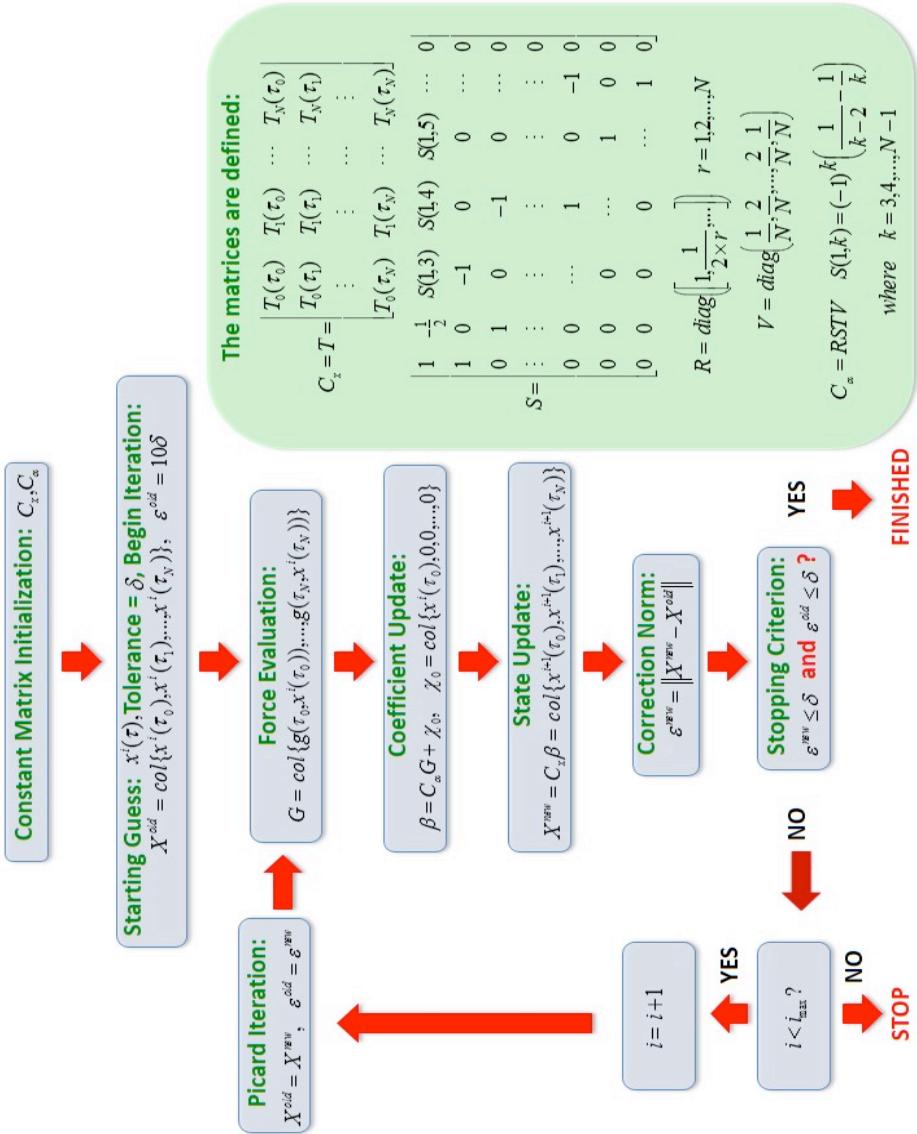


Figure 1.

A flow diagram outlining the MCPI Initial Value Problem algorithm in Vector-Matrix form.

The procedure is very similar for the Boundary Value Problem, with the minor differences being a few elements in the S matrix and the inclusion of the final boundary condition.

As a consequence, only the S matrix and manner in which constants are determined changes; Two-point boundary problems and initial value problems are solved by almost identical codes.

RESULTS AND DISCUSSION

In the KS transformed u -space the time variable has been transformed to a form of eccentric anomaly, and the final eccentric anomaly is now unknown. To determine this the Lambert TPBVP problem is solved analytically in the KS u -space (using Eqn (8)) for an iterative approximation of eccentric anomaly, and is transformed back to (x, y, z) . The Lambert/Kepler time – eccentric anomaly relationship is iterated by a Newton/Secant method to converge on the correct eccentric anomaly. A crude circular orbit guess was used in all cases. Typically 6 iterations are required to achieve an accuracy of 10^{10} . See Figures 2a – 2d.

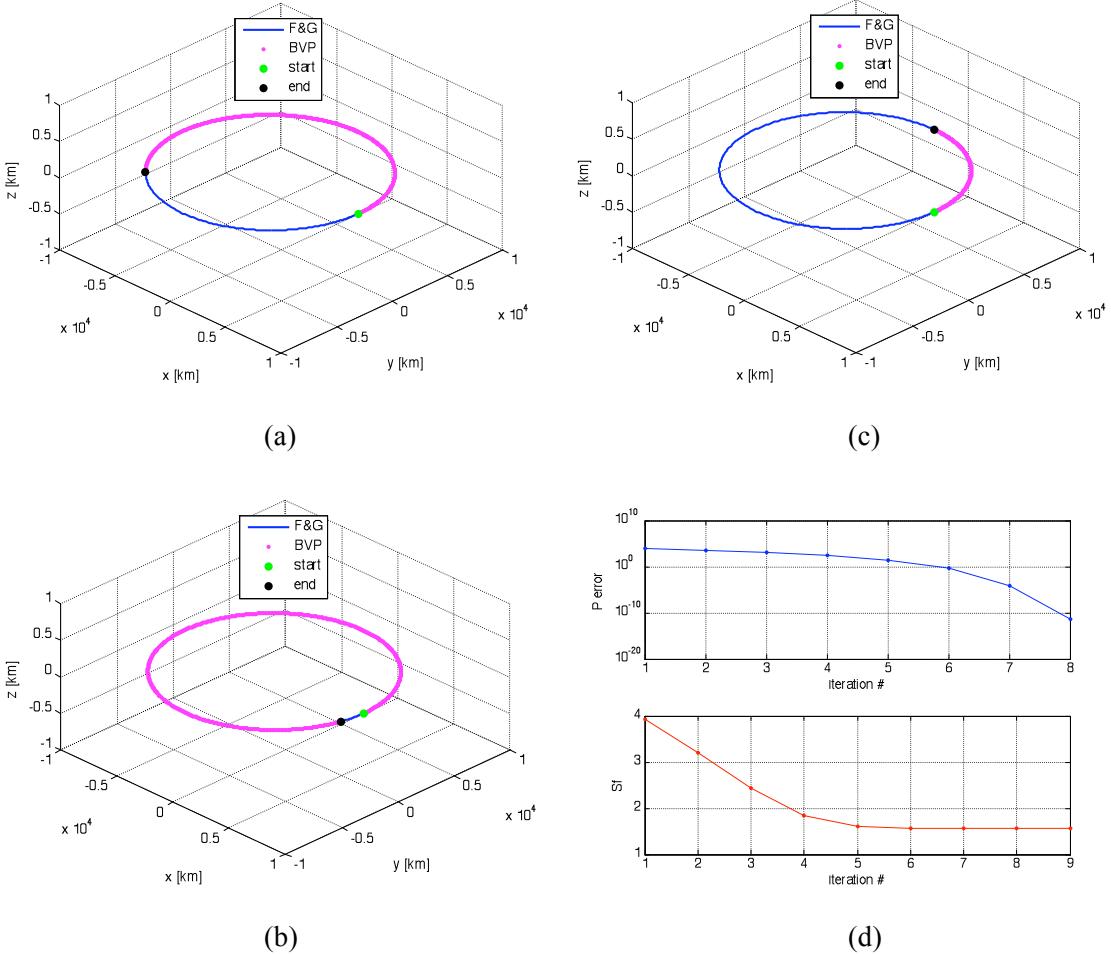


Figure 2: Lambert’s Problem is solved analytical using Eqn (8). Increasing orbital arc lengths are shown in (a) – (c), with a typical iteration convergence pattern shown in (d).

Solving Lambert’s Problem analytically provides the final eccentric anomaly (corresponding to the final time), and also a *warm start* solution approximation for solving the perturbed problem. The EGM2008 gravity model [13] is implemented for solving the perturbed problem. More on this later.

The two-body TPBVP is solved using MCPI with the known final eccentric anomaly that was calculated analytically. Previous MCPI results [5] for solution of the TPBVP have been limited to convergence over a maximum of 38% of an orbit. Implementing this KS transformation has enabled the interval of convergence to be vastly improved. The maximum convergence attainable is now 96% of an orbit!

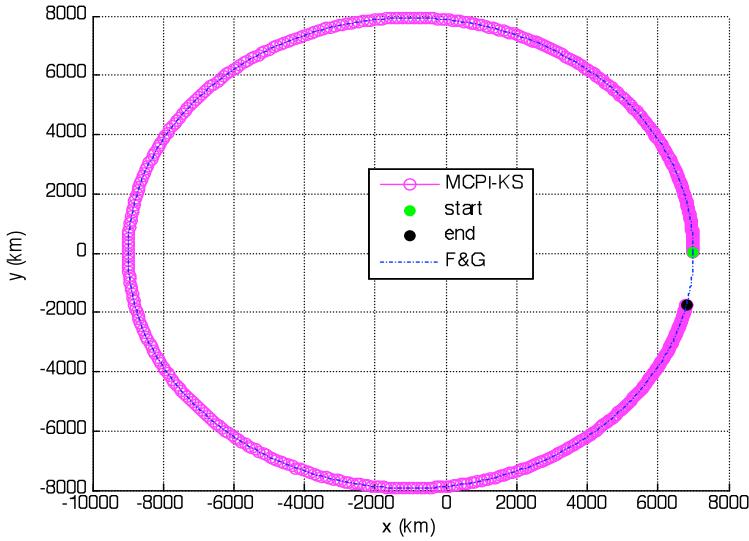


Figure 3: Lambert's Problem is solved for the two-body problem over an interval of 96% (467 iterations, 600 nodes) of an orbit. This is significantly further than previous MCPI results (38% of an orbit) [5].

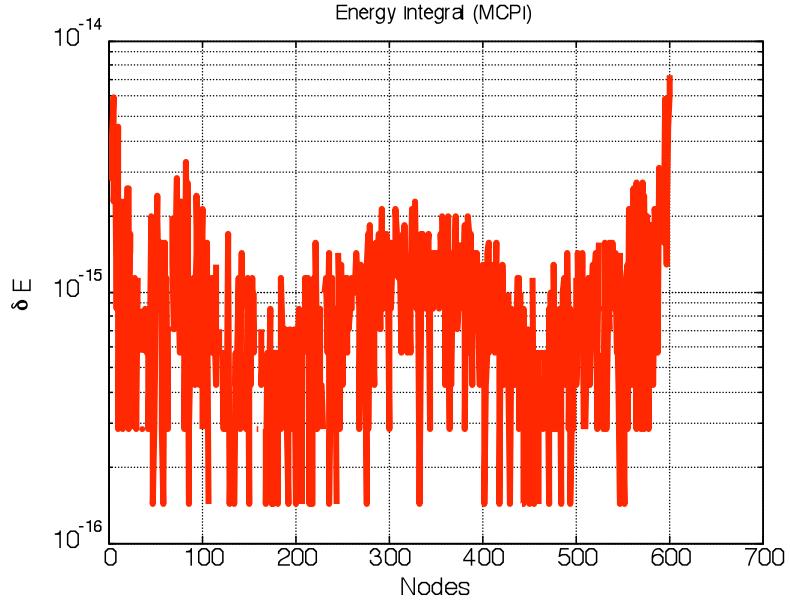


Figure 4: The Hamiltonian for the above orbit is constant.

The KS transformation can also be applied to the Initial Value Problem. Similarly, the final eccentric anomaly is determined analytically using the well-known F & G solution. As expected the domain of convergence achievable for the IVP is greatly increased compared with previous MCPI results [6]. Figures 5, 6 and 7 show the superiority of the KS transform with regard to number of iterations, constancy of the Hamiltonian (dE), and number of nodes required to achieve the desired propagation orbital arc length. Table 1 shows the four orbits that were used for testing our algorithm.

Table 1: Semi-major axis and eccentricity for the orbits that were used for testing our algorithm. In this paper all orbits start at perigee.

| Orbit Type | Semimajor axis (a) | Eccentricity (e) |
|------------|------------------------|----------------------|
| LEO | 8000 km | 0.125 |
| MEO | 10963 km | 0.4 |
| GEO | 26352 km | 0.6 |
| HEO | 32890 km | 0.8 |

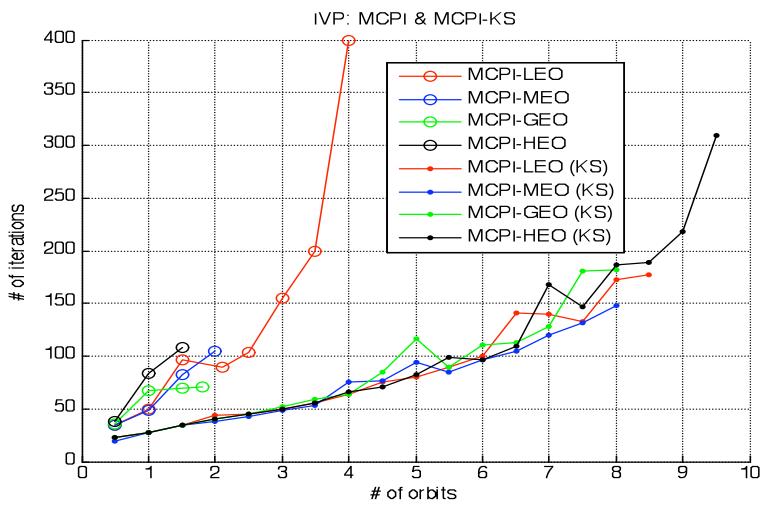


Figure 5: Comparative performance of MCPI and MCPI-KS for four different orbits (LEO, MEO, GEO, HEO). The figure shows that the KS version achieves a convergence domain of about 8.5 orbits with much fewer iterations required than the standard method.

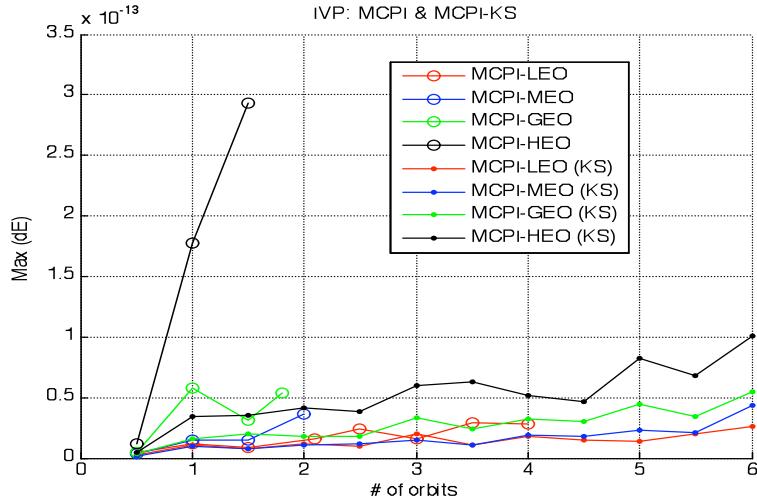


Figure 6: Each point represents the maximum relative change in energy over the specified orbital arc length. An accuracy of 10^{-14} is generally maintained.

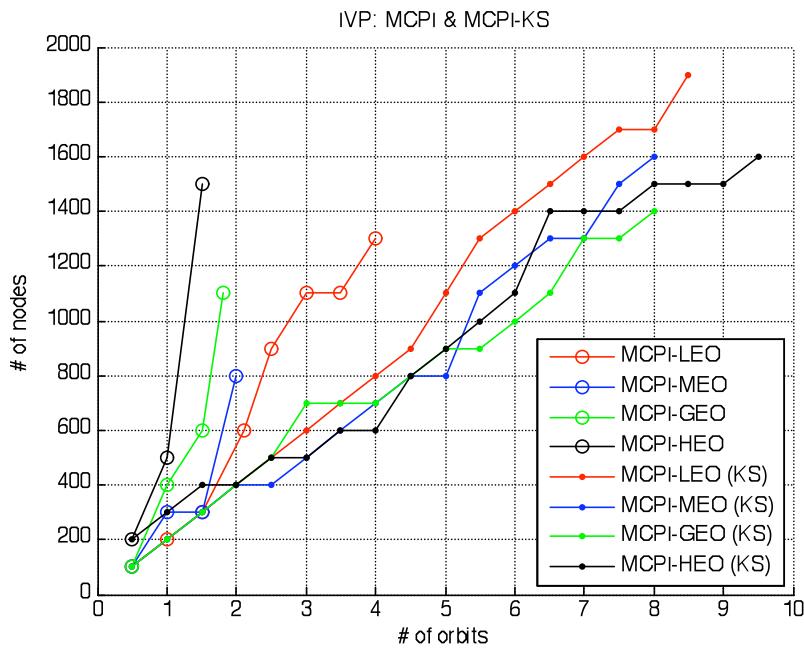


Figure 7: Comparative performance of MCPI and MCPI-KS for four different orbits (LEO, MEO, GEO, HEO). The figure shows that the KS version requires less nodes for propagating the same orbital arc length.

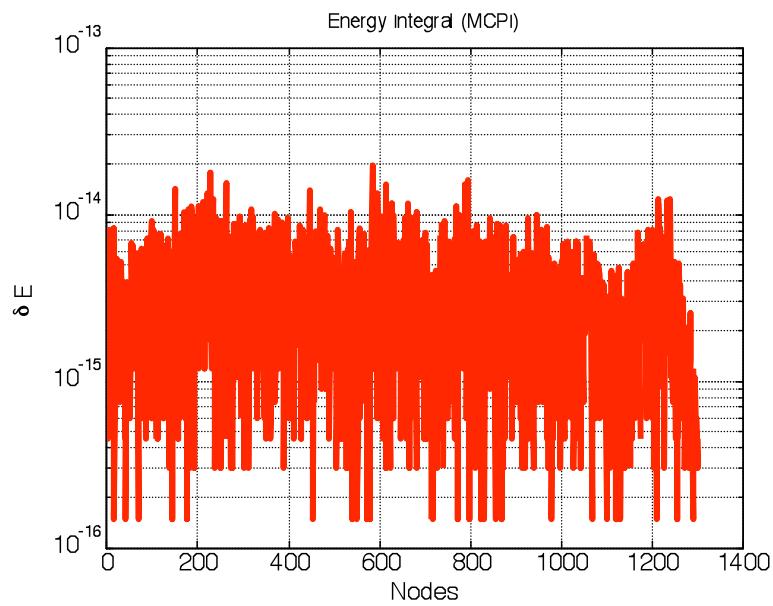


Figure 8: An example of an IVP solved over 5.5 LEO orbits reveals a constant Hamiltonian.

The EGM2008 gravity model [13] is implemented for solving perturbed orbits. As expected, the inclusion of perturbations decreases the interval of convergence. Note, as the degree increases the domain of convergence decreases (Figure 9). A constant Hamiltonian is maintained (Figure 11).

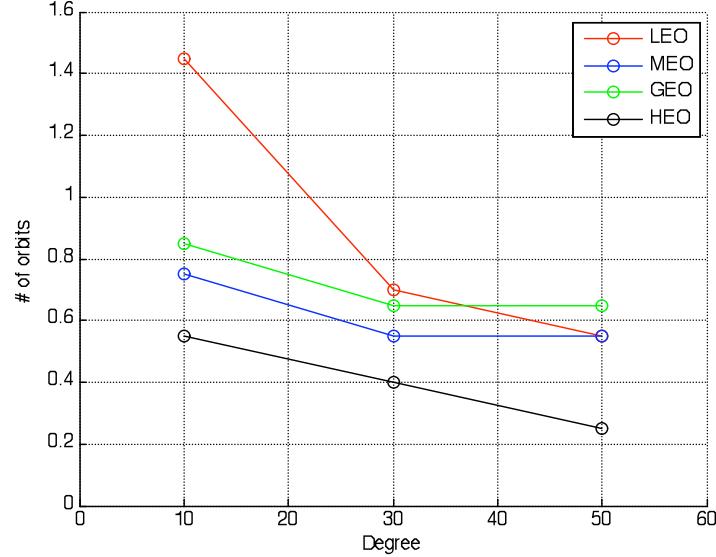


Figure 9: Domain of convergence for the IVP decreases as the degree of gravity spherical harmonic increases.

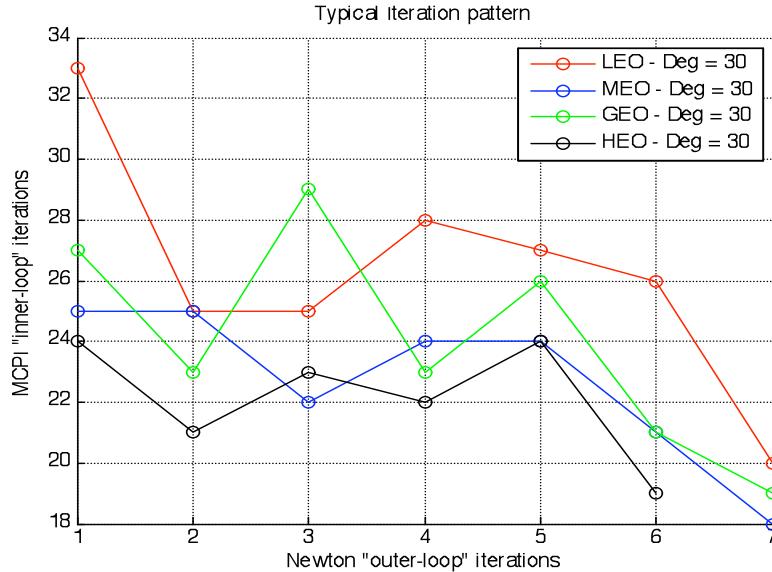


Figure 10: Typical iteration pattern of “inner” (MCPI) and “outer” (Newton) loop convergence.

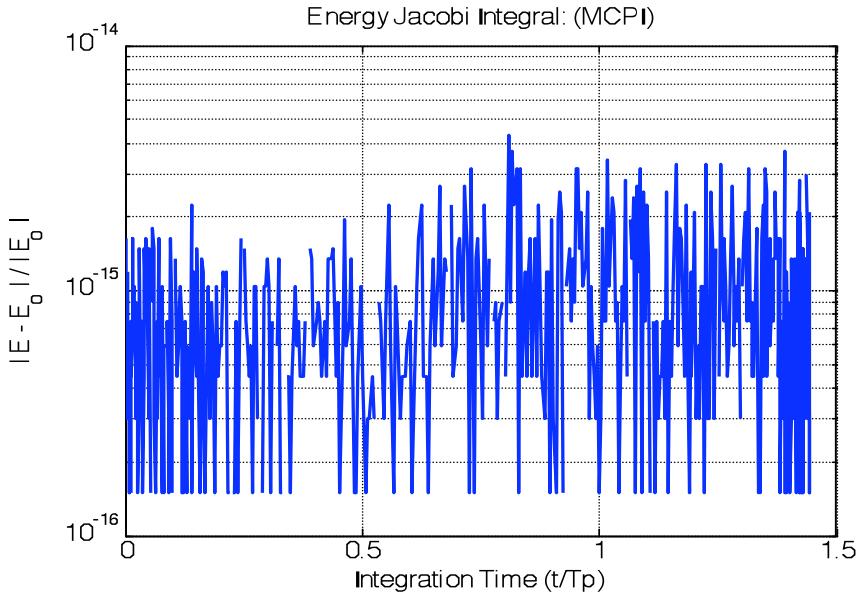


Figure 10: The Hamiltonian is constant for almost 1.5 LEO orbits when the EGM2008 perturbations are included (Degree = 10).

CONCLUSION

A method is presented for solving boundary and initial value problems in celestial mechanics. In particular we consider the well-known Lambert TPBVP. The approach is quite general, however certain details in the transformation space pose challenges. We have been able to resolve these difficulties fully for the planar classical two-body case, and we are engaged in a study to extend our numerical algorithm to the generally perturbed case. This method fuses three sets of ideas: (i) Picard Iteration, (ii) Orthogonal approximation, and regularizing transformation of the equations of motion. Two variants of the approach are considered, with the first model utilizing a Picard Iteration operating on the general differential equations in rectangular coordinates, which are approximated by Chebyshev polynomials. The second variant makes use of the KS transformation.

A numerical study demonstrates the superiority of the KS transformation with regard to accuracy and efficiency, and it shows a vastly increased domain of convergence for Lambert's Problem. In this paper, we have focused mainly on establishing greatly expanded time intervals over which initial value problems and boundary value problems can be solved for the KS-transformed orbit mechanics problem. Computational efficiency optimization will be addressed in future studies. These new algorithms and exciting new results address improvements in the solutions of a fundamental problem in astrodynamics and should find widespread use in contemporary and future applications.

ACKNOWLEDGEMENTS

We thank our sponsors: AFOSR, AFRL, LANL, and Applied Defense Systems for their support and collaborations under various contracts and grants.

REFERENCES

- [1] Battin, R., *An Introduction to the Mathematics and Methods of Astrodynamics*. Reston, VA: American Institute of Aeronautics and Astronautics, Inc, revised ed., 1999.
- [2] Kustaanheimo, P. and Stiefel, E. L., *Perturbation theory of Kepler motion based on spinor regularization*, Journal fur die Reine und Angewandt Mathematik 218, 204-219, 1965.
- [3] Levi-Civita T., *Sur la regularization du problem des trois corps*, Acta Mathematica 42, 99-144, 1920.
- [4] Clenshaw, C. W. and Norton, H. J., *The Solution of Nonlinear Ordinary Differential Equations in Chebyshev Series*, The Computer Journal, 6(1):88-92, 1963.
- [5] Bai, X., *Modified Chebyshev-Picard Iteration Methods for Solution of Initial Value and Boundary Value Problems*, PhD. Dissertation, Texas A&M University, College Station, Tex, USA, 2012.
- [6] 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.
- [7] Bai, X., and Junkins, J. L., *Modified Chebyshev-Picard Iteration Methods for Solution of Boundary Value Problems*, Advances in the Astronautical Sciences, vol. 140, pp. 381–400, 2011.
- [8] 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.
- [9] Junkins, J., Bani Younes, A., Woollards, R.. and Bai, X., *Orthogonal Approximation in Higher Dimensions: Applications in Astrodynamics*, ASS 12-634, JN Juang Astrodynamics Symp, College Station, TX, June 14-26, 2012.
- [10] Junkins, J. L., Bani Younes, A., Woollards, R M., and Bai, X., *Orthogonal Approximation in Higher Dimensions: Applications in Astrodynamics*, submitted to The Journal of the Astronautical Sciences, June, 2013.
- [11] Junkins, J. L., Bani Younes, A., Woollards, R. M., and Bai, X., *Picard Iteration, Chebyshev Polynomial and Chebyshev Picard Methods: Application in Astrodynamics*, accepted in The J of the Astronautical Sciences, July, 2013.
- [12] Macomber, B., Woollards, R. M., Probe, A., Bani Younes, A., Junkins, J. L., *Modified Chebyshev Picard Iteration for Efficient Numerical Integration of ordinary Differential Equations*, Advanced Maui Optical and Space Surveillance Technologies Conference, Maui, Hawaii, June 2013.
- [13] Bani Younes, A., *Orthogonal Polynomial Approximation in Higher Dimensions: Applications in Astrodynamics.*, Ph.D. dissertation, Texas A&M Univ, College Station, TX, 2013.
- [14] Feagin, T. and Nacozy, P.. “Matrix Formulation of the Picard Method for Parallel Computation,” *Celestial Mechanics and Dynamical Astronomy*, Vol. 29, Feb. 1983, pp 107–115.
- [15] Fukushima, T. “Vector Integration of Dynamical Motions by the Picard-Chebyshev Method,” *The Astronomical Journal*, Vol. 113, Jun. 1997, pp. 2325–2328.
- [16] Shaver, J. S. *Formulation and Evaluation of Parallel Algorithms for the Orbit Determination Problem*, Ph.D. Dissertation, Department of Aeronautics and Astronautics, MIT, Cambridge, MA, Mar. 1980.

Submitted to the AIAA Journal for Guidance, Control and Dynamics
for consideration to be included in the special issue dedicated to Richard H. Battin

NEW SOLUTIONS FOR LAMBERT'S PROBLEM UTILIZING REGULARIZATION AND PICARD ITERATION

Robyn M. Woollards,^{*} Ahmad Bani Younes,[†] and John L. Junkins[‡]

This paper presents three sets of related new developments with regard to solving the fundamental two-point-boundary-value problem of astrodynamics: (1) A solution of the Keplerian Lambert Problem based on the KS regularized transformation of the equations of motion. (2) A shooting technique to solve the KS transformed differential equations for the perturbed Lambert Problem starting with the Keplerian solution. (3) A Picard iteration approach to solving the perturbed Lambert Problem which does not require a shooting technique. The last two methods are validated using moderate degree and order (40,40) spherical harmonic expansion to represent near-earth gravity field. The solution of the Keplerian Lambert problem in KS variables is elegantly simple and solves both the fractional order and multi-revolution cases. The second development is general in that it is applicable to general multi-revolution case and general perturbations. The third contribution is very attractive, but is only applicable for orbit transfers with a time interval within the domain of convergence of Picard iteration. Significantly, we show that the time interval for Picard iteration convergence is increased from about 0.38 of an orbital period for LEO orbits represented in traditional Cartesian coordinates to about 0.98 of an orbital period for the same case represented in the KS variables.

INTRODUCTION AND MOTIVATION

Lambert's problem is *the* classical two-point boundary value problem in celestial mechanics, which was first developed and solved by Johann Heinrich Lambert in 1761. Solving the problem requires determining the orbital arc between a prescribed initial and final position corresponding to a specified flight time. In the modern literature, Battin [1] developed the most widely used and general algorithm for solving the unperturbed Lambert's Problem, for the case of Keplerian motion. For general perturbations, the most common approach is to utilize the state transition matrix sensitivity of the final state with respect to the initial velocity, and iterate via Newton's method on the three components of initial velocity to "hit" the final desired position at the prescribed final time. The unperturbed Lambert solution (e.g., Battin's Lambert algorithm, the p-iteration method,

^{*} Graduate Research Assistant, Department of Aerospace Engineering, Texas A&M University, TAMU-3141, College Station, Texas 77843-3141.

^{††} Assistant Professor, Department of Aerospace Engineering, Khalifa University, Abu Dhabi, UAE 127788.

[‡] Regents Professor, Distinguished Professor of Aerospace Engineering, Holder of the Royce E. Wisenbaker '39 Chair in Engineering, Department of Aerospace Engineering, Texas A&M University, TAMU-3141, College Station, Texas 77843-3141. Honorary Fellow of AIAA.

or the alternative KS Keplerian Lambert solution presented herein) can be used to start the perturbed case iterations.

One motivation for our paper are the various challenges in space situational awareness with a difficult “data association” problem, wherein short tracks of many newly observed objects must be processed to determine orbits and correlate the tracked objects, if possible, with existing space object data bases. In the current state of the practice, hundreds of thousands of hypotheses must frequently be tested to find feasible preliminary orbits connecting time-displaced short tracks of unknown space objects, and these preliminary orbits and the underlying data associations are taken as the starting estimates for further correlation and final hypotheses and orbit determination. “Short” tracks may be up to several orbits, so the effects of perturbations, if ignored, will typically introduce residual errors much larger than the measurement errors themselves. In the current state of practice, data association hypotheses are tested for preliminary orbit estimation using the Keplerian Lambert solutions for sufficiently short arcs, but higher precision is needed to accommodate multi-orbit hypothesis testing. When hundreds of thousands of hypotheses are tested daily and perturbations are included, the computational cost can exceed many CPU days per month. So the issue of finding an optimal solutions [2] to a two-point boundary value problems lie near the heart of critically important computational challenges of vital interest in SSA. The inclusion of perturbations in Lambert’s problem and the development of efficient and robust methods is therefore of strong interest.

KS REGULARIZING TRANSFORMATION

The Kustaanheimo-Stiefel (KS) transformation [3] is a method for rigorously linearizing, without local approximation, the two-body problem through a judicious coordinate transformation.

We begin by writing the classical differential equations of orbital motion in the most familiar rectangular coordinates:

$$\frac{d^2\mathbf{r}}{dt^2} = -\frac{\mu}{r^3}\mathbf{r} + \mathbf{F}, \quad (1)$$

where $\mathbf{r} = [x \ y \ z]^T$, $r = |\mathbf{r}|$. The KS transformation involves transforming both the position coordinates and the time variable. The position transformation can be written compactly in matrix form as

$$\begin{Bmatrix} x \\ y \\ z \\ 0 \end{Bmatrix} = L(\mathbf{u})\mathbf{u}, \text{ where } L(\mathbf{u}) = \begin{bmatrix} u_1 & -u_2 & -u_3 & -u_4 \\ u_2 & u_1 & -u_4 & -u_3 \\ u_3 & u_4 & u_1 & u_2 \\ u_4 & -u_3 & u_2 & -u_1 \end{bmatrix}, \text{ Inverse: } \mathbf{u} = \begin{Bmatrix} ((r+x)/2)^{1/2} \\ \frac{y}{(2(r+x))^{1/2}} \\ \frac{z}{(2(r+x))^{1/2}} \\ 0 \end{Bmatrix}. \quad (2)$$

The operator $L(\mathbf{u})$ has many interesting properties [3, 4, 5] including

$$L^{-1}(\mathbf{u}) = \frac{1}{r} L^T(\mathbf{u}), \text{ and } L(\mathbf{u})\mathbf{v} = L(\mathbf{v})\mathbf{u}. \quad (3)$$

Also, the \mathbf{u} vector has the nice property: $r = \mathbf{u}^T \mathbf{u}$. Obviously, quadratic combinations of the elements of \mathbf{u} produce the rectangular coordinates and the radial distance. The mapping from \mathbf{u} to (x, y, z) is unique, for all \mathbf{u} . However, the inverse from (x, y, z) to \mathbf{u} (the right most of Eqs (2)) is *not unique*, the given inverse transformation is the most popular of the finite set of inverse mappings (more on this point below). Any member of a feasible set of inverse points can be used to compute the initial state in \mathbf{u} -space to establish valid initial condition for a \mathbf{u} trajectory and permit valid trajectory computation. We mention that the finite set of feasible initial conditions in the generally complex (\mathbf{u}) space generates a corresponding finite set of geodesic curves on the surface of the 4 dimensional sphere whose time varying radius is $|\mathbf{u}(E(t))| = \sqrt{r(t)}$, and *each of the ensuing \mathbf{u} -trajectories corresponds to exactly the same true physical motion in Cartesian space $\{x(t), y(t), z(t)\}$.*

It is well known that any new time coordinate that is linearly proportional to r , together with Eqs (2) maps the nonlinear differential equations (1) into 4 oscillators in \mathbf{u} -space where the nonlinearities vanish identically as $\mathbf{F} \rightarrow 0$. We restrict attention in the present discussion to the case of perturbed elliptic orbits for which the instantaneous Keplerian energy ($\alpha = 1/a$) is positive, where $a = a(t)$ and we adopt the following implicit time transformation $E \rightarrow t$:

$$\frac{dt}{dE} = \left(\frac{1}{\sqrt{\mu\alpha}} \right) r, \quad \alpha = \frac{2}{r} - \frac{\dot{\mathbf{r}}^T \dot{\mathbf{r}}}{\mu} \quad (4)$$

Then the resulting rigorously linear differential equations can be shown to have the form

$$\frac{d^2 \mathbf{u}}{dE^2} + \frac{1}{4} \mathbf{u} = \frac{r}{2\mu\alpha} [I + \frac{4}{r} \frac{d\mathbf{u}}{dE} \frac{d\mathbf{u}^T}{dE}] L^T(\mathbf{u}) \mathbf{F}, \quad (5)$$

and time is related to the change in eccentric anomaly from Eq. (4) through the integral:

$$t = t_0 + \int_0^E \left(\frac{r(\phi)}{\sqrt{\mu\alpha(\phi)}} \right) d\phi. \quad (6)$$

Finally, for $\mathbf{F} \neq 0$ case, it can be shown that [5] α satisfies the variation-of-parameters differential equation

$$\frac{d\alpha}{dE} = -\frac{4}{\mu} \frac{d\mathbf{u}^T}{dE} L^T(\mathbf{u}) \mathbf{F} \quad \text{and also, due to osculation: } \alpha = \frac{2}{r} \left[1 + \frac{4}{r} \frac{d\mathbf{u}^T}{dE} \frac{d\mathbf{u}}{dE} \right]^{-1}. \quad (7)$$

The second expression for α is the KS transformed Keplerian energy equation which holds in the presence of perturbations for $\alpha(t)$ due as an osculation constraint in the variation of parameters. Therefore α in Eq. (5) does not have to be solved by a differential equation as is frequently done, rather it is a known function of the instantaneous KS state variables, given in Eq. (7).

Substitution of the equation from Eq. (7) into Eq. (5) gives the new and elegant form for the generally perturbed differential equation of motion in the KS variables:

$$\frac{d^2 \mathbf{u}}{dE^2} + \frac{1}{4} \mathbf{u} = \frac{r^2}{4\mu} \left[1 + \frac{4}{r} \frac{d\mathbf{u}^T}{dE} \frac{d\mathbf{u}}{dE} \right] [I + \frac{4}{r} \frac{d\mathbf{u}}{dE} \frac{d\mathbf{u}^T}{dE}] L^T(\mathbf{u}) \mathbf{F} \quad (8)$$

Notice, since the spherical harmonic series first term $1/r^2$ and all higher n^{th} order terms contain $1/r^n$, the multiplication by r^2 on the RHS of Eq. (8), simply reduces by 2 the power of r in all the gravitational perturbations.

THE KEPLERIAN LAMBERT PROBLEM IN KS VARIABLES

For the $\mathbf{F} = 0$ case, of course the integral in (6) can be done analytically, and also it is evident that solving the four uncoupled harmonic oscillators of Eq. (5) or (8) is simply

$$\mathbf{u} = \mathbf{u}_0 \cos \frac{E}{2} + 2 \frac{d\mathbf{u}}{dE} \Big|_0 \sin \frac{E}{2}, \quad \frac{d\mathbf{u}}{dE} = -\frac{1}{2} \mathbf{u}_0 \sin \frac{E}{2} + \frac{d\mathbf{u}}{dE} \Big|_0 \cos \frac{E}{2} \quad (9)$$

Or, in state transition matrix form:

$$\begin{Bmatrix} \mathbf{u} \\ \frac{d\mathbf{u}}{dE} \end{Bmatrix} = \begin{bmatrix} \Phi_{11} & \Phi_{12} \\ \Phi_{21} & \Phi_{22} \end{bmatrix} \begin{Bmatrix} \mathbf{u}_0 \\ \frac{d\mathbf{u}}{dE} \Big|_0 \end{Bmatrix}, \text{ where the } 4 \times 4 \text{ submatrices are simply} \quad (10)$$

$$\Phi_{11} = \cos \frac{E}{2} I, \quad \Phi_{12} = 2 \sin \frac{E}{2} I, \quad \Phi_{21} = -\frac{1}{2} \sin \frac{E}{2} I, \quad \Phi_{22} = \cos \frac{E}{2} I$$

The integral of Eq. (6) can be carried out analytically for the $\mathbf{F} = 0$ case to obtain [4]

$$\alpha^{3/2} \sqrt{\mu} (t - t_0) + N(2\pi) = E - (1 - \alpha r_0) \sin E - \alpha^{1/2} \sigma_0 (1 - \cos E), \quad \sqrt{\mu} \sigma_0 \equiv \mathbf{r}_0^T \dot{\mathbf{r}}_0. \quad (11)$$

In all cases above E denotes the change in eccentric anomaly from initial conditions to the current state, i.e., E is not referenced to perigee. N is the specified integer number of completed orbits before reaching the desired final position. If N is set to an infeasible value, no roots exist. A maximum of $2N+1$ roots exist in general (for $N=0$, there is a single unique orbit for the fractional orbit transfer case, except for the co-linear position case in which the orbit plane is not unique). The singularity structure for the Keplerian special case has been found to carry over to the gravitational perturbed generalization of this two-point boundary value problem.

More generally, for an arbitrary force, the differential equations (5)-(8) must be solved numerically, however for small forces, they represent a weakly-coupled, weakly-nonlinear oscillator description of orbital motion and these equations are attractive from several points of view.

From the work of Bai and Junkins [8-10] and the classical Picard literature, we know that the convergence of the Picard method is a function of the “strength” of the dominant terms of the differential equation. Therefore, we can anticipate the $1/4$ coefficient of Eqs. (5), (8) suggests a basis for optimism that significant advantages will be achieved in these transformed differential equations, compared to Eqs. (1), for reducing the number of Picard iterations and also increasing the maximum interval over which the Picard contraction mapping iterations will converge. We anticipate these advantages for both the initial value problem and for the two point boundary value

problem. As will be evident below, these heuristic expectations are consistent with numerical reality and represent a significant computational advantage for both initial and boundary value problems.

Before looking at the general three dimensional Lambert problem in detail, it is useful to consider the planar, Keplerian special case. The upper left 2x2 sub-matrix of $L(\mathbf{u})$ is the needed subset of the position transformation and the resulting equations turn out to be the classical Levi-Civita transformation [6] discovered in 1920, some forty years prior to the more general KS result.

Restricting the motion to the plane ($z(t) = 0$), then the general KS transformation simplifies as follows:

$$\begin{Bmatrix} x \\ y \end{Bmatrix} = L(\mathbf{u})\mathbf{u}, \quad L(\mathbf{u}) = \begin{bmatrix} u_1 & -u_2 \\ u_2 & u_1 \end{bmatrix}. \quad (12)$$

The inverse mapping is

$$\text{for } x \geq 0: \quad \mathbf{u} = \pm \begin{Bmatrix} ((r+x)/2)^{1/2} \\ \frac{y}{(2(r+x))^{1/2}} \end{Bmatrix} \quad \text{or for } x < 0: \quad \mathbf{u} = \pm \begin{Bmatrix} \frac{y}{(2(r-x))^{1/2}} \\ ((r-x)/2)^{1/2} \end{Bmatrix} \quad (13)$$

As evident above in Eqs. (2) and the planar special case of Eq. (11), the mapping from \mathbf{u} -space to Cartesian space is unique. In fact for the planar case ($z(t) = 0$), only four real points in \mathbf{u} -space exist, given in Eq. (12). For initial value problems, so long as we avoid the potential division by zero at $x = \pm r$ (by following the sign of x rules evident in Eq. (12)), the solution of these equations is very well behaved. For initial value problems, we only need to use these equations once at initial time, and the inverse mapping in Eq. (11) (or more generally Eq. (2)) is not branched. For two-point boundary value problems, however, we have to resolve the sign ambiguities carefully, otherwise we may accidentally “tell” the algorithm to look for a one-and-a-fraction orbit transfer instead of a fractional orbit transfer. Note that two revolutions occur in Cartesian space for each revolution in \mathbf{u} -space, quite analogous to quaternion representation of rotational motion. We find, that after selecting the sign on the real boundary condition in \mathbf{u} -space at initial time, while there are only two real boundary conditions possible at any subsequent time, we visit both of these in the Keplerian problem, separated by an orbital period (these differ only in sign and along a particular Keplerian \mathbf{u} trajectory, these sign switches occur when the change E in eccentric anomaly passes through multiples of 2π).

From Eq. (9), we can eliminate initial velocity in \mathbf{u} space as a function of the final boundary conditions

$$\frac{d\mathbf{u}}{dE} = \frac{1}{2 \sin \frac{E_f}{2}} \left(\mathbf{u}_f - \mathbf{u}_0 \cos \frac{E_f}{2} \right). \quad (14)$$

We now outline the completion of the solution of the Keplerian Lambert’s problem in KS variables. Using the energy equation $\alpha = \frac{2}{r_0} - \frac{dr^T}{dt} \Big|_{t_0} \frac{dr}{dt} \Big|_{t_0} = \frac{2}{r_0} \left[1 + \frac{4}{r_0} \frac{d\mathbf{u}^T}{dE} \Big|_0 \frac{d\mathbf{u}}{dE} \Big|_0 \right]^{-1}$ and also Eq.

(14), we can eliminate α and σ_0 in Eq. (11) as a function of $(\mathbf{u}_0, \mathbf{u}_f, E_f)$, leaving only E_f as an unknown. Then the modified Eq. (11) with all unknowns on the RHS eliminated except E_f can be iterated (for a given t_f) to determine E_f . This method is well-behaved and convergence is reliable, including all fractional and multi-revolution cases, somewhat analogous to Battin's classical Lambert solution, but in new variables.

MODIFIED CHEBYSHEV PICARD ITERATION

Modified Chebyshev Picard Iteration (MCPI) is an attractive numerical method for solving linear or non-linear differential and integral equations. MCPI combines the discoveries of two great mathematicians: Emile Picard (Picard Iteration) and Rafnuty Chebyshev (Chebyshev Polynomials), and recent developments in the associated linear algebra by Bai, Junkins, Feagin, et al. The original fusion of orthogonal approximation theory and Picard iteration was apparently introduced by Clenshaw and Norton in 1963 [7].

Picard observed that any first order differential equation

$$\dot{x}(t) = f(t, x(t)), \quad (15)$$

with an initial condition $x(t_0) = x_0$, and any integrable right hand side may be rearranged, without approximation, to obtain the following integral equation:

$$x(t) = x(t_0) + \int_{t_0}^t f(\tau, x(\tau)) d\tau. \quad (16)$$

This re-arrangement, at first glance, does not appear to have made any progress, since the unknown trajectory $x(t)$ is contained in the integrand on the right hand side. A sequence of approximate solutions $x^i(t)$, ($i = 1, 2, 3, \dots, \infty$), of the true solution $x(t)$ that satisfies this differential equation may be obtained through Picard iteration using the following Picard sequence of approximate paths $\{x^0(t), x^1(t), \dots, x^{i-1}(t), x^i(t), \dots\}$:

$$x^i(t) = x(t_0) + \int_{t_0}^t f(\tau, x^{i-1}(\tau)) d\tau, \quad i = 1, 2, \dots. \quad (17)$$

In first step toward the MCPI method, orthogonal Chebyshev polynomials are used as basis functions to approximate the integrand in Eq. (17) along the previous approximate trajectory $x^{i-1}(t)$. Chebyshev polynomials are defined over the domain $\{-1 \leq \tau \leq 1\}$, and can be generated from the two term recursion as:

$$T_0(\tau) = 1, \quad T_1(\tau) = \tau, \quad T_{k+1}(\tau) = 2\tau T_k(\tau) - T_{k-1}(\tau). \quad (18)$$

Unlike traditional step-by-step integrators, for example the Runge-Kutta methods, MCPI is a *path iteration method* in which long state trajectory arcs are approximated and updated at all time instances on each iteration. Under usually satisfied and known theoretical circumstances, we can show [7] that the Picard sequence is a contraction mapping guaranteed to converge to the solution of Eq. (14). The system dynamics are normalized such that the timespan of integration is projected onto the domain $\{-1 \leq \tau \leq +1\}$ of the Chebyshev polynomials, thus the system states can be

approximated using the Chebyshev polynomial basis functions. The orthogonal nature of the basis function means that the coefficients that linearly scale the basis functions can be computed independently as simple ratios of inner products with no matrix inversion. Since The Chebyshev polynomials are a complete set and due no orthogonality, no loss of precision matrix inversions are required, we can achieve machine precision (if desired) approximation of the integrand in Eq. (17) on each iteration and the resulting converged trajectory can approach machine precision solution of the differential equation over large time spans. In Bai's dissertation [8], she found that the actual time interval over which convergence is obtained for problems in celestial mechanics is over 3 orbits (for the case of an initial value problem and the usual Cartesian coordinate formulation of the equations of motion). For the two-point boundary value problem in Cartesian coordinates, however, she found that the time interval for which Picard iteration converges is reduced to about 0.38 of an orbit period.

As a consequence of the independence and orthogonality of the basis functions, the coefficients multiplying the Chebyshev basis functions may be computed, as an inner product of the basis functions with the integrand, in parallel by separate independent threads with no matrix inversion required. This independently computable integrand approximation coefficients is the first of two available layers of parallelization in the MCPI method. The second layer of parallelization is much more important and is enabled by the fact that acceleration over the entire state trajectory $x^{i-1}(t)$ permits us to compute independently and simultaneously. Thus the calculation of the integrand functions (which must be computed as a function of the system states along the current approximate trajectory, at the nodes, as required for the discrete inner products leading to the approximation coefficients) can be performed at all nodes simultaneously in parallel processor threads. Using MCPI, over an order of magnitude speedup from traditional methods is achieved in serial processing, and an additional one-to-two orders of magnitude, are achieved in parallel architectures, depending on the specifics of the parallel implementation.

A key feature of MCPI is a non-uniform cosine sampling of the $\{-1 \leq \tau \leq 1\}$ domain of the called Chebyshev-Gauss-Lobatto (CGL) nodes:

$$\tau_j = \cos(j\pi/N), j = 0, 1, 2, \dots, N \quad (17)$$

This set of samples has higher nodal density near the ± 1 domain boundaries, which enables a higher accuracy solution near the boundaries to compensate for the Runge phenomena (a common concern whereby larger oscillatory errors may occur near the edges of the domain due to lack of support for the approximation outside the boundaries of the domain). The coefficients that linearly combine the Chebyshev basis functions are approximated by the method of least squares, which generally requires a matrix inversion. A consistent choice of basis functions, weights, and node locations to ensure orthogonality means that the matrix required to be inverted in the Normal Equations of least squares is diagonal, thus the inverse is trivial and the coefficient computation is independent.

Bai's dissertation [8] extended the classical work of Clenshaw and Norton [7], and the more recent and related works of Feagin [17] Fukushima [18] and Shaver [19]. Bai established new convergence insights and optimized the solution of initial value problems utilizing vector-matrix formulations. Bai and Junkins applied MCPI to non-linear IVPs and orbit propagation in [9], and then showed promising results comparing MCPI to other higher order integrators such as Runge-Kutta-Nystrom 12(10). In [10] Bai and Junkins applied MCPI to efficiently solve Lambert's transfer problem in the usual Cartesian coordinates, and to solving an optimal control trajectory design problem more accurately and efficiently than the Chebyshev pseudospectral method. Notably, over intervals where the Picard iteration converges, there is no need to use a shooting

method to solve Lambert problems and similar two point boundary value problems (TPBVPs). In [11] Bai and Junkins use MCPI in three-body station-keeping control problem for halo orbits, formulated as a sequence of TPBVPs. Subsequent publications by Junkins et al. [12], [13], and [14] further clarify the concept and derivation of MCPI and orthogonal approximation in general, and apply the method to various problems in astrodynamics. The most recent publication [15] discusses how the MCPI algorithm for the IVP has been made into an easily accessible library, mainly focused on the case of Cartesian coordinates.

A full derivation of MCPI is outside the scope of this paper. Instead we present flow charts in Figure 1 that briefly summarize the algorithms represented in the compact vector/matrix formulation, which is computationally the most efficient way to implement the method. The above references provide detailed derivations, as well as examples and results that demonstrate the power of the MCPI algorithms with regard to efficiency and accuracy. Additionally, those references contain comparisons to other well-known integrators including high-order Runge-Kutta methods and the Gauss-Jackson method.

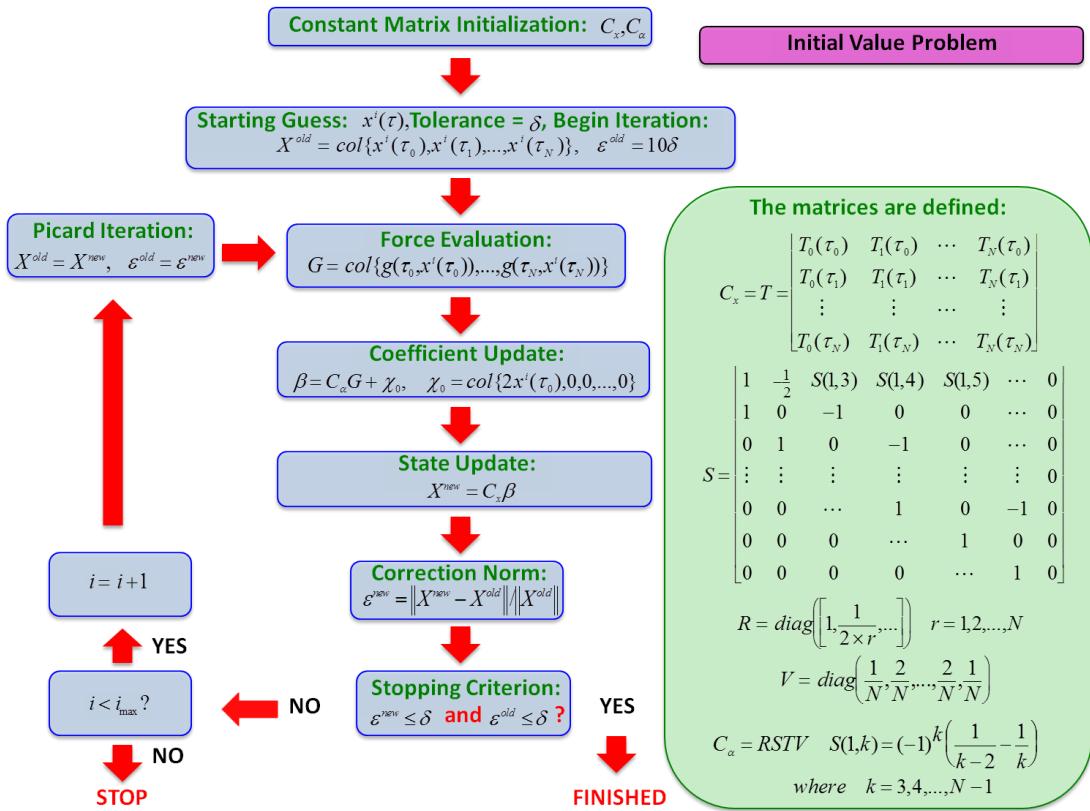


Figure 1: Vector matrix form for the Initial Value Problem. The procedure is very similar for the Boundary Value Problem, with the minor differences being a few elements in the S matrix and the inclusion of the final boundary condition.

KS TRANSFORMED KEPLERIAN LAMBERT PROBLEM

In the KS transformed u -space the time variable has been transformed to a form of eccentric anomaly, and the final eccentric anomaly is now unknown. To determine this the Lambert TPBVP problem is solved analytically in the KS u -space (using Eqn (9)) for an iterative approx-

imation of eccentric anomaly, and is transformed back to (x, y, z) . The Lambert/Kepler time – eccentric anomaly relationship is iterated by a Newton/Secant method to converge on the correct eccentric anomaly. An initial guess for the eccentric anomaly is computed from the dot product of the initial and final position vectors. Typically 6 iterations are required to achieve an accuracy of 10^{-10} or more precise. Engineering precision of course, typically requires fewer digits for applications. See Figures 2a – 2d.

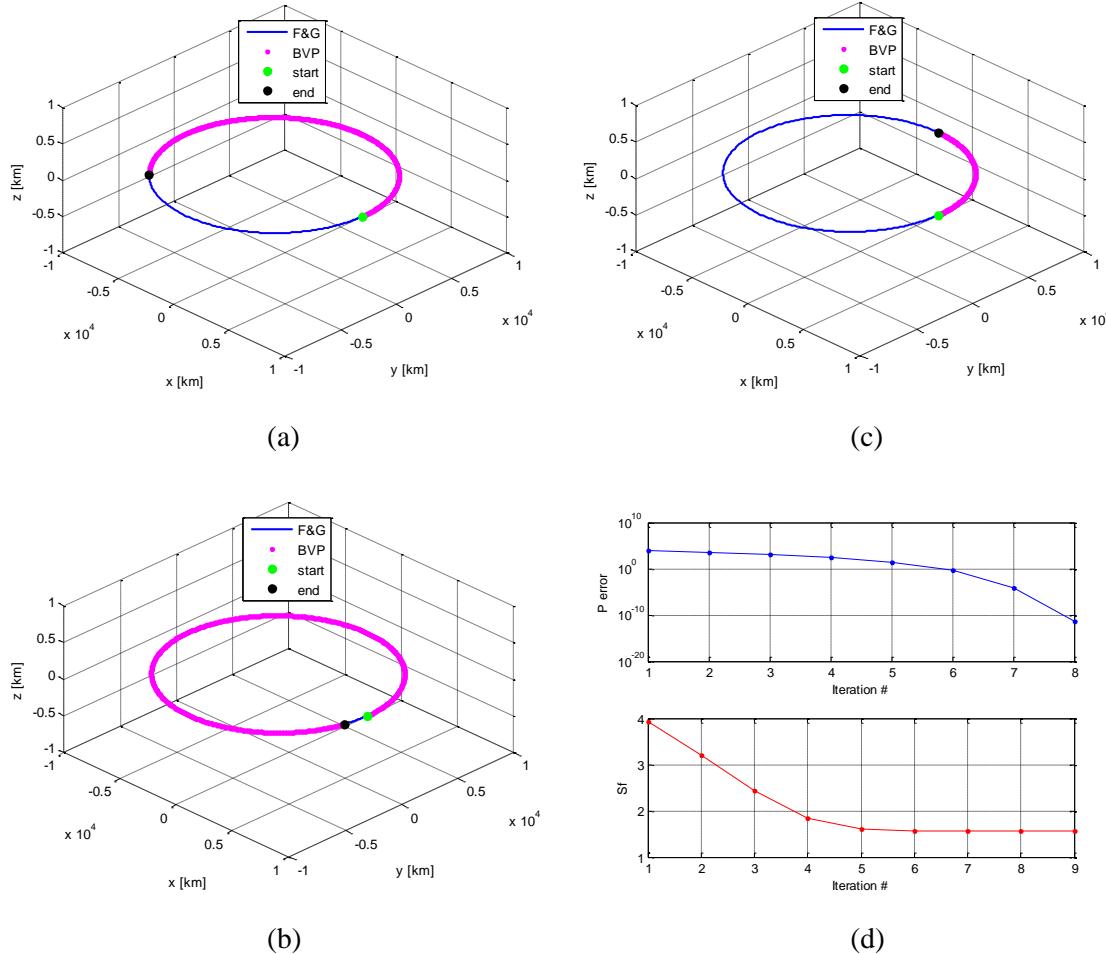


Figure 2: Lambert’s Problem is solved analytical using Eqn (8). Increasing orbital arc lengths are shown in (a) – (c), with a typical iteration convergence pattern shown in (d).

Determining the correct sign on the final boundary condition for the TPBVP is important to ensure that the desired trajectory is actually the one that is being integrated. For a given set of boundary conditions, there are two possible solutions that depend on the sign of the final position in \mathbf{u} -space. If the angle between the initial and final boundary conditions is less than 180 degrees, and the number of revolutions is odd (during the first orbit it is considered odd and during the second orbit it is even), then there is no sign change on the final position. Still in the odd orbit, if there is more than a 180 angle between the initial and final position then there is a sign change on

the final boundary condition. The opposite sign convention occurs for an even orbit. If a retrograde orbit is required then the sign convention is again reversed.

Solving Lambert's Problem analytically provides the final eccentric anomaly (corresponding to the final time), and also a *warm start* solution approximation for solving the perturbed problem. The two-body TPBVP is solved using MCPI with the known final eccentric anomaly computed analytically. Previous MCPI results [8] for solution of a LEO TPBVP have been limited to convergence over a maximum of 38% of an orbit. Implementing this KS transformation has enabled the interval of convergence to be vastly improved, without resorting a local linearization-based shooting method; the maximum convergence attainable is now $\sim 96\%$ of an orbit for the LEO case. Similar results are achievable for the MEO, GTO and HEO cases, and are shown in Figure 3.

The following four test case orbits (Table 1) were used for studying the performance MCPI Cartesian and KS algorithms. All the results presented in this paper are machine precision accuracy, maintaining 15 digits of precision in the Hamiltonian.

Table 1: Test Case Orbits

| Orbit Type | Semimajor axis (a) | Eccentricity (e) |
|------------|------------------------|----------------------|
| LEO | 8000 km | 0.125 |
| MEO | 10963 km | 0.4 |
| GTO | 26352 km | 0.6 |
| HEO | 32890 km | 0.8 |

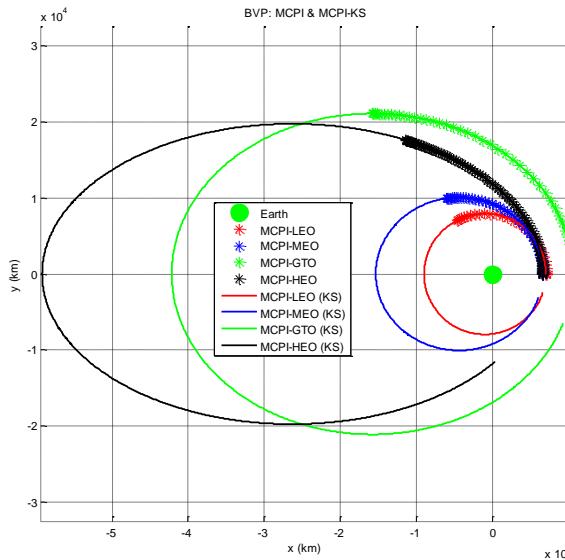


Figure 3: The Keplerian KS implementation has a far greater domain of convergence than the Cartesian implementation, for all four test cases.

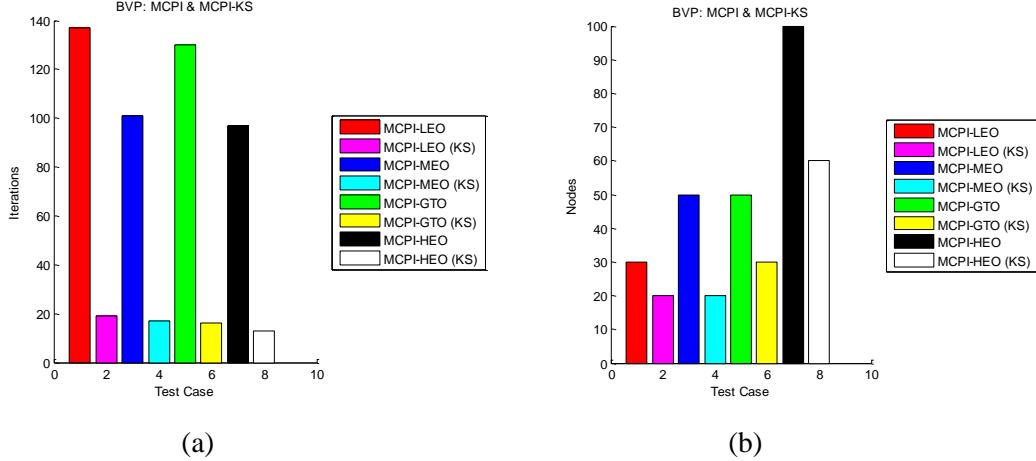


Figure 4: Comparison of the number of iterations (a) and number of nodes (b) required by MCPI Cartesian and MCPI KS for converging with 15 digits of accuracy over the same arc length (max attainable by MCPI Cartesian).

PERTURBED LAMBERT PROBLEM IN KS VARIABLES: A PICARD ITERATION METHOD

The perturbed TPBVP is solved in the same way as the unperturbed problem described in the previous section. The EGM2008 gravity model [16] is implemented with spherical harmonic degree and order (40,40).

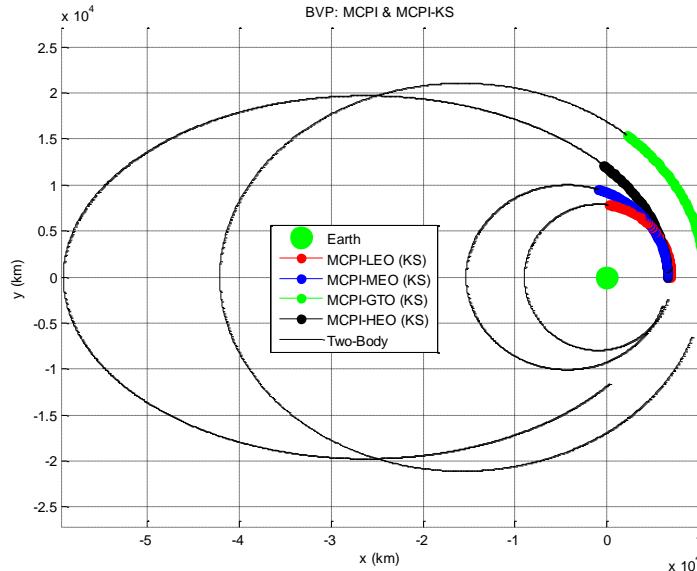


Figure 5: The MCPI KS implementation showing the domain of convergence for the perturbed orbit (40,40) with respect to the domain of convergence for the KS unperturbed orbit.

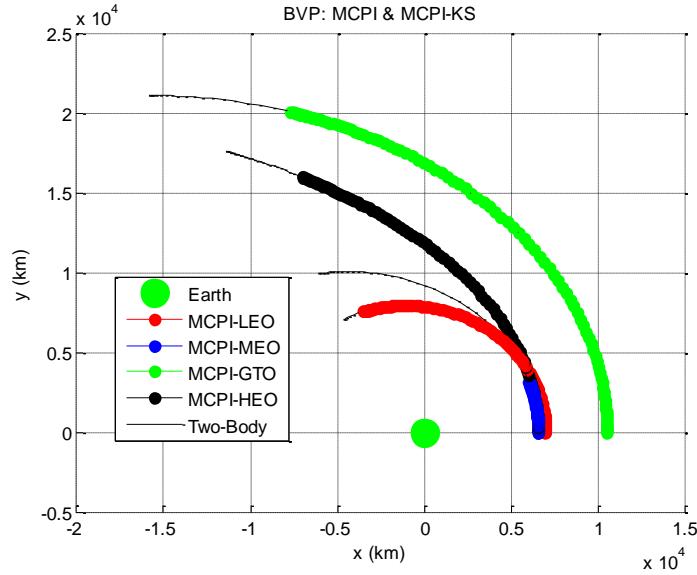


Figure 6: The MCPI Cartesian implementation showing the domain of convergence for the perturbed orbit (40,40) with respect to the domain of convergence for the Cartesian unperturbed orbit.

In both the KS and Cartesian cases the domain of convergence is reduced in the presence of perturbations. This is expected as the algorithms are now required to approximate more rapidly varying changes in the trajectory. The Cartesian seems to perform better than the KS in this perturbed environment.

We can also solve the perturbed TPBVP for single and multi-revolutions in KS space. It requires a shooting method. The Method of Particular Solutions [20] is employed to solve this problem and requires solving the Initial Value Problem using MCPI.

INITIAL VALUE PROBLEM IN KS VARIABLES: A PICARD ITERATION METHOD

The KS transformation can also be applied to the Initial Value Problem (IVP). Similarly, the final eccentric anomaly is determined analytically using the classical Kepler equation for the change in eccentric anomaly given the time. As expected, we find that the domain of convergence achievable for the IVP is greatly increased compared with previous MCPI results [9] in Cartesian coordinates with time as the independent variable. Figures 7 and 8 show the superiority of the KS transform with regard to number of iterations, and number of nodes required to achieve the desired propagation orbital arc length, for the two-body problem.

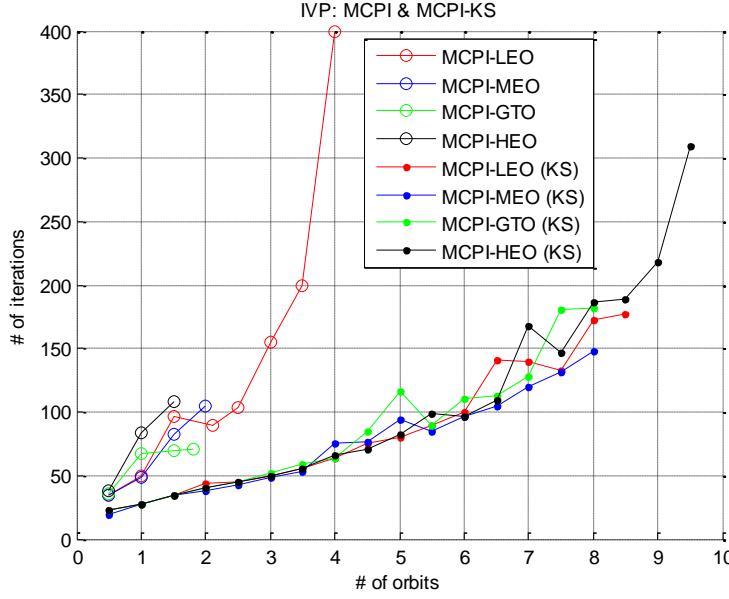


Figure 7: Comparative performance of MCPI and MCPI-KS for the four different test cases (LEO, MEO, GTO, HEO). The KS algorithm achieves a convergence domain of about 8.5 orbits with much fewer iterations required than the standard method.

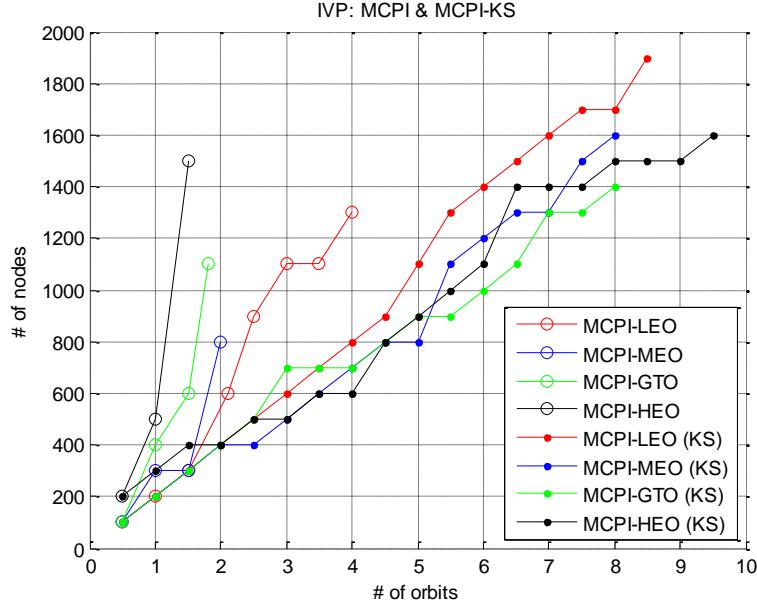


Figure 8: Comparative performance of MCPI and MCPI-KS for the four different test cases (LEO, MEO, GTO, HEO). The KS algorithm requires less nodes for propagating the same orbital arc length.

The EGM2008 gravity model [16] is implemented for solving perturbed orbits. A two-body plus J2 warm start and a two-body final eccentric anomaly are used to start the iterations. The

perturbed KS IVP is solved with a final two-body eccentric anomaly value as the initial guess plus 2%. This ensures that the converged solution overshoots the desired final eccentric anomaly value. This slightly longer than desired solution is fit with MATLAB's *interp* and *spline* commands to generate the states at the desired times, thus allowing the "shorter" solution to be extracted over the desired time interval.

Figures X and Y show the superiority of the KS solution to the Cartesian solution for the perturbed cases (40,40). The interpolation procedure does reduce the accuracy of the results significantly, showing millimeter precision between the MCPI KS solution and the MCPI Cartesian solution. We are currently investigating new methods of interpolating the data to allow MCPI's machine convergence precision to be maintained.

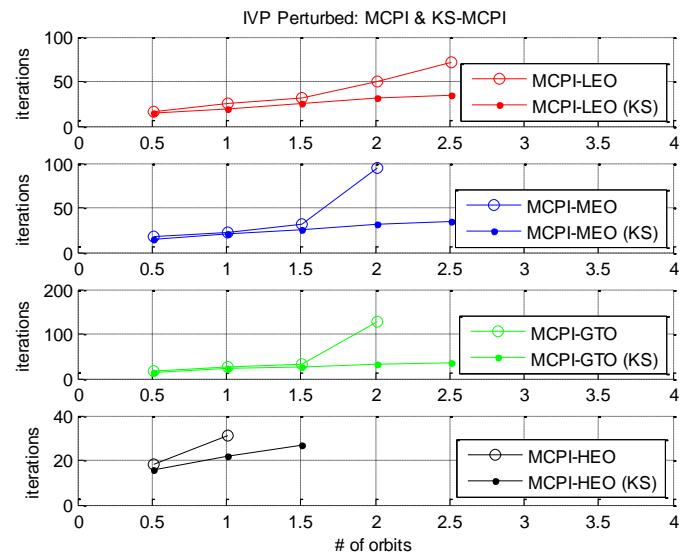


Figure 9: Comparative performance of Cartesian and KS for the four different perturbed (40,40) test cases (LEO, MEO, GTO, HEO). The KS algorithm requires less iterations for propagating the same orbital arc length.

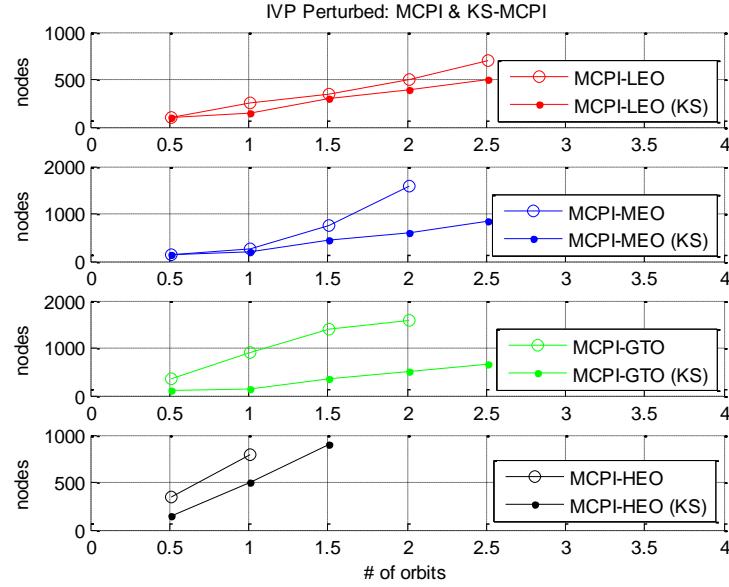


Figure 10: Comparative performance of Cartesian and KS for the four different perturbed (40,40) test cases (LEO, MEO, GTO, HEO). The KS algorithm requires less nodes for propagating the same orbital arc length.

PERTURBED LAMBERT PROBLEM IN KS VARIABLES: A SHOOTING TECHNIQUE

The multi-revolution TPBVP can be solved using a shooting technique via the Method of Particular Solutions. A guess for the initial velocity is found by solving the analytical TPBVP. We then propagate the orbit using the IVP approach in KS variables. More on the Method of Particular Solutions can be found in [20].

The increased convergence demonstrated by KS over Cartesian for the IVP allows the TPBVP to be solved up to 2.5 LEO perturbed orbits. This is a fantastic new result for MCPI and it means that this larger domain of transfer orbit possibilities can be explored for finding the optimal transfer trajectory. Figure 11 shows two orbits (A and B) and a transfer trajectory linking the two.

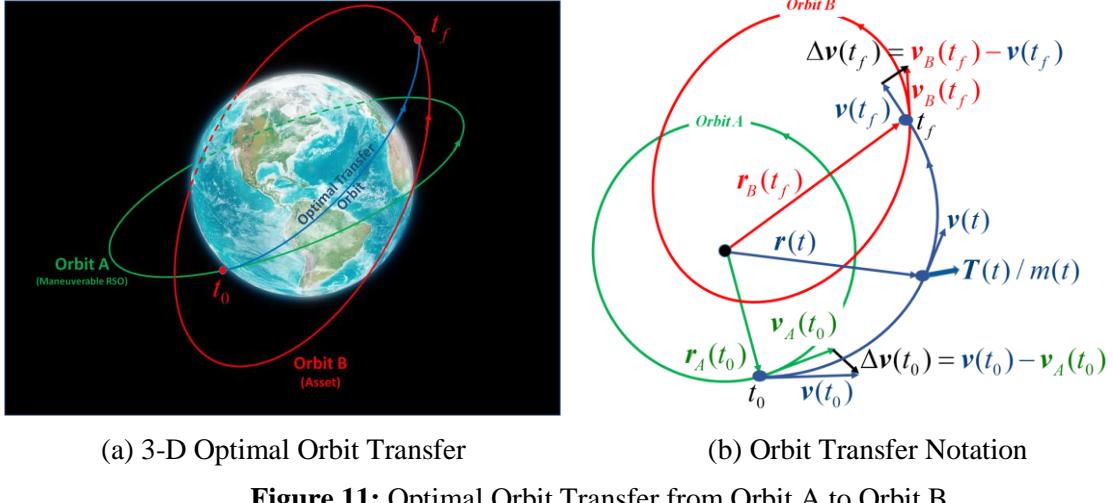


Figure 11: Optimal Orbit Transfer from Orbit A to Orbit B

Exploring the domain for the optimal trajectory requires generating a pork chop that shows the Δv required to transfer between two orbits at any location around the orbit and at any time. Figure 12 shows this for two LEO orbits. The respective orbital elements are shown in Table 2. The pork chop was created using the Method of Particular Solutions, and each orbit was propagated using the analytical F&G solution. Since we have seen such promising results with the increase of the domain of convergence for the perturbed IVP, we anticipate that integrating the KS transformed equations of motion will produce a similar pork chop. The range would be about 2.5 LEO orbits.

Table 2: Orbit Elements for Example Optimal Orbit Transfer Problem

| osculating elements | symbol | Orbit A | Orbit B |
|----------------------------------|---------------|----------------|----------------|
| semi-major axis | a | 8000km | 9000km |
| eccentricity | e | 0.125 | 0.050 |
| inclination | i | 0 | 5deg |
| longitude of the ascending node | Ω | 0 | 0 |
| argument of perigee | ω | 0 | 0 |
| initial true anomaly @ $t_0 = 0$ | f_0 | 0 | 117deg |

Performance Index J [km/sec] 2

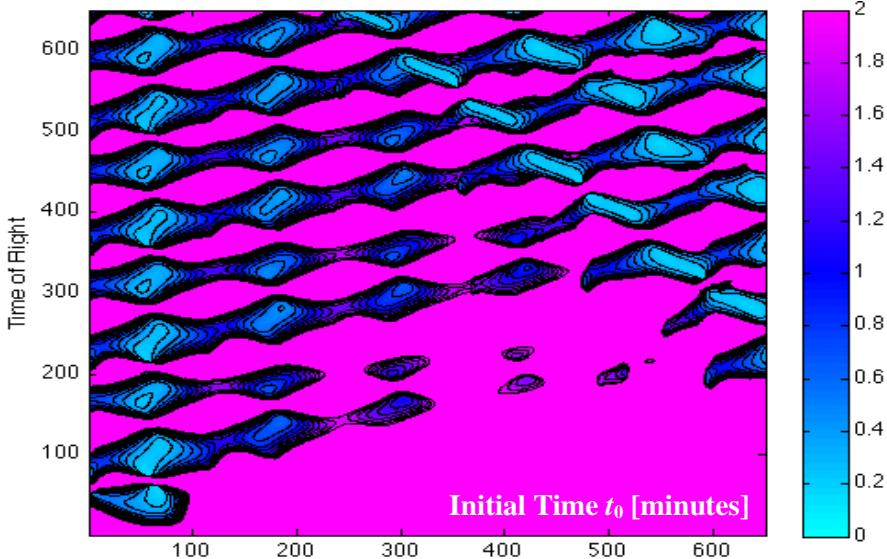


Figure 12: Minimum Velocity Orbit Transfer Maneuvers From Orbit A to Orbit B. The maximum allowable delta v is 2 km/s.

CONCLUSION

Novel methods are presented that are designed to efficiently solve the Lambert Problem for a state-of-the-art gravity field model. Two methods are the focus of the paper. In the first method, we consider the two-point-boundary value version of the Modified Chebyshev Picard Iteration approach, mapped into KS variables. This formulation in theory solves the Keplerian and perturbed Lambert problems without requiring a shooting method. Numerical results show that Picard iterations for this method converges reliably for about 98% of one LEO unperturbed orbit. The second method utilizes the MCPI initial value solver which can be solved over arbitrary time intervals, and establishes a shooting method suitable for multiple revolution solutions of the perturbed Lambert problem in KS variables. Both methods are illustrated with a family of numerical demonstrations that show that the formulations are valid and offer some advantages relative to conventional algorithms in Cartesian coordinates. In particular, we believe that we have established the first method that solves the perturbed Lambert problem without the necessity of a state transition matrix, albeit for the fractional orbit case. The second method is introduced for completeness for solving the multi-revolution case in KS variables, but at this point, it is qualitatively analogous to other well-known shooting methods in Cartesian coordinates. Computational efficiency optimization will be also be addressed in future studies. These new algorithms and exciting new results address improvements in the solutions of a fundamental problem in astrodynamics and should find widespread use in contemporary and future applications.

ACKNOWLEDGEMENTS

We acknowledge the other members of the MCPI team: Julie Read, Brent Macomber, Austin Probe, Donghoon Kim, Xiaoli Bai and James Turner.

We thank our sponsors: AFOSR, AFRL, LANL, and Applied Defense Systems for their support and collaborations under various contracts and grants.

REFERENCES

- [1] Battin, R., *An Introduction to the Mathematics and Methods of Astrodynamics*. Reston, VA: American Institute of Aeronautics and Astronautics, Inc, revised ed., 1999.
- [2] Prussing, J. E., *A Class of Optimal Two-Impulse Rendezvous Using Multiple-Revolution Lambert Solutions*, Journal of Astronautical Sciences, April 2000.
- [3] Kustaanheimo, P. and Stiefel, E. L., *Perturbation theory of Kepler motion based on spinor regularization*, Journal fur die Reine und Angewandt Mathematik 218, 204-219, 1965.
- [4] Engels, R. C. and Junkins, J. L., *The Gravity-Perturbed Lambert Problem: A KS Variation of Parameters Approach*, Celestial Mechanics, May 1981.
- [5] Woollards, R. and Junkins, J., *A New Solution for the General Lambert Problem*. 37th Annual AAS Guidance & Control Conference, 2014
- [6] Levi-Civita T., *Sur la regularization du problem des trois corps*, Acta Mathematica 42, 99-144, 1920.
- [7] Clenshaw, C. W. and Norton, H. J., *The Solution of Nonlinear Ordinary Differential Equations in Chebyshev Series*, The Computer Journal, 6(1):88-92, 1963.
- [8] Bai, X., *Modified Chebyshev-Picard Iteration Methods for Solution of Initial Value and Boundary Value Problems*, PhD. Dissertation, Texas A&M University, College Station, Tex, USA, 2012.
- [9] 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.
- [10] Bai, X., and Junkins, J. L., *Modified Chebyshev-Picard Iteration Methods for Solution of Boundary Value Problems*, Advances in the Astronautical Sciences, vol. 140, pp. 381–400, 2011.
- [11] 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.
- [12] Junkins, J., Bani Younes, A., Woollards, R., and Bai, X., *Orthogonal Approximation in Higher Dimensions: Applications in Astrodynamics*, ASS 12-634, JN Juang Astrodynamics Symp, College Station, TX, June 14-26, 2012.
- [13] Junkins, J. L., Bani Younes, A., Woollards, R M., and Bai, X., *Orthogonal Approximation in Higher Dimensions: Applications in Astrodynamics*, submitted to The Journal of the Astronautical Sciences, June, 2013.
- [14] Junkins, J. L., Bani Younes, A., Woollards, R. M., and Bai, X., *Picard Iteration, Chebyshev Polynomial and Chebyshev Picard Methods: Application in Astrodynamics*, accepted in The J of the Astronautical Sciences, July, 2013.
- [15] Macomber, B., Woollards, R. M., Probe, A., Bani Younes, A., Junkins, J. L., *Modified Chebyshev Picard Iteration for Efficient Numerical Integration of ordinary Differential Equations*, Advanced Maui Optical and Space Surveillance Technologies Conference, Maui, Hawaii, June 2013.
- [16] Bani Younes, A., *Orthogonal Polynomial Approximation in Higher Dimensions: Applications in Astrodynamics*, Ph.D. dissertation, Texas A&M Univ, College Station, TX, 2013.
- [17] Feagin, T. and Nacozy, P.. “Matrix Formulation of the Picard Method for Parallel Computation,” *Celestial Mechanics and Dynamical Astronomy*, Vol. 29, Feb. 1983, pp 107–115.
- [18] Fukushima, T. “Vector Integration of Dynamical Motions by the Picard-Chebyshev Method,” *The Astronomical Journal*, Vol. 113, Jun. 1997, pp. 2325–2328.

[19] Shaver, J. S. *Formulation and Evaluation of Parallel Algorithms for the Orbit Determination Problem*, Ph.D. Dissertation, Department of Aeronautics and Astronautics, MIT, Cambridge, MA, Mar. 1980.

[20] Miele, A. and Iyer, R. R., General Technique for Solving Nonlinear, Two-Point Boundary-Value Problems Via the Method of Particular Solutions, J of Optimization Theory and Applications: V 5, No 5, pp 392-399, 1970.

Validation of Accuracy and Efficiency of Long-Arc Orbit Propagation Using the Method of Manufactured Solutions and the Round-Trip-Closure Method

**Robyn Woollards, Ahmad Bani Younes, Brent Macomber,
Austin Probe, Donghoon Kim and John L. Junkins**

*Texas A & M University, Aerospace Engineering Dept, H.R. Bright, 3141 TAMU, College
Station, TX, 77843-3141*

ABSTRACT

For conservative systems, a common method for validating accuracy is that the Hamiltonian or a similar energy integral of a converged solution maintains constancy to a desired tolerance. While a Hamiltonian metric is a very useful for conservative systems, for non-conservative systems a Hamiltonian check is not applicable and other methods for validating solution accuracy must be employed. While we can utilize various ad-hoc methods for comparing the state history from a new integrator with some other well-tested code, or compare solutions using various accuracy tunings for a given method, there always remains uncertainty since rigorous convergence conclusions are difficult when comparing approximate solutions.

Two independent measures of solution accuracy are considered in this paper, based on the ***Method of Manufactured Solutions (MMS) and the Round-Trip-Closure Method (RTC)***. These metrics have the attractive property that they are both theoretically exactly zero if the integrator introduces zero error. For RTC, the convergence test is applied directly to the original differential equations and boundary conditions, whereas for MMS, a close neighbor of the unknown exact solution is established, with a known small perturbing force. The neighboring solution is the exact solution of the original differential equations with the known small perturbing force. Application of the solution methodology to this slightly perturbed problem permits strong conclusions on the algorithm's accuracy of convergence.

MMS and RTC metrics are useful for virtually any numerical process for solving differential equations. MMS and RTC are useful in evaluating the relative merits of competing algorithms; the utility of these ideas are demonstrated in an accuracy study for three numerical integrators: Modified Chebyshev Picard Iteration (MCPI), an 8th order Gauss Jackson (GJ8) algorithm and Runge-Kutta-Nystrom (RKN12(10)). We utilize an intermediate order spherical-harmonic gravity (40,40) model. Since this problem is conservative, we check the Hamiltonian constancy with MMS and RTC. Results demonstrate the consistency of the two metrics and high efficiency vs accuracy of MCPI relative to the other integrators, for long-arc orbit propagation. MMS is readily applied to MCPI, since the solution process produces automatically an interpolating polynomial for the state variables. However, for most methods, one must introduce an auxiliary interpolation process, as discussed herein. We show MMS and RTC errors for these state of the art algorithms.

1. INTRODUCTION

A common method for validating the accuracy of numerical integrators is confirming that the Hamiltonian of an apparently converged solution maintains constancy to a desired tolerance for the time interval over which the computation was performed. This Hamiltonian metric is a useful test for conservative systems. For non-conservative systems, for example a Low Earth Orbit (LEO) that is under the influence of aerodynamic drag, the Hamiltonian check is not applicable and other methods for validating the accuracy of the solution must be employed. While we can utilize various ad hoc methods of comparing the state history (ephemeris) from a new integrator with some other well-tested integrator, or compare solutions using a given method with itself, there always remains uncertainty since neither solution is exact.

Two independent measures of solution accuracy are introduced, based on the ***Method of Manufactured Solutions (MMS) and the Round-Trip-Closure Method (RTC)***. These metrics have the attractive property that they are both zero if the integrator introduces zero error. Healy and Berry [1] used a number of different tests to study the accuracy of two numerical integrators, Runge-Kutta 45 and an 8th order Gauss-Jackson. Both MMS, or Zadunaisky's test as they refer to it, and RTC, or Reverse Test, are mentioned in their work.

An additional benefit for using MMS and RTC metrics over the Hamiltonian metrics when determining the accuracy of numerical solutions is specifically important for symplectic integrators. Symplectic integrators, such as the Implicit Runge-Kutta methods, enforce the accuracy of the Hamiltonian at each iteration and thus reduce the “purity” of the Hamiltonian when it is used as a final *independent* check of accuracy of the converged solution. We utilize a recently developed path approximation method: the Modified Chebyshev Picard Iteration (MCPI). MCPI differs from the symplectic methods in that the constancy of the Hamiltonian is not explicitly enforced during the numerical integration process, but MCPI relies instead on the property that the Picard iteration process is theoretically a contraction mapping attracted to the exact solution under the conditions of the Picard convergence theorem. The conditions under which Picard is proven to be a contraction to the solution is that the acceleration function be smooth and at least once differentiable, and that the time interval over which the solution is sought belongs to a bounded interval (typically, less than three orbit periods) and finally, that the starting orbit approximation must have a bounded error relative to the unknown true solution. Typically, convergence is achieved over large time intervals, even with a straight line starting approximation, however a “warm start” closer to the solution is needed for efficient convergence. MMS and RTC lead to “exact” metrics for performing accuracy checks and comparisons between different numerical integrators. In the case of RTC, the convergence test is directly on the original differential equations and boundary conditions. In the case of MMS, a close neighbor of the sought solution is established with a small perturbing force for which a given smooth approximate solution exactly satisfies the slightly perturbed differential equations. The metrics associated with MMS and RTC are easily computed. These “external” validation/accuracy checks are not to be confused with adaptive tuning of the solution segments and the number of nodes when orthogonal function approximations are fused with Picard iteration to maintain accuracy of acceleration approximation and numerical quadratures that are a part of the implementation of MCPI. Furthermore, the MMS and RTC metrics are useful for virtually any method for numerical solution of differential equations and therefore has utility in evaluating the relative merits of competing algorithms.

2. METHOD OF MANUFACTURED SOLUTIONS

The Method of Manufactured Solutions [2, 3, 4] computes an analytical function near to the actual problem of interest. A new system of differential equations, that is slightly different from the original problem, is constructed and solved. The solution to this problem has an analytical solution, which if compared to the numerical solution will allow the numerical accuracy of the integrator to be tested.

Consider the nonlinear differential equation, with specified initial conditions:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t)), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad t_0 \leq t \leq t_f \quad (1)$$

Suppose that the differential equation of Eq. (1) does not have an analytical solution. Further suppose that an approximate solution $\mathbf{x}_r(t)$ is available that does not satisfy Eq. (1) exactly but is believed to satisfy it with “small” but unknown errors. Suppose that $\mathbf{x}_r(t)$ is smooth and at least once differentiable. On substituting $\mathbf{x}_r(t)$ into Eq. (1), we can obtain an explicit algebraic solution for the error as

$$\mathbf{d}_r(t) = \dot{\mathbf{x}}_r(t) - \mathbf{f}(t, \mathbf{x}_r(t)) \quad (2)$$

or

$$\dot{\mathbf{x}}_r(t) = \mathbf{f}(t, \mathbf{x}_r(t)) + \mathbf{d}_r(t) \quad (3)$$

We can compute the norm $\|\mathbf{d}_r(t)\|$ to see if it is sufficiently small to consider $\mathbf{x}_r(t)$ a good starting approximation.

Comparing Eqs (3) and (1) and reflecting for a moment, it is clear that $\mathbf{x}_r(t)$ is the exact analytical solution of the slightly disturbed differential equation

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t)) + \mathbf{d}_r(t), \quad \mathbf{x}(t_0) = \mathbf{x}_r(t_0), \quad t_0 \leq t \leq t_f \quad (4)$$

Since we have a candidate solution with a small $\|\mathbf{d}_r(t)\|$, Eq (4) can be considered a very close neighboring problem to the original one of Eq. (1), but with the important advantage that we know the exact analytical solution

$\mathbf{x}_r(t)$. One can argue that whatever numerical method is under evaluation for solving Eq. (1), can be evaluated on the perturbed system of Eq. (4), which should prove slightly more difficult for the numerical solver than solving the original unforced system of Eq. (1). Whatever numerical solution process under study can be used to solve the perturbed system of Eq. (4) and obtain an approximate solution $\tilde{\mathbf{x}}(t)$, however we know the exact solution of Eq. (4) is $\mathbf{x}_r(t)$, so we can compute the exact solution error $\tilde{\mathbf{x}}(t) - \mathbf{x}_r(t)$ at any/all times. If the numerical method we are studying to solve Eq. (1) gives, for example, a 15 digit solution of the more difficult perturbed problem of Eq. (4), then we can be justifiably optimistic that it will solve Eq. (1) with similar precision. In particular, if $\frac{\|\mathbf{d}_r(t)\|}{\|\mathbf{f}(t, \mathbf{x}_r(t))\|} < \varepsilon$, say (10^{-14}) , then it is virtually certain that the numerical method used to generate $\tilde{\mathbf{x}}(t)$, when applied to Eq. (1), will produce a solution with 14 or more significant figures.

The main weakness with the MMS test is that that acceleration, $\dot{\mathbf{x}}_r(t)$, must be obtained by differentiating an approximation to the converged velocity solution. The quality of the approximation limits the ability of MMS to test the quality of the integrator. This is a drawback for the *step* integrators, but for MCPI the coefficients of the acceleration fit are already available due to the *path* approximation nature of the algorithm. No differentiation of the state trajectory approximation of the velocity is necessary, thus allowing MMS to honestly test the accuracy of the integrator, without the necessity of introducing other approximations.

3. ROUND-TRIP-CLOSURE

Round Trip Closure (RTC) is a test that measures the accumulative error that results during numerical integration. Consider the nonlinear differential equation, with specified initial conditions:

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t)), \quad \mathbf{x}(t_0) = \mathbf{x}_0, \quad t_0 \leq t \leq t_f \quad (5)$$

Suppose that the differential equation of Eq. (5) does not have an analytical solution. An approximate solution may be obtained through numerical integration. As a specific example, consider propagating the trajectory of a spacecraft about the Earth, with specified initial conditions and final time. The gravitational acceleration experienced by the spacecraft varies as a function of position along the trajectory.

Having computed the trajectory, the final position is used as the new initial position, and the final time as the new initial time, as shown in Eq. (6).

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t)), \quad \mathbf{x}(t_0) = \mathbf{x}_f, \quad t_f \leq t \leq t_0 \quad (6)$$

The new initial conditions are propagated backwards in time along the trajectory in order to recover the initial conditions used for the forward integration. The error metric is evaluated as follows:

$$J = 0.5 \left(\left(\frac{|\mathbf{r}_0 - \mathbf{r}_f|}{|\mathbf{r}_0|} \right) + \left(\frac{|\mathbf{v}_0 - \mathbf{v}_f|}{|\mathbf{v}_0|} \right) \right) \quad (7)$$

For MCPI, which is a path approximation integrator, slightly varying the node locations along the reverse trajectory allows the solution to be computed using a slightly different gravity field, thus eliminating possible bias and/or aliasing issues that may arise due to performing the reverse calculations at the exact same node locations as the forward solution. Healy and Berry [1] mention this as being a disadvantage when testing step integrators in a perturbed environment. They comment that it does not measure any reversible integration error as it will be cancelled on the reverse trip when the sign of the step changes. However, the RTC method has been used extensively for performing numerical integration accuracy checks [5]. A high fidelity numerical integrator should recover the initial conditions with an accuracy of 14 significant figures, however, this will begin to decrease with long-term propagation and is a good measure of the achievable long-term propagation range of a numerical integrator.

4. MODIFIED CHEBYSHEV PICARD ITERATION

Modified Chebyshev Picard Iteration (MCPI) is an attractive numerical method for solving linear or non-linear differential and integral equations. MCPI combines the discoveries of two great mathematicians: Emile Picard (Picard Iteration) and Rafnuty Chebyshev (Chebyshev Polynomials), and recent developments in the associated linear algebra by Bai, Junkins, Feagin, et al. [6-15]. The original fusion of orthogonal approximation theory and Picard iteration was apparently introduced by Clenshaw and Norton in 1963 [16].

Picard observed that any first order differential equation

$$\dot{x}(t) = f(t, x(t)), \quad (8)$$

with an initial condition $x(t_0) = x_0$, and any integrable right hand side may be rearranged, without approximation, to obtain the following integral equation:

$$x(t) = x(t_0) + \int_{t_0}^t f(\tau, x(\tau)) d\tau. \quad (9)$$

This re-arrangement, at first glance, does not appear to have made any progress, since the unknown trajectory $x(t)$ is contained in the integrand on the right hand side. A sequence of approximate solutions $x^i(t)$, ($i = 1, 2, 3, \dots, \infty$), of the true solution $x(t)$ that satisfies this differential equation may be obtained through Picard iteration using the following Picard sequence of approximate paths $\{x^0(t), x^1(t), \dots, x^{i-1}(t), x^i(t), \dots\}$:

$$x^i(t) = x(t_0) + \int_{t_0}^t f(\tau, x^{i-1}(\tau)) d\tau, \quad i = 1, 2, \dots . \quad (10)$$

Picard proved an important convergence theorem that essentially states that for smooth, differentiable, single-valued nonlinear functions $f(t, x(t))$, there is a time interval $|t_f - t_0| < \delta$ and a starting trajectory $x^0(t)$ satisfying $\|x^0(t) - x(t)\|_\infty < \Delta$, for suitable finite bounds (δ, Δ) , the Picard sequence of trajectories represents a contraction operator that converges to the unique solution of the initial value problem. What was not apparently appreciated until the work of Bai et al. [6-8], is that these bounds are surprisingly large for the main initial value problem in celestial mechanics (δ exceeds an orbit period, and the starting approximation for the trajectory can be a straight line osculating initial position and velocity).

In the first step toward the MCPI method, orthogonal Chebyshev polynomials are used as basis functions to approximate the integrand in Eq. (10) along the previous approximate trajectory $x^{i-1}(t)$. Unlike traditional step-by-step integrators, for example the Runge-Kutta methods, MCPI is a *path iteration method* in which long state trajectory arcs are approximated and updated at all time instances on each iteration. Under usually satisfied and known theoretical circumstances, we can show [6] that the Picard sequence is a contraction mapping guaranteed to converge to the solution of Eq. (8). The system dynamics are normalized such that the timespan of integration is projected onto the domain $\{-1 \leq \tau \leq +1\}$ of the Chebyshev polynomials, so the system states can be conveniently approximated using the Chebyshev basis functions. The orthogonal nature of the basis function means that the coefficients that linearly scale the basis functions can be computed independently as simple ratios of inner products with no matrix inversion. Since the Chebyshev polynomials are a complete set, we can achieve machine precision (if desired) approximation of any smooth integrand in Eq. (10) on each iteration and the resulting converged trajectory can approach a machine precision solution of the differential equation over large time spans. In Bai's dissertation [6], she found that the actual time interval over which convergence is obtained for problems in celestial mechanics is over 3 orbits (for the case of an initial value problem and the usual Cartesian coordinate formulation of the equations of motion). For the two-point boundary value problem in Cartesian coordinates, however, she found that the time interval for which Picard iteration converges is reduced to about 0.38 of an orbit period.

As a consequence of the independence and orthogonality of the basis functions, the coefficients multiplying the Chebyshev basis functions may be computed, as an inner product of the basis functions with the integrand, in parallel by separate independent threads with no matrix inversion required. This independently computable integrand approximation coefficients is the first of two available layers of parallelization in the MCPI method. The second layer of parallelization is much more important and is enabled by the fact that acceleration over the entire state trajectory $x^{i-1}(t)$ permits us to compute independently and simultaneously. Thus the calculation of the integrand functions (which must be computed as a function of the system states along the current approximate trajectory, at the nodes, as required for the discrete inner products leading to the approximation coefficients) can be performed at all nodes simultaneously in parallel processor threads. Using MCPI, over an order of magnitude speedup from traditional methods is achieved in serial processing, and an additional one-to-two orders of magnitude, are achieved in parallel architectures, depending on the specifics of the parallel implementation.

A key feature of MCPI is a non-uniform cosine sampling of the $\{-1 \leq \tau \leq 1\}$ domain of the called Chebyshev-Gauss-Lobatto (CGL) nodes: $\tau_j = -\cos(j\pi/N)$, $j = 0, 1, 2, \dots, N$. This set of samples has higher nodal density near the ± 1 domain boundaries, which enables a higher accuracy solution near the boundaries to compensate for the Runge phenomena (a common concern whereby larger oscillatory errors may occur near the edges of the domain due to lack of support for the approximation outside the boundaries of the domain). The coefficients that linearly combine the Chebyshev basis functions are approximated by the method of least squares, which generally requires a matrix inversion. A consistent choice of basis functions, weights, and node locations to ensure orthogonality means that the matrix required to be inverted in the Normal Equations of least squares is diagonal, thus the inverse is trivial and the coefficient computation is independent.

Bai's dissertation [6] extended the classical work of Clenshaw and Norton [16], and the more recent and related works of Feagin [15] Fukushima [17] and Shaver [18]. Bai established new convergence insights and optimized the solution of initial value problems utilizing vector-matrix formulations. Bai and Junkins applied MCPI to non-linear IVPs and orbit propagation in [7], and then showed promising results comparing MCPI to other higher order integrators such as Runge-Kutta-Nystrom 12(10). In reference [8] Bai and Junkins applied MCPI to efficiently solve Lambert's orbit transfer problem in the usual Cartesian coordinates, and to solve an optimal control trajectory design problem, formulated in polar coordinates, more accurately and efficiently than the Chebyshev pseudospectral method. Notably, over intervals where the Picard iteration converges, there is no need to use a shooting method to solve Lambert problems and similar two point boundary value problems (TPBVPs). Furthermore, the MCPI algorithm renders the indirect (Pontryagin's Principle) state, co-state differential equations solvable without a shooting method for a large class of problems. In [9] Bai and Junkins use MCPI in three-body station-keeping control problem for halo orbits, formulated as a sequence of TPBVPs. Subsequent publications by Junkins et al. [10], [11], and [12] further clarify the concept and derivation of MCPI and orthogonal approximation in general, and apply the method to various problems in astrodynamics. Reference [13] discusses an implementation of the MCPI algorithm for the IVP as an easily accessible library, mainly focused on the case of Cartesian coordinates.

A full derivation of MCPI is outside the scope of this paper. Instead we present flow charts in Fig. 1 that briefly summarize the algorithms represented in the compact vector/matrix formulation, which is computationally the most efficient way to implement the method. The above references provide detailed derivations, as well as examples and results that demonstrate the power of the MCPI algorithms with regard to efficiency and accuracy. Additionally, those references contain comparisons to other well-known integrators including high-order Runge-Kutta methods and the Gauss-Jackson method.

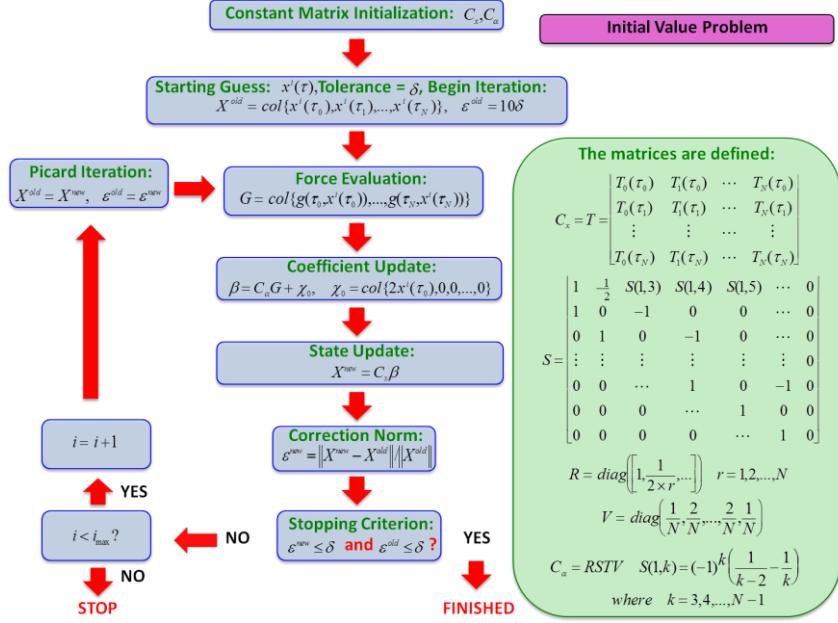


Fig. 1. Vector matrix form for the Initial Value Problem

5. RESULTS AND DISCUSSION

Method of Manufactured Solutions

We consider three test cases (LEO, GEO, Molniya) and propagate each for 10 two-body orbital periods. The MMS error metric for determining the closeness of the analytical solution is computed as

$$\frac{\|d_r(t)\|}{\|f(t, x_r(t))\|} < \epsilon, \text{ say } (10^{-14}), \quad (11)$$

and the error metric for determining the closeness of the numerical approximation to the analytical solution is calculated as the maximum absolute difference between the position norms of the two trajectories (numerical and analytical).

In this paper we consider only MCPI and RKN(12)10. As mentioned earlier, a drawback for *step* integrators is that the velocity must be approximated and differentiated to obtain acceleration. This limits the ability of MMS to quantify the accuracy of the algorithm. For the MATLAB implementation of RKN(12)10 used for this analysis, it is possible to specify an input time array on a cosine distribution, thus allowing the velocity to be fit and approximated with Chebyshev polynomials, and then differentiated using Chebyshev polynomials of the second kind [6]. Interpolation with Chebyshev polynomials is very accurate compared with other methods such as power series approximation. The MATLAB Gauss-Jackson algorithm used for this analysis does not permit specification of a desired input time array, thus leading to a less accurate approximation. Until further investigation is performed with regard to better fitting techniques, we have decided to exclude it from the analysis.

Fig. 2 shows MMS error for MCPI and RKN(12)10 plotted as a function of increasing orbital distance. Both integrators are relatively stable, maintaining 12 digits of accuracy over this interval. For this GEO test case, MCPI appears to be more accurate than RKN(12)10. Fig. 3 shows how the MMS error for MCPI (LEO, GEO, Molniya) gradually increases over a range of decreasing Hamiltonian accuracies. The test is performed over 50 orbits.

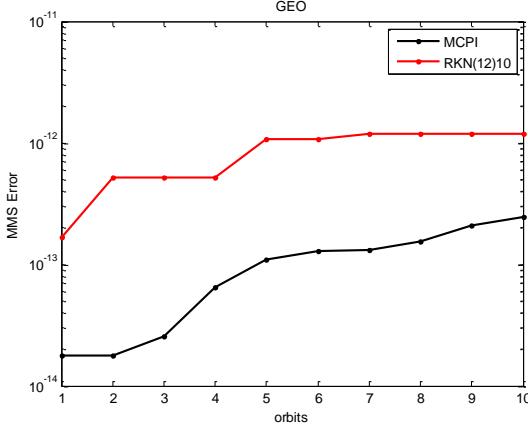


Fig. 2 Comparison of MMS errors for MCPI and RKN(12)10, over 10 orbits.

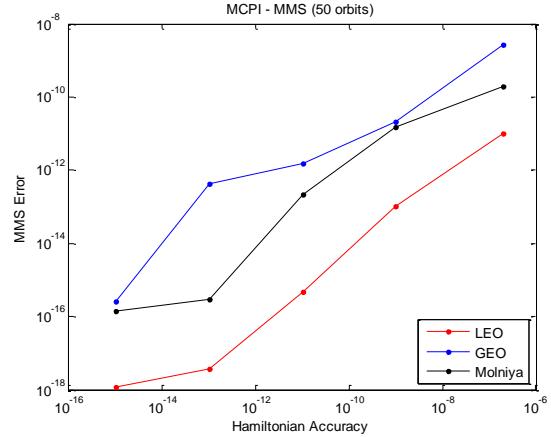


Fig. 3 Comparison of MMS errors for MCPI and RKN(12)10, as a function of Hamiltonian accuracy.

Round Trip Closure

We consider 3 test cases (LEO, GEO, Molniya) and propagate each forward in time for 50 two-body orbital periods. The final states are then used as the initial conditions and time is reversed to propagate backwards and recover the initial conditions. Fig. 4 shows the Molniya test case integrated forward and backward in time with MCPI. The node locations on the return trajectory are intentionally selected to be at different positions from that on the forward trajectory. This is done to eliminate aliasing and error cancellation that may arise from performing the reverse calculation at the exact same node locations as the forward solution. A subtraction of the states is done over the entire trajectory and the difference is plotted in Fig. 4.

Neither solution in Fig. 4 is correct because each is effected by numerical error that accumulates during the propagation. The left most values are most significant because these represent the error in the recovery of the initial conditions after propagating forward for 50 orbits and then backwards for 50 orbits. The error is smallest on the far right because this is where the final states at the end of 50 orbits of forward propagation are set as the initial conditions for the backward propagation. The general trend shows the error slowly accumulating over time. In addition to the general trend there are also variations that occur periodically as a function of position around the orbit. See the enlarge inset at the bottom left of the figure.

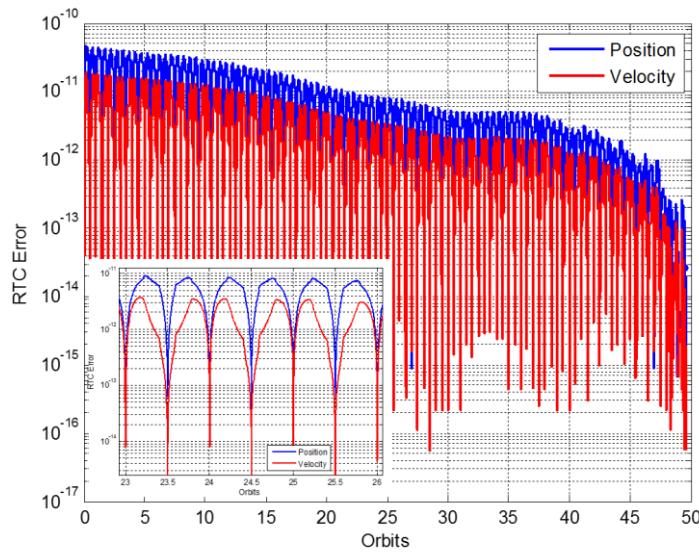


Fig. 4. Reverse Integration Closure over 50 Molniya orbits (Period = 12 hours).

RTC errors are computed for each test case over a range of tolerances ($1 \times 10^{-7}, 1 \times 10^{-9}, 1 \times 10^{-11}, 1 \times 10^{-13}, 1 \times 10^{-15}$). The error metric, Eq. (12), is used to quantify the accuracy of each RTC solution. In Figs. 5 through 7, the RTC error is plotted as a function of Hamiltonian accuracy. As expected, the general trend shows the RTC error increasing as the accuracy of the solution decreases. In general, MCPI seems to perform better at higher solution accuracies than Gauss-Jackson. For the Molniya case at low Hamiltonian accuracies MCPI has a large RTC error. This is likely due to sub-optimal segmentation and node distribution. The algorithm was tuned by hand for the high accuracy cases, and the same segmentation is used for the low accuracy cases. We anticipate improved results for the low accuracy range once the optimal segmentation scheme is implemented [19]. Fig. 8 shows how the RTC error changes as the propagation time increases from 10 to 50 orbits. We note that the MCPI Molniya test case shows some variation (improved accuracy after 50 orbit RTC compared with 40 orbit RTC). This is likely depicting the variations observable in Fig. 4. In Fig. 9 we take the average of RTC error values on either side of the desired 50 orbit propagation distance. Considerable fluctuation is evident over this small range, thus highlighting the importance of computing an average value. Overall, the three integrators show relatively similar stability trends.

$$J = 0.5 \left(\left(\frac{|\mathbf{r}_0 - \mathbf{r}_f|}{|\mathbf{r}_0|} \right) + \left(\frac{|\mathbf{v}_0 - \mathbf{v}_f|}{|\mathbf{v}_0|} \right) \right) \quad (12)$$

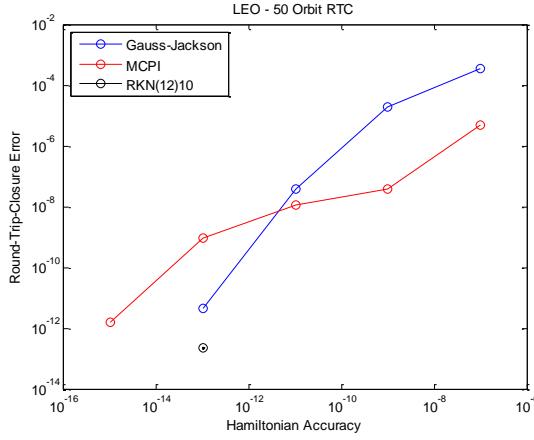


Fig. 5 LEO: Comparison of RTC errors over 50 orbits for Gauss-Jackson, MCPI and RKN(12)10, as a function of Hamiltonian accuracy.

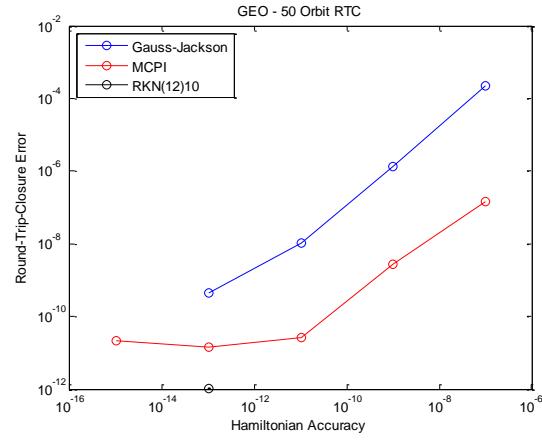


Fig. 6 GEO: Comparison of RTC errors over 50 orbits for Gauss-Jackson, MCPI and RKN(12)10, as a function of Hamiltonian accuracy.

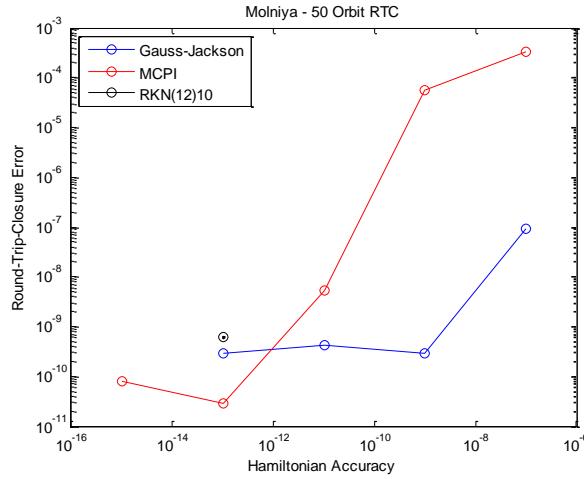


Fig. 7 Molniya: Comparison of RTC errors over 50 orbits for Gauss-Jackson, MCPI and RKN(12)10, as a function of Hamiltonian accuracy.

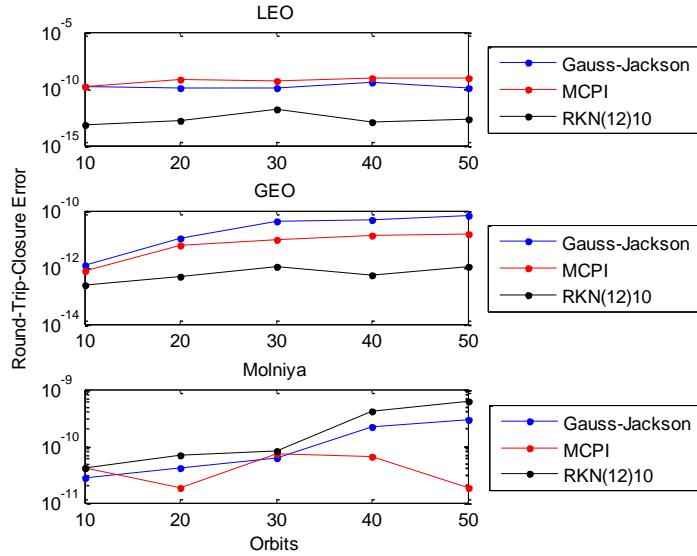


Fig. 8 Comparison of RTC errors for Gauss-Jackson, MCPI and RKN(12)10, as a function of orbital propagation distance. Top to bottom: LEO, GEO, Molniya.

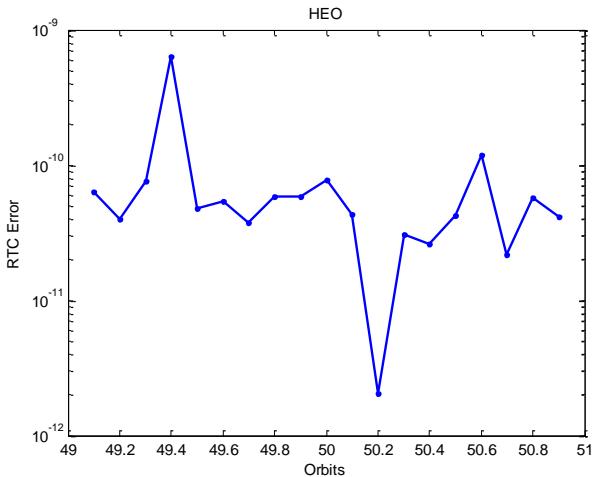


Fig. 9 Variation of RTC error over two orbital periods in the vicinity of 50 orbits RTC.

6. CONCLUSION

A common method for validating the accuracy of numerical integrators is confirming that the Hamiltonian of the converged solution maintains machine precision for the time interval over which the computation was performed. This is a useful test for conservative systems, but for non-conservative systems the Hamiltonian check is no longer sufficient and other methods for validating the accuracy of the solution must be employed.

Two methods, the **Method of Manufactured Solutions (MMS)** and **Round Trip Closure (RTC)** are employed for comparing the accuracy of three numerical integrators: Modified Chebyshev Picard Iteration, Gauss-Jackson and Runge-Kutta-Nystrom. The two tests reveal all three integrators are comparably stable for long-term integration for LEO, GEO and highly eccentric orbits. Of the three MCPI is the most efficient for serial computation (see sister paper [20]) and is also ideally suited for parallel computation to enable further speedup.

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. Thanks also to Julie Read (Texas A&M) for enlightening discussions.

8. REFERENCES

- [1] Berry, M., and Healy, L., *Comparison of Accuracy Assessment Techniques for Numerical Integration*. 13th AAS/AIAA Space Flight Mechanics Meeting, American Institute of Aeronautics and Astronautics, 2003.
- [2] Zadunaisky, P. E., *A Method for the Estimation of Errors Propagated in the Numerical Solution of a System of Ordinary Differential Equations*, In Contopoulos, G., editor, *The Theory of Orbits in the Solar System and in Stellar Systems*, pp. 281–287, New York, 1966. International Astronomical Union, Academic Press.
- [3] Zadunaisky, P. E., *On the Accuracy in the Numerical Computation of Orbits*, In Giacaglia, G. E. O., editor, *Periodic Orbits, Stability and Resonances*, pp. 216–227, Dordrecht, Holland, 1970. D. Reidel Publishing Company.
- [4] Zadunaisky, P. E., *On the Estimation of Errors Propagated in the Numerical Integration of Ordinary Differential Equations*, *Numerische Mathematik*, Vol. 27, No. 1, 1976, pp. 21–39, 1976.
- [5] Hadjifotinou, K. G. and Gousidou-Koutita, M., *Comparison of Numerical Methods for the Integration of Natural Satellite Systems*, *Celestial Mechanics and Dynamical Astronomy*, Vol. 70, No. 2, pp. 99–113, 1998.
- [6] Bai, X., *Modified Chebyshev-Picard Iteration Methods for Solution of Initial Value and Boundary Value Problems*, PhD. Dissertation, Texas A&M University, College Station, Tex, USA, 2012.
- [7] 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.
- [8] Bai, X., and Junkins, J. L., *Modified Chebyshev-Picard Iteration Methods for Solution of Boundary Value Problems*, *Advances in the Astronautical Sciences*, vol. 140, pp. 381–400, 2011.
- [9] 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.
- [10] Junkins, J., Bani Younes, A., Woollards, R.. and Bai, X., *Orthogonal Approximation in Higher Dimensions: Applications in Astrodynamics*, ASS 12-634, JN Juang Astrodynamics Symp, College Station, TX, June 14-26, 2012.
- [11] Junkins, J. L., Bani Younes, A., Woollards, R M., and Bai, X., *Orthogonal Approximation in Higher Dimensions: Applications in Astrodynamics*, submitted to *The Journal of the Astronautical Sciences*, June, 2013.
- [12] Junkins, J. L., Bani Younes, A., Woollards, R. M., and Bai, X., *Picard Iteration, Chebyshev Polynomial and Chebyshev Picard Methods: Application in Astrodynamics*, accepted in *The J of the Astronautical Sciences*, July, 2013.
- [13] Macomber, B., Woollards, R. M., Probe, A., Bani Younes, A., Junkins, J. L., *Modified Chebyshev Picard Iteration for Efficient Numerical Integration of ordinary Differential Equations*, Advanced Maui Optical and Space Surveillance Technologies Conference, Maui, Hawaii, June 2013.
- [14] Bani Younes, A., *Orthogonal Polynomial Approximation in Higher Dimensions: Applications in Astrodynamics*, Ph.D. dissertation, Texas A&M Univ, College Station, TX, 2013.
- [15] Woollards, R. and Junkins, J., *A New Solution for the General Lambert's Problem*., 37th Annual AAS Guidance & Control Conference, 2014.
- [16] Feagin, T. and Nacozy, P., *Matrix Formulation of the Picard Method for Parallel Computation*, *Celestial Mechanics and Dynamical Astronomy*, Vol. 29, Feb. 1983, pp 107–115.
- [17] Clenshaw, C. W. and Norton, H. J., *The Solution of Nonlinear Ordinary Differential Equations in Chebyshev Series*, *The Computer Journal*, 6(1):88-92, 1963.
- [18] Fukushima, T. “Vector Integration of Dynamical Motions by the Picard-Chebyshev Method,” *The Astronomical Journal*, Vol. 113, Jun. 1997, pp. 2325–2328.
- [19] Shaver, J. S. *Formulation and Evaluation of Parallel Algorithms for the Orbit Determination Problem*, Ph.D. Dissertation, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA, Mar. 1980.
- [20] Macomber, B., and Junkins, J., *Optimal Segmentation for Modified Chebyshev Picard Iteration*, In preparation.
- [21] Probe, A., Macomber, B., Kim, D., Woollards, R., Junkins, J., *Terminal Convergence Approximation Modified Chebyshev Picard Iteration for Efficient Numerical Integration of Orbital Trajectories*, Advanced Maui Optical and Space Surveillance Technologies Conference, Maui, Hawaii, Sep 2014.

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. Woollards, 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]

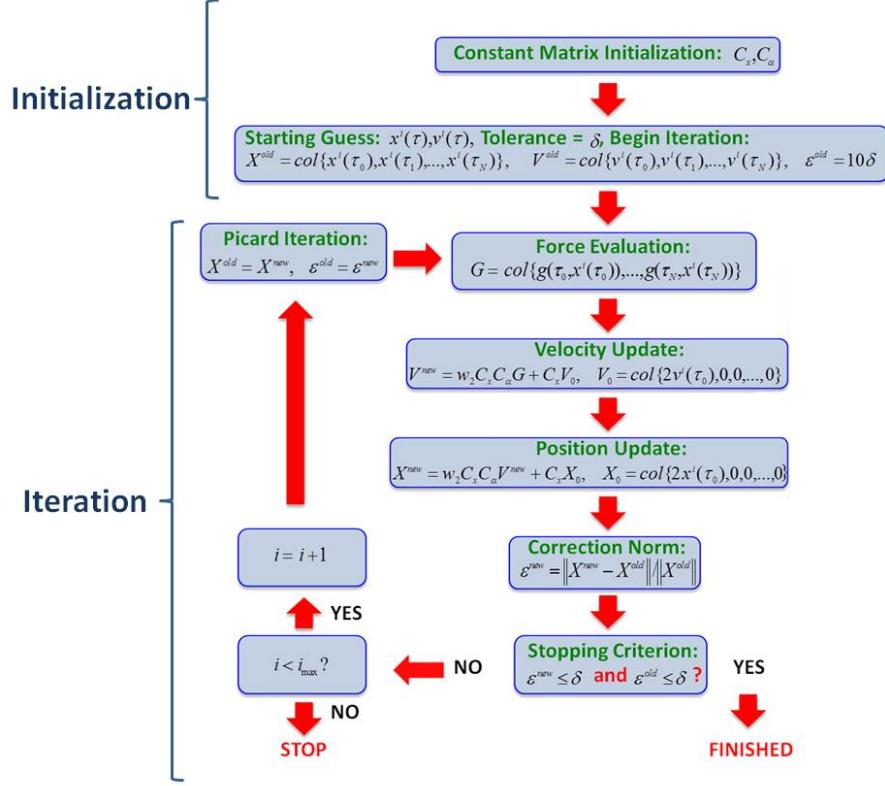


Figure 1: Second Order Cascade MCPI Algorithm for Solving Initial Value Problems

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 ($e \sim 0.7$) 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.

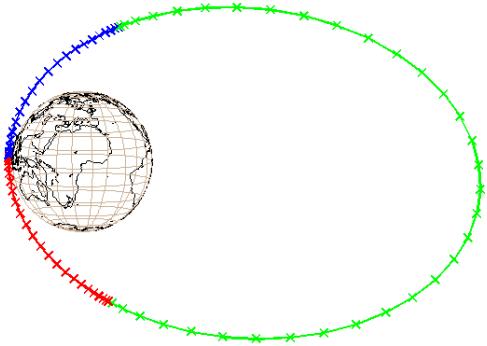


Figure 2: Example MCPI Segmentation Scheme

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]

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

$$\Delta g(\tau) = g(\tau, x(\tau)) - J2J6(x(\tau)) \quad (2)$$

$$g(\tau, x(\tau)) = J2J6(x(\tau)) - \Delta g(\tau) \quad (3)$$

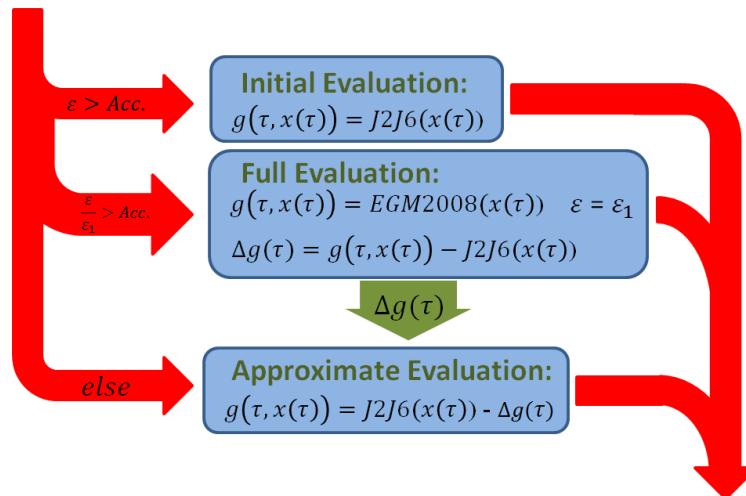


Figure 3: Gravity Function Evaluation Flow Chart

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.

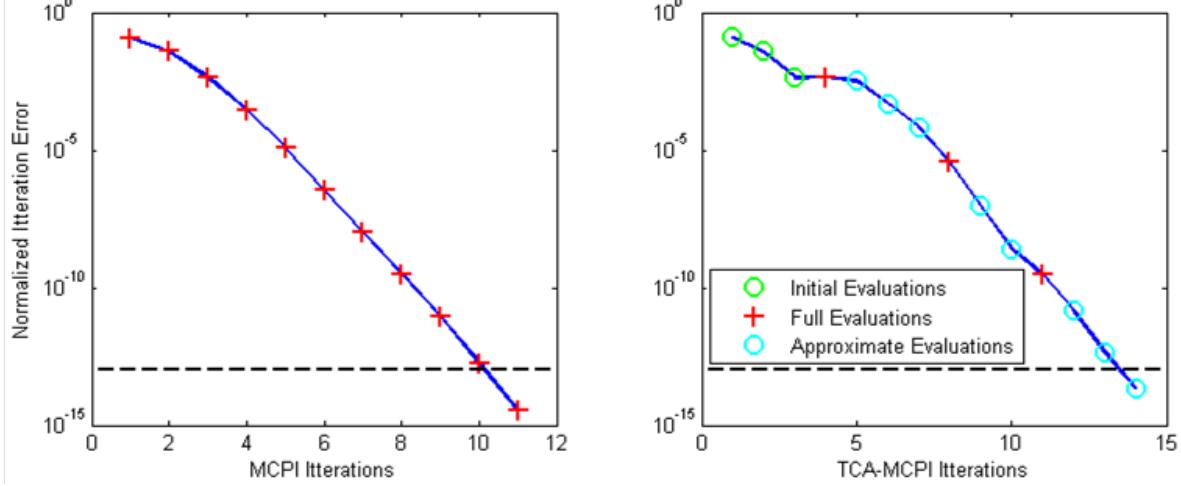


Figure 4: MCPI and TCA-MCPI Convergence Progress vs. Iteration Number

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.

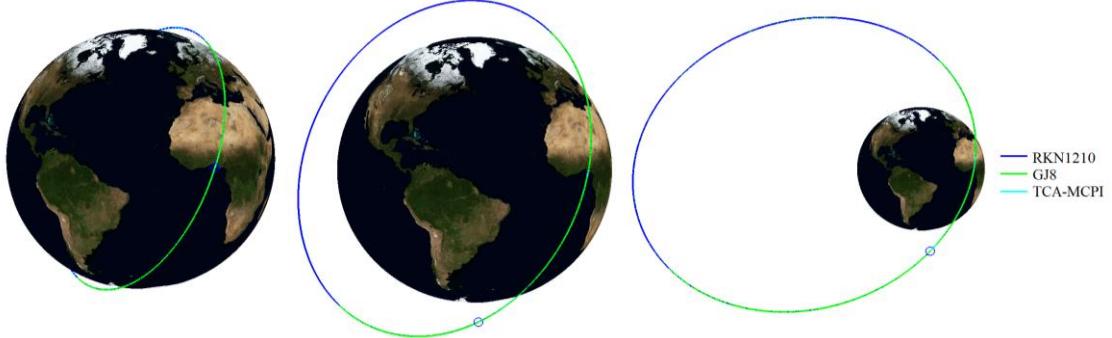


Figure 5: Circular, Moderately Eccentric and Highly Eccentric Orbital Trajectories with a LEO Perigee Altitude

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 |

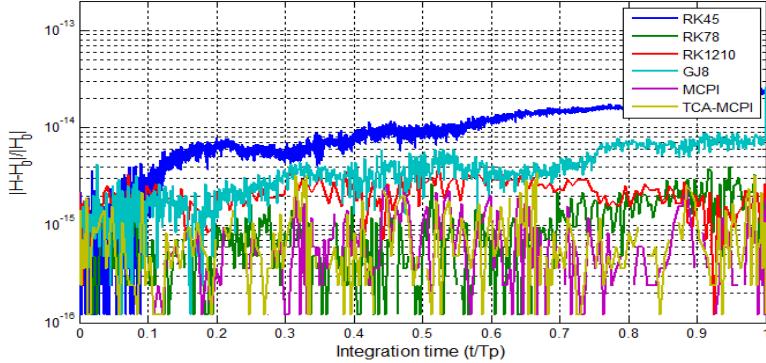


Figure 6: Case LC Relative Hamiltonian Preservation

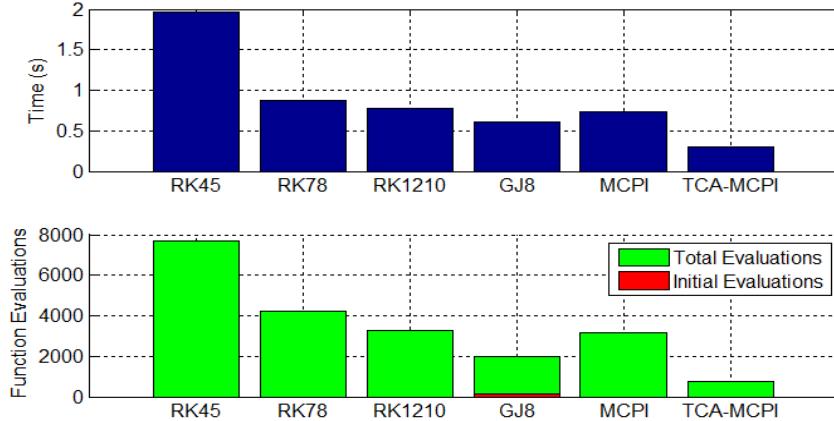


Figure 7: Case LC Timing and Function Evaluation Results

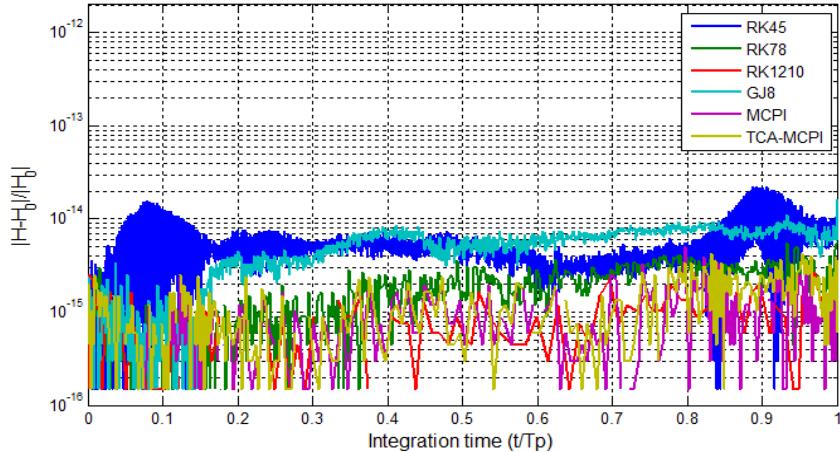


Figure 8: Case LME Relative Hamiltonian Preservation

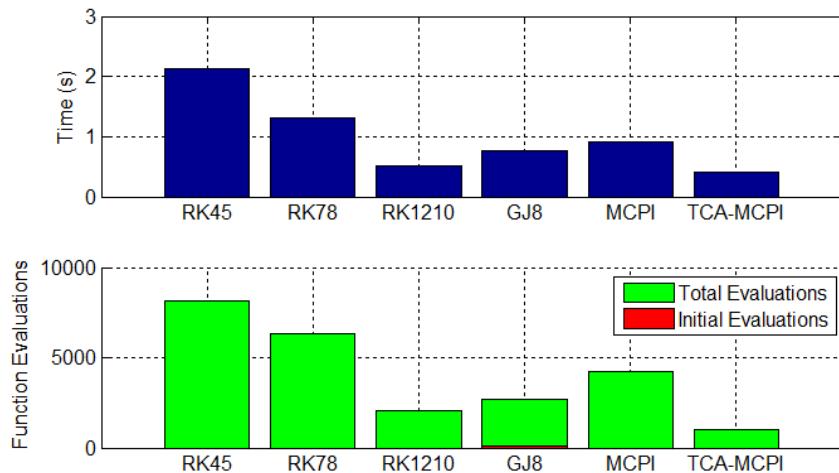


Figure 9: Case LME Timing and Function Evaluation Results

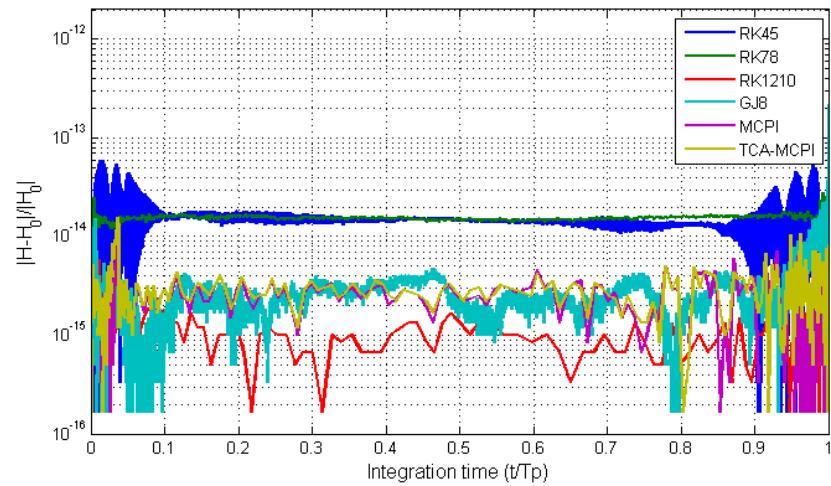


Figure 10: Case LHE Relative Hamiltonian Preservation

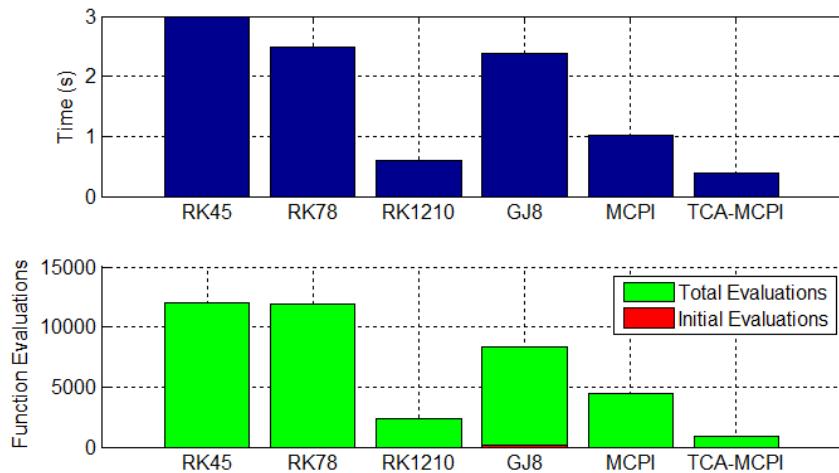


Figure 11: Case LHE Timing and Function Evaluation Results

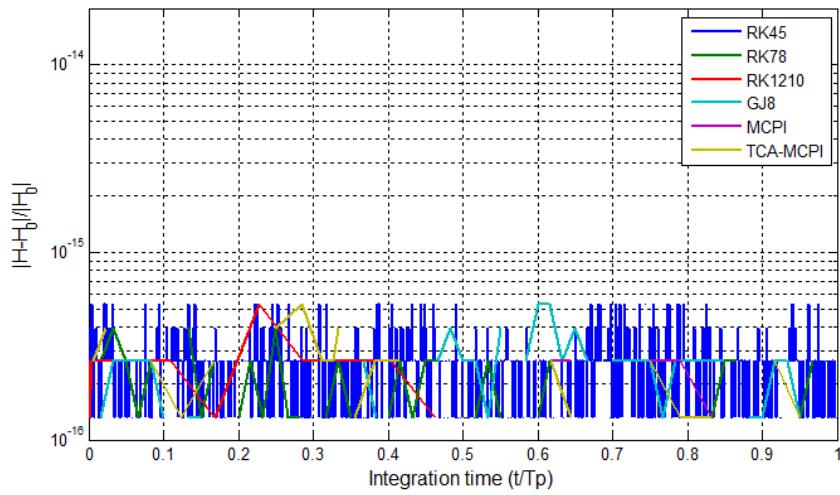


Figure 12: Case GC Relative Hamiltonian Preservation

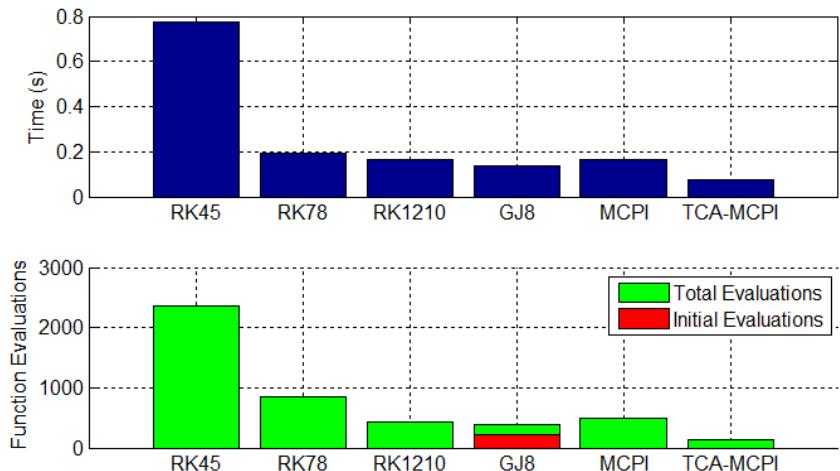


Figure 13: Case GC Timing and Function Evaluation Results

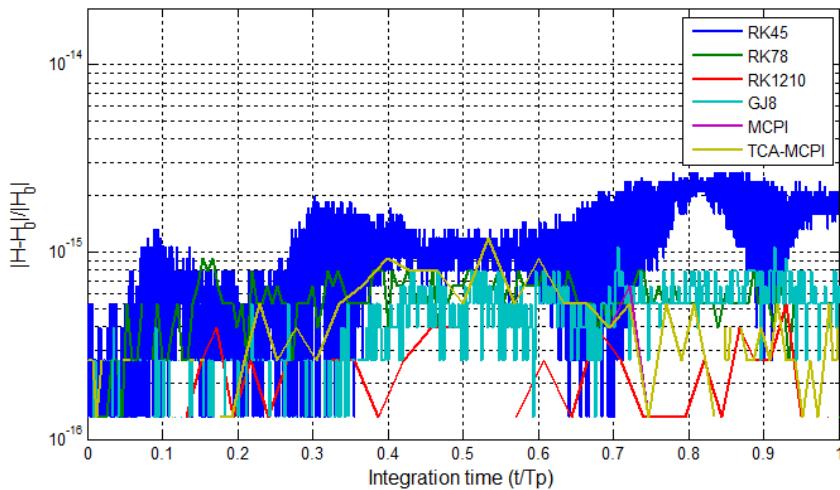


Figure 14: Case GME Relative Hamiltonian Preservation

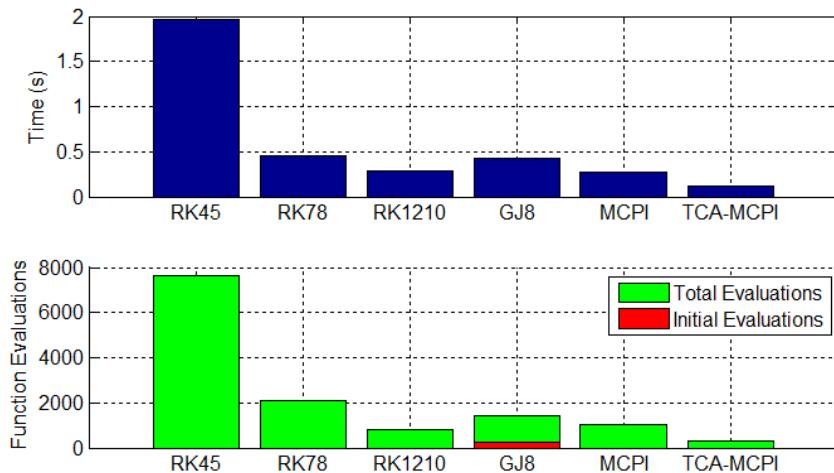


Figure 15: Case GME Timing and Function Evaluation Results

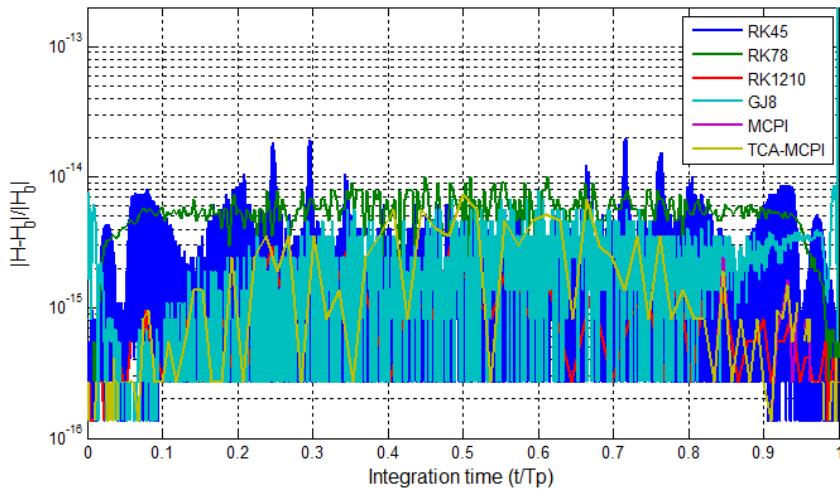


Figure 16: Case GHE Relative Hamiltonian Preservation

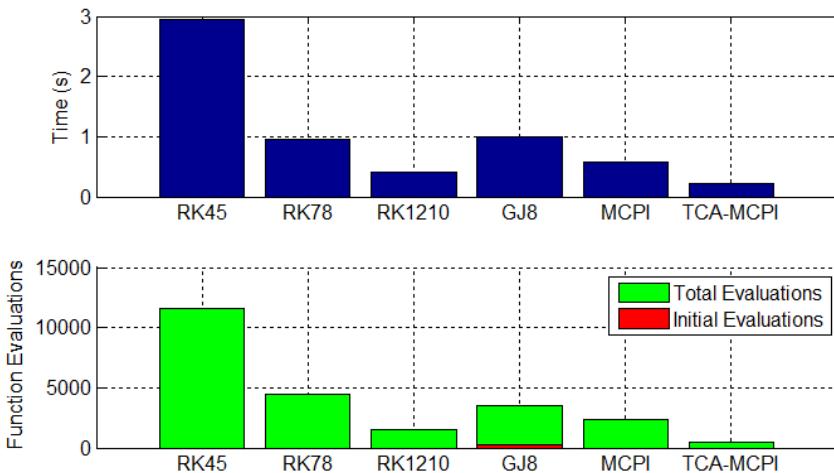


Figure 17: Case GHE Timing and Function Evaluation Results

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