The Journal of the Astronautical Sciences
https://doi.org/10.1007/s40295-019-00167-2

Accuracy and Efficiency Comparison of Six Numerical Integrators for Propagating Perturbed Orbits

Ahmed M. Atallah[1,2] · Robyn M. Woollands[3] · Tarek A. Elgohary[4] · John L. Junkins[3]

Published online: 21 May 2019

Abstract

We present the results of a comprehensive study in which the precision and efficiency of six numerical integration techniques, both implicit and explicit, are compared for solving the gravitationally perturbed two-body problem in astrodynamics. Solution of the perturbed two-body problem is fundamental for applications in space situational awareness, such as tracking orbit debris and maintaining a catalogue of over twenty thousand pieces of orbit debris greater than the size of a softball, as well as for prediction and prevention of future satellite collisions. The integrators used in the study are a $5^{th}/4^{th}$ and $8^{th}/7^{th}$ order Dormand-Prince, an 8^{th} order Gauss-Jackson, a $12^{th}/10^{th}$ order Runga-Kutta-Nystrom, Variable-step Gauss Legendre Propagator and the Adaptive-Picard-Chebyshev methods. Four orbit test cases are considered, low Earth orbit, Sun-synchronous orbit, geosynchronous orbit, and a Molniya orbit. A set of tests are done using a high fidelity spherical-harmonic gravity (70×70) model with and without an exponential cannonball drag model. We present three metrics for quantifying the solution precision achieved by each integration method. These are conservation of the Hamiltonian for conservative systems, round-trip-closure, and the method of manufactured solutions. The efficiency of each integrator is determined by the number of function evaluations required for convergence to a solution with a prescribed accuracy. The present results show the region of applicability of the selected methods as well as their associated computational cost. Comparison results are concisely presented in several figures and are intended to provide the reader with useful information for selecting the best integrator for their purposes and problem specific requirements in astrodynamics.

Keywords Orbit propagation · Numerical integration · Picard-Chebyshev

✉ Ahmed M. Atallah
aatallah@eng.ucsd.edu

Extended author information available on the last page of the article.

Introduction

The perturbed two-body problem is a fundamental problem in celestial mechanics that can only be solved through the use of numerical integration methods. Historically, the numerical integrators that have been considered for solving this problem can be categorized as either single-step methods or multi-step methods. For single-step methods, the state at a specific time $\boldsymbol{x}(t_k)$ is used to compute the state at some future time $\boldsymbol{x}(t_k + h)$ through a linear combination of weighted evaluations of the ordinary differential equation at intermediate times $t_k \leq t \leq t_k + h$. The method is considered "single-step" because only information from the current state is used to compute the state at a future time $\boldsymbol{x}(t_k + h)$. Multi-step methods differ in that the state at some future time is estimated using the current state value as well as state values at several previous times. Multi-step methods are also known as predictor-corrector methods because the value of the state at some future time is extrapolated from previous states in a forward estimation step (predictor step) and this is then refined in a backwards estimation step (corrector step), [24]. Implicit methods have also been considered for solution of the perturbed two-body problem. These methods approximate the differential equation function along the trajectory using orthogonal basis functions, such as Chebyshev or Legendre polynomials, and iterate the "path approximation" for large segments around the orbit, [4, 6, 29].

The family of explicit Runge-Kutta methods are an example of single-step methods. This family is generally further classified by considering two parameters. The first is the number of "stages", which refers to the number of function evaluations required at each time step, and the second is the order p, which refers to the local truncation error at each step h and matches a local Taylor series expansion with an error of $O(h)^{p+1}$, [13]. Butcher developed a classification of Runge-Kutta methods, known as the Butcher Tableau, where the methods were arranged based on the number of stages and their order, [11, 12]. Some simple explicit methods use fixed time steps where the forward prediction time interval is always the same, however more advanced methods make use of lower and higher order algorithms so that the difference in the state predicted by the two algorithms can be used to adaptively adjust the time step. The $5^{th}/4^{th}$ order Dormand-Prince (DP5(4)), the $8^{th}/7^{th}$ order Dormand-Prince (DP8(7)) and the $12^{th}/10^{th}$ order Runge-Kutta-Nystrom (RKN(12)10) are examples of adaptive step-size methods where the step-size control leads to essentially uniform errors for the duration over which integration is performed. DP5(4) combines a fifth and fourth order explicit Runga-Kutta method and requires seven stages whereas the DP8(7) combines an eighth and seventh order explicit Runge-Kutta method and requires 13 stages, [15, 16]. RKN(12)10 combines a twelfth and tenth order polynomial and operates directly on the second order system of differential equations (double integrator method), whereas the Dormand-Prince methods typically require decomposition of the system into a set of first order differential equations, [14]. Adaptive step-size control is very useful when the local complexity of the ordinary differential equation varies and smaller or larger steps are required to meet the desired accuracy as well as maximize efficiency. The explicit Runge-Kutta methods require an initial guess for the step-size in order to start the adaptive algorithm, and several algorithms exist to aid in the selection of the starting step-size for general ordinary differential equations, [21, 40].

Multi-step, or predictor-corrector methods, use the current state and a set of backpoints (previous state values) to predict the state at some future time. The predicted state is then added to the set of backpoints and a separate corrector algorithm is used to refine the approximation of the future state. The prediction and correction phases of the algorithm are performed by linearly combining forward, backward and central differences of the ordinary differential equation evaluated at the set of backpoints until the refined estimate at some future state meets an error criterion. The 8^{th} order Gauss-Jackson (GJ8) is an example of a predictor-corrector method that has historically been the state-of-the-practice for solution of the perturbed two-body problem in celestial mechanics, [31]. In general, predictor-corrector methods use fixed step-size, however Berry presented a second-order Stormer-Cowell method with variable step-size and internal error control, [7]. A severe limitation of the predictor-corrector methods is that the initial set of backpoints must be computed using a “startup procedure” and this can be computationally expensive, especially if a restart is frequently required during the orbit propagation. For highly elliptic orbits, such as a Molniya orbit, the algorithm must either use restarts or a fixed time interval everywhere that is small enough to produce a solution that meets the user specified tolerance at perigee where the dynamics are the most nonlinear. An alternative may be to reformulate the differential equations using the Sundman transformation such that the independent variable step is true/eccentric anomaly instead of time, [8]. Typically, an iterative startup procedure is implemented whereby the backpoints are determined via a semi-analytic approximation or a lower order numerical method, [10].

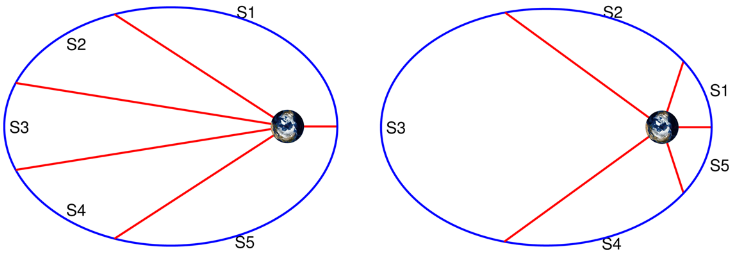
Collocation, Picard-Chebyshev and Implicit-Runge-Kutta (IRK) methods all fall under the umbrella of implicit methods. IRK methods require the solution of a set of coupled nonlinear differential equations and need an initial guess of the intermediate state values in order for the algorithm to start, [26, 42]. Collocation is a numerical method that makes use of a set of basis functions, usually orthogonal, to approximate state values subject to known boundary conditions under the constraint that the derivative of the approximation exactly satisfies the ordinary differential equation at a set of prior determined sample points, [17–19, 39]. Compared to explicit Runge-Kutta (ERK) methods, IRK methods can achieve the same order of accuracy with a fewer number of stages. Van Der Houwen et al. [38], investigated high order IRK methods with step size control. They showed that 10th order IRK is more efficient than DP8(7). Jones, [27], introduced a scheme that uses Gauss-Legendre (GL) collocation for first order systems, dubbed as VGL-s. In terms of function calls and execution time, VGL-s outperforms DP8(7), especially when high accuracy is required [1, 27]. Picard-Chebyshev methods solves the differential equations in integral form where the integrand is approximated with Chebyshev polynomials and the integration is performed iteratively using Picard iteration. In general, the computational cost in serial mode is greater for implicit methods than explicit methods, however implicit methods are parallelizable and can also be made more efficient through the use of local force approximations that are not possible with explicit integrators. In particular, the recently developed Adaptive-Picard-Chebyshev (APC) algorithm makes use of local force approximations and achieves better efficiency,

Table 1 Numerical Integrators

Method	Type	Explicit/ Implicit	Single/ Double	Adaptation	Stepping	Ref.
DP5(4)	Single-step	Explicit	Single	Adaptive	Time	[15]
DP8(7)	Single-step	Explicit	Single	Adaptive	Time	[16]
RKN12(10)	Single-step	Explicit	Double	Adaptive	Time	[14]
VGL-s	Single-step	Implicit	Single	Adaptive	Time	[27]
GJ8	Multi-step	Explicit	Double	Unadaptive	Time	[10]
APC	Path Approx.	Implicit	Double	Adaptive	True-Anomaly	[41]

without loss of accuracy, compared with explicit methods for solution of the perturbed two-body problem, [28, 41]. Unlike explicit methods, implicit methods are inherently parallelizable. Thus, they can benefit from advanced computer architectures and parallel formulation to meet future computational demands, [32]. Parallel implementations of implicit methods show over an order of magnitude speedup compared with serial implementations [2, 23, 35].

In this paper we present the results of a comprehensive study in which the precision and efficiency of six numerical integration techniques, both implicit and explicit, are compared for solving the gravitationally perturbed two-body problem in astrodynamics. As shown in Table 1, the integrators used in the comparison are the single-step DP(5)4, DP(8)7, RKN(12)10, VGL-s methods, the multistep GJ8 method and the path approximation APC method. The VGL-s and APC are implicit integrators whereas, the others are explicit methods. Some other characteristics are also shown in the table, such as single/double methods and whether or not the method is adaptive. Double integration methods, such RKN12(10), can evaluate the position from the acceleration directly; whereas single integration methods require reformulating an n second-order system of equations into a system of $2n$ first-order equations. For implicit methods, double integration decreases the number of iterations by a factor of two. Adaptive integrators select the step size to achieve a prescribed accuracy. Finally, the stepping is highlighted in either time or true-anomaly as shown in Fig. 1.



(a) Equal time segments

(b) Fixed true anomaly segments

Fig. 1 Schematic of moderately eccentric orbit divided into segments of even length in time and true anomaly, [3]

The four orbit test cases considered are, low Earth orbit (LEO), Sun-synchronous orbit (SSO), geosynchronous orbit (GEO) and a Molniya orbit (HEO). A set of tests are done using a high fidelity spherical-harmonic gravity 70 × 70 model with and without an exponential cannonball drag model. Three metrics are used for quantifying the solution precision achieved by each integration method. These are conservation of the Hamiltonian for conservative systems (no drag cases), round-trip-closure (RTC), and the method of manufactured solutions (MMS). Computational efficiency is determined by the number of function evaluations required for convergence to a solution with a prescribed tolerance. In "Integrator Overview" the selected numerical integration methods are explained and discussed. In "Integrator Error Metrics" the rationale and the development of the error evaluation techniques are presented. In "Integrator Comparison" we introduce the numerical comparison results and discuss the findings. Finally, in "Conclusion", we summarize our findings and draw conclusions.

Integrator Overview

In Cowell's formulation, the relative orbit dynamics for the perturbed two-body problem in an Earth-Centered inertial coordinate system is given by

$$\ddot{\boldsymbol{r}} = -\frac{\mu}{r^3}\boldsymbol{r} + \boldsymbol{a}_p, \tag{1}$$

where $\boldsymbol{r}$ is the geocentric position vector, $r = \sqrt{\boldsymbol{r}^{\mathrm{T}}\boldsymbol{r}}$, μ is the Earth's gravitational mass constant and $\boldsymbol{a}_p$ is the acceleration due to perturbations. For the two-body propagation, $\boldsymbol{a}_p = \boldsymbol{0}$. Letting $\boldsymbol{r} = [x, y, z]^T$, Eq. 1 represents a system of three second order ordinary differential equations (ODEs) that could be expressed in a general form as

$$\ddot{\boldsymbol{r}} = \boldsymbol{f}(t, \boldsymbol{r}, \boldsymbol{v}), \quad \boldsymbol{f}(t, \boldsymbol{r}, \boldsymbol{v}) \in \mathbb{R}^3, \tag{2}$$

with initial conditions $\boldsymbol{r}(t_0) = \boldsymbol{r}_0$ and $\boldsymbol{v}(t_0) = \boldsymbol{v}_0$. Alternatively, Eq. 2 could also be reduced to a system of first order ODEs whose general form is

$$\dot{\boldsymbol{x}} = \boldsymbol{f}\left(t, \boldsymbol{x}\right), \quad \boldsymbol{f}(t, \boldsymbol{x}) \in \mathbb{R}^6, \tag{3}$$

with the initial conditions $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$, where

$$\boldsymbol{x} = [\boldsymbol{r}; \boldsymbol{v}]. \tag{4}$$

Equations 3 and 2 are initial value problems (IVPs) of first and second order systems, respectively. General and special perturbations techniques are used to address the orbit problem. General perturbations techniques are analytic and involve the series expansion of the perturbing accelerations. They are generally low fidelity. However, their reliance on average dynamics produces physical insight into the secular, short-period, and long-period variations. Moreover, the individual contributions of gravitational and nongravitational forces could be eliminated and studied, [32]. Special perturbations, also known as numerical integration, require numerical integration of the equations of motion including all necessary perturbing accelerations, [37]. Therefore, they provide accurate solutions, especially for short-period variations.

Equation 3 may be arranged in the forward integration form as

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \int_{t_k}^{t_k+h} \boldsymbol{f}(t, \boldsymbol{x}(t))\, dt, \tag{5}$$

where $\boldsymbol{x}_k$ is the state vector at a given time t_k and $\boldsymbol{x}_{k+1}$ is the state vector at a later time t_{k+1} with timestep h. The difference between numerical integration methods is the way to approximate the integral term. Runge-Kutta methods approximate the integral term as a combination of a number of weighted evaluations of the ODE function at intermediate stages within the timestep whereas multistep methods approximate the integrand using Newton polynomials[1] exploiting the evaluations at a number of backpoints.

Alternatively, Eq. 3 can also be arranged into the path integration form as

$$\boldsymbol{x}(t) = \boldsymbol{x}(t_0) + \int_c \boldsymbol{f}(t, \boldsymbol{x}(t))\, dt, \tag{6}$$

where $\boldsymbol{x}(t)$ is the state path vector at a given time t. Picard-Chebyshev methods approximate the differential function along the trajectory using Chebyshev basis functions and iterate the "path approximation" for large segments as opposed to small timesteps.

5^{th} and 8^{th} order Dormand-Prince

Runge-Kutta methods approximate the integral term in Eq. 5 as a weighted linear combination of the ODE evaluations. The solution of a differential equation via explicit Runge-Kutta methods is given by

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + h\psi_f(t_k, \boldsymbol{x}_k, h), \tag{7}$$

where $\psi_f(t_k, \boldsymbol{x}_k, h)$ is the increment function for the ODE function f, at time t_k and states $\boldsymbol{x}_k$, within a time step h. It is evaluated according to

$$\psi_f(t_k, \boldsymbol{x}_k, h) = \sum_{i=1}^{s} b_i \boldsymbol{F}_i, \tag{8}$$

where s is the number of stages and $\boldsymbol{F}_i$ represents the ODE evaluation at stage i, and has the form

$$\begin{aligned} \boldsymbol{F}_1 &= f(t_k, \boldsymbol{x}_k), \\ \boldsymbol{F}_i &= f(t_k + c_i h, \boldsymbol{x}_k + h \sum_{j=1}^{s} a_{ij} \boldsymbol{F}_j); \quad (i = 2, ...s). \end{aligned} \tag{9}$$

For a particular Runge-Kutta method with s number of stages, the coefficients b_i (for $i = 1, 2, ..., s$), a_{ij} (for $1 \leqslant j < i \leqslant s$) and c_i (for $i = 1, 2, ..., s$) can be arranged in the Butcher Tableau as in Table 2.

[1] Given a set of $n+1$ data points $(x_0, y_0), \dots, (x_n, y_n)$, the Newton interpolating polynomial is written as $N(x) = [y_0] + [y_0, y_1](x - x_0) + \cdots + [y_0, \dots, y_n](x - x_0)(x - x_1)\cdots(x - x_{n-1})$, and $[y_0, \dots, y_n]$ denotes divided differences, e.g. $[y_0, y_1] = \frac{y_1 - y_0}{x_1 - x_0}$.

Table 2 Butcher Tableau for Explicit Runge-Kutta methods

c_1					
c_2	a_{21}				
c_3	a_{31}	a_{32}			
$\vdots$	$\vdots$	$\vdots$	$\ddots$		
c_s	$a_{s,1}$	$a_{s,2}$	$\dots$	$a_{s,s-1}$	
	b_1	b_2	$\dots$	b_{s-1}	b_s

Local truncation error represents the terms neglected by truncating the Taylor series. The accuracy of a Runge-Kutta method order (p) is comparable to that of a p^{th} order Taylor polynomial. Its error is defined by

$$e(h) = |\boldsymbol{x}(t_{k+1}) - \boldsymbol{x}(t_k) - h\psi_f(t_k, \boldsymbol{x}(t_k), h)| \leqslant C.h^{p+1}, \tag{10}$$

where C is a constant independent of h and $\boldsymbol{x}(t_k)$ and $\boldsymbol{x}(t_{k+1})$ are the exact solution at steps k and $k+1$, respectively.

Adaptive Methods

The purpose of adaptive stepsize methods is to achieve some predetermined accuracy in the solution with minimum computational effort, [34]. To fulfill this purpose, each step of integration should contribute uniformly to the total integration error, [31]. The adaptive methods are designed to control the stepsize using an estimate of the local truncation error of a single integration step. A common technique of stepsize control is the error estimate available with embedded methods.

Embedded methods produce two independent approximations

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{x}_k + h\sum_{j=1}^{s} b_i \boldsymbol{F}_j, \\ \hat{\boldsymbol{x}}_{k+1} &= \boldsymbol{x}_k + h\sum_{j=1}^{s} \hat{b}_i \boldsymbol{F}_j, \end{aligned} \tag{11}$$

of orders p and $p+1$, respectively. The local truncation error is approximated as the difference between the two approximations. For a certain step size h, the error estimate is

$$e(h) \approx |\hat{\boldsymbol{x}}_{k+1} - \boldsymbol{x}_{k+1}| / |\boldsymbol{x}_{k+1}|. \tag{12}$$

As the error $e(h)$ is proportional to h^{p+1}, the error presented for the next step h^* is

$$e(h^*) = e(h)\left(\frac{h^*}{h}\right)^{p+1}. \tag{13}$$

The predicted error on the next step should be smaller than a tolerance tol so the maximum allowable stepsize for the next step is

$$h^* = \left(\frac{tol}{e(h)}\right)^{\frac{1}{p+1}} h. \tag{14}$$

The new stepsize could also be used for the current step if the error is larger than the allowable tolerance. In practical implementation, we can reformulate Eq. 14 as [34]

$$h^* = h\beta \min\left(maxScale, \max(minScale), \left(\frac{tol}{e}\right)^{\frac{1}{p+1}}\right), \tag{15}$$

where $\beta < 1$ is a safety factor that decreases the probability of rejection of the next timestep. The values $maxScale$ and $minScale$ limit the step size increase and decrease. The current state-of-the-practice algorithms are DP5(4) and DP8(7), [15, 16].

12th/10th order Runga-Kutta

The motion of satellites is described by a system of second-order differential equations as is the dynamics of all natural dynamical systems. Instead of transforming second-order systems to first-order systems, Nystrom developed a second-order formulation of Runge-Kutta methods, [33]. For Runge-Kutta-Nystrom (RKN) methods, both the position and velocity are calculated directly from the acceleration as

$$\begin{aligned}\boldsymbol{r}_{k+1} &= \boldsymbol{r}_k + h\boldsymbol{v}_k + h^2\sum_{i=1}^{s}\bar{b}_i\boldsymbol{G}_i,\\ \boldsymbol{v}_{k+1} &= \boldsymbol{v}_k + h\sum_{i=1}^{s}b_i\boldsymbol{G}_i,\end{aligned} \tag{16}$$

with,

$$\begin{aligned}\boldsymbol{G}_1 &= g(t_k, \boldsymbol{r}_k, \boldsymbol{v}_k),\\ \boldsymbol{G}_i &= g(t_k + c_i h, \boldsymbol{r}_k + c_i h\boldsymbol{v}_k + h^2\sum_{j=1}^{i-1}\bar{a}_{ij}\boldsymbol{G}_j, \boldsymbol{v}_k + h\sum_{j=1}^{i-1}a_{ij}\boldsymbol{G}_j); \quad (i = 2, ...s)\end{aligned} \tag{17}$$

and coefficients

$$\bar{a}_{ij} = \sum_{k=1}^{s}a_{ik}a_{kj}, \quad \bar{b}_i = \sum_{j=1}^{s}b_j a_{ij}. \tag{18}$$

The adaptive RKN methods were first introduced by Fehlberg, [20]. The current state-of-the-practice RKN method is RKN12(10) which was developed by Dormand et al., [14].

Implicit Runge-Kutta Method

For Implicit Runge-Kutta (IRK) methods, the integral term in Eq. 5 is approximated as a combination of nonlinear coupled evaluations of the ODE. The s-stage implicit Runge-Kutta dubbed as the Variable-step Gauss Legendre method (VGL-s) is given in [27] as,

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + h \sum_{i=1}^{s} b_i \boldsymbol{F}_i, \tag{19}$$

where

$$\boldsymbol{F}_i = f(t_k + c_i h, \boldsymbol{x}_k + h \sum_{j=1}^{s} a_{ij} \boldsymbol{F}_j); \quad (j = 1, ..., s). \tag{20}$$

The upper limit of the summation to evaluate the function is the number of stages, s. Compared with Eq. 9, where the upper limit to evaluate the function at stage i is $i-1$, IRK has both current and future states in the algebraic expression. The coefficients for IRK schemes are obtained via collocation methods, such as Gauss-Legendre.

As shown above, implicit Runge-Kutta transforms the differential equations into a system of algebraic equations for the internal stages that are solved using Newton's method. However, using Newton's method in orbit propagation is very costly as it needs the calculation of the Jacobian. To avoid Jacobian evaluation, the fixed-point iteration is used instead.

Variable-step implicit Runge-Kutta methods use the rate of convergence for the iterative process to approximate the local truncation error. This concept was suggested by Van Der Houwen, [38]. VGL-s follows the variable-step implementation as described in "Adaptive Methods". The difference is the approximation of the local truncation error equation. It is defined for first order systems as

$$e(h) \approx \frac{|\mathbf{y}_{n+1}^{(m)} - \mathbf{y}_{n+1}^{(m-1)}|}{\mathbf{y}_{n+1}^{(m-1)}}. \tag{21}$$

If convergence is not achieved within the maximum number of iterations M, the step size is rejected and the final value of e is used to generate a new step size.

It has to be noted that, the variable step size scheme explained in the original VGL-s algorithm, [27], did not produce the best results in our tests especially for the HEO orbit. In those cases, uniform distribution in the true anomaly was adopted to produce results that are representative of the method's computational accuracy and efficiency.

8th Order Gauss-Jackson Method

Gauss-Jackson is a summed double predictor-corrector multistep method. It first appeared in the 1924 paper by Jackson, [25] where he clearly stated that the method was known to Gauss beforehand. The U.S. space surveillance centers have used an eighth-order Gauss-Jackson algorithm since the 1960s. As a double integration method, GJ8 obtains the position directly from the acceleration. Therefore, it is combined with the summed Adams method to propagate the velocity.

The GJ8 method approximates the integrand using Newton's polynomial of order $m - 1$, that interpolates m points

$$(t_{k-m+1}, \mathbf{G}_{k-m+1}), ..., (t_k, \mathbf{G}_k).$$

The predicted position and velocity are given by

$$\begin{aligned} \boldsymbol{r}_{k+1} &= h^2 \sum_{j=0}^{m+1} \delta_j \nabla^{j-2} \boldsymbol{G}_k, \\ \boldsymbol{v}_{k+1} &= h \sum_{j=0}^{m} \gamma_j \nabla^{j-1} \boldsymbol{G}_k, \end{aligned} \tag{22}$$

and they are corrected via

$$\begin{aligned} \boldsymbol{r}_{k+1} &= h^2 \sum_{j=0}^{m+1} \delta^* \nabla^{j-2} \boldsymbol{G}_{k+1}, \\ \boldsymbol{v}_{k+1} &= h \sum_{j=0}^{m} \gamma_j^* \nabla^{j-1} \boldsymbol{G}_{k+1}. \end{aligned} \tag{23}$$

The backward differences of $\boldsymbol{G}_n$ are recursively defined by

$$\begin{aligned} \nabla^0 \boldsymbol{G}_k &= \boldsymbol{G}_k, \\ \nabla \boldsymbol{G}_k &= \boldsymbol{G}_k - \boldsymbol{G}_{k-1}, \\ \nabla^k \boldsymbol{G}_k &= \nabla^{m-1} \boldsymbol{G}_k - \nabla^{m-1} \boldsymbol{G}_{k-1}. \end{aligned} \tag{24}$$

The evaluation of both $\boldsymbol{r}$ and $\boldsymbol{v}$ requires the evaluation of the first and second sums, ∇^{-1} and ∇^{-2}. They are evaluated recursively as

$$\begin{aligned} \nabla^{-1} \boldsymbol{G}_k &= \nabla^{-1} \boldsymbol{G}_{k-1} + \boldsymbol{G}_k, \\ \nabla^{-2} \boldsymbol{G}_k &= \nabla^{-2} \boldsymbol{G}_{k-1} + \nabla^{-1} \boldsymbol{G}_k. \end{aligned} \tag{25}$$

The initial values of first and second sums are calculated using

$$\begin{aligned} \nabla^{-1} \boldsymbol{G}_0 &= \frac{\boldsymbol{v}_0}{h} - \sum_{j=1}^{m} \gamma^* \nabla^{j-1} \boldsymbol{G}_0, \\ \nabla^{-2} \boldsymbol{G}_0 &= \frac{\boldsymbol{r}_0}{h^2} - \sum_{j=1}^{m+1} \delta^* \nabla^{j-2} \boldsymbol{G}_0. \end{aligned} \tag{26}$$

These coefficients could be obtained recursively from

$$\begin{aligned} \gamma_j &= 1 - \sum_{k=0}^{j-1} \frac{1}{j+1-k}\gamma_k, \\ \gamma_j^* &= -\sum_{k=0}^{j-1} \frac{1}{j+1-k}\gamma_k^*, \\ \delta_j &= (1-j)\gamma_j^*, \\ \delta_j^* &= \delta_j - \delta_{j-1}, \end{aligned} \tag{27}$$

with $\gamma_0 = \gamma_0^* = \delta_0 = \delta_0^* = 1$.

Berry and Healy, [10], presented the $PE(CE)^n$ algorithm for the Gauss-Jackson method, where n is the number of iterations used. This could be a fixed number or the number of iterations required to achieve a certain level of accuracy within each integration step.

Adaptive-Picard-Chebyshev

Picard-Chebyshev iteration is an implicit method where long trajectory arcs of the differential equation function are approximated using Chebyshev polynomials. The approximation is then integrated via Picard iteration to update the trajectory arcs until convergence is achieved, [4, 5].

Consider the following second order ordinary differential equation

$$\frac{d^2\mathbf{r}(t)}{dt^2} = \mathbf{f}(t, \mathbf{r}(t), \mathbf{v}(t)), \quad t \in [t_0, t_f], \quad \mathbf{r} \in R^{n\times 1}, \quad \mathbf{v} \in R^{n\times 1}, \quad \mathbf{f} \in R^{n\times 1}, \tag{28}$$

with initial conditions $\boldsymbol{r}(t_0) = \boldsymbol{r}_0$ and $\boldsymbol{v}(t_0) = \boldsymbol{v}_0$. This can be rearranged to obtain the corresponding integral equation for velocity:

$$\boldsymbol{v}(\tau) = \boldsymbol{v}(-1) + \int_{-1}^{s} \boldsymbol{f}(q, \boldsymbol{r}(q), \boldsymbol{v}(q))dq. \tag{29}$$

A sequence of approximate solutions $\boldsymbol{v}^i(t)$, $(i = 1, 2, 3..., \infty)$, of the true solution $\boldsymbol{v}(t)$ that satisfies this integral equation may be obtained through Picard iteration using the following set of approximate paths:

$$\boldsymbol{v}^i(\tau) = \boldsymbol{v}(-1) + \int_{-1}^{s} \boldsymbol{f}(q, \boldsymbol{r}^{i-1}(q), \boldsymbol{v}^{i-1}(q))dq. \tag{30}$$

Picard proved that for smooth, differentiable, single-valued nonlinear functions $\boldsymbol{f}(t, \boldsymbol{r}(t), \boldsymbol{v}(t))$, there is a time interval $\left|t_f - t_0\right| < \delta$ and a starting trajectory $\boldsymbol{v}^0(t)$ satisfying $\left\|\boldsymbol{v}^0(t) - \boldsymbol{v}(t)\right\|_\infty < \Delta$, that 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.

The velocity on the left hand side of Eq. 29 can also be written in terms of a Chebyshev series where $\boldsymbol{\beta}$ represents the the velocity coefficients and $\boldsymbol{\alpha}$ represents the coefficients of the acceleration that are computed through least squares. $p = \frac{t_f - t_0}{2}$

is a scale factor that transforms time to a new independent variable τ that exists on the domain from $[-1, 1]$. Note that the upper limit on the summation for the least squares coefficients is $N-2$ for this second order system. For first order systems the upper limit of the summation is $N - 1$.

$$\boldsymbol{v}^i(\tau) = \sum_{k=0}^{N-1} \boldsymbol{\beta}_k^i T_k(s) = \boldsymbol{v}(-1) + p \int_{-1}^{s} \sum_{k=0}^{N-2} \boldsymbol{a}_k^{i-1} T_k(q) dq, \tag{31}$$

The position coefficients ($\boldsymbol{\alpha}$) are computed directly from the velocity coefficients ($\boldsymbol{\beta}$) as follows:

$$\boldsymbol{r}^i(\tau) = \sum_{k=0}^{N} \boldsymbol{\alpha}_k^i T_k(\tau) = \boldsymbol{r}(-1) + p \int_{-1}^{\tau} \sum_{k=0}^{N-1} \boldsymbol{\beta}_k^i T_k(s) ds. \tag{32}$$

Junkins and Woollands, [28, 41], provide a full derivation of APC as well as an explanation of the adaptive segmentation scheme, the accelerated error feedback (quasilinearization), and the radially adaptive and variable fidelity gravity models, [29].

Integrator Error Metrics

In this section we present three methods for quantifying the accuracy of a numerical integration method. These are *conservation of the Hamiltonian*, *round-trip-closure* (RTC), and *method of manufactured solutions* (MMS). There are advantages and disadvantages to each method, neither is the perfect way to measure integrator error. Additionally, there exists other qualified methods, e.g. step-size halving, [7, 9]. However, studying the performance of each integration method using these three error metrics provides valuable insight that can aid in answering the question: *Which integrator should be used to solve a given problem, considering some application specific accuracy and efficiency requirements?*

Conservation of the Hamiltonian

The conservative nature of the the Earth's potential acceleration means that the total energy of an object in orbit must be constant for all time. Computing the variation in the Hamiltonian at various locations along the propagated trajectory can be used as a metric to quantify the precision of the numerical integration method. The main drawback of this method is that it can only be used for conservative systems (no atmospheric drag), and the Hamiltonian is also blind to in-track errors. This is most pronounced for circular orbits, and we have found that the Hamiltonian check is usually an order of magnitude more optimistic than reality because all points along a circular orbit obviously have the same constant Hamiltonian.

For an Earth-Centered-Earth-Fixed (ECEF) system[2] the Hamiltonian (H) is calculated as

$$H = \frac{1}{2}\boldsymbol{v}_{ECEF}.\boldsymbol{v}_{ECEF} - \frac{1}{2}\omega^2(\boldsymbol{r}^2_{xECEF} + \boldsymbol{r}^2_{yECEF}) + U(\boldsymbol{r}_{ECEF}) \tag{33}$$

where $\boldsymbol{r}_{ECEF}$ and $\boldsymbol{v}_{ECEF}$ are the position and velocity vectors in the rotating frame, U is the Earth's gravitational potential field and ω is the earth angular velocity. The precision (ϵ) is computed as

$$\epsilon = -\log\left(\frac{max(|H_k - H_0|)}{H_0}\right), \quad k = 1, .., N_s, \tag{34}$$

where N_s is the number integration steps (or number of nodes for implicit integrators) and H_0 and H_k are the Hamiltonian at the initial time and time k, respectively.

Round-Trip-Closure

RTC is an integrator error metric that measures the accumulative error which results during numerical integration. Consider the nonlinear differential equation, with specified initial conditions:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t)), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \quad t_f \le t \le t_0. \tag{35}$$

Suppose that the differential equation of Eq. 35 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.

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. 36.

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t)), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}(t_f) = \boldsymbol{x}_f, \quad t_f \ge t \ge t_0. \tag{36}$$

The new initial conditions are propagated backwards in time along the trajectory in order to recover the initial conditions used for the forward integration. Note that for the orbit problem the Earth is rotated backwards in time. The state $\{\boldsymbol{r}_f(t_0), \boldsymbol{v}_f(t_0)\}$ is the final state at time t_0 on the backward integration from time t_f. The error metric is evaluated as follows:

$$J = \frac{1}{2}\left(\left(\frac{|\boldsymbol{r}_0(t_0) - \boldsymbol{r}_f(t_0)|}{|\boldsymbol{r}_0|}\right) + \left(\frac{|\boldsymbol{v}_0(t_0) - \boldsymbol{v}_f(t_0)|}{|\boldsymbol{v}_0|}\right)\right). \tag{37}$$

For Implicit methods (APC and VGL-s), slightly varying the node locations (increasing the number of nodes is recommended) along the reverse trajectory allows the solution to be computed using different points in the 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. Berry et al. [9], mention this as being a disadvantage when testing step integrators in a perturbed

[2] Generally, the gravitational potential is defined in ECEF coordinate system wherein the Earth rotation is accounted for.

environment. They comment that it does not measure any reversible integration error as it will be canceled 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, [22]. 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.

One subtlety we have noticed is that each numerical integrator, when applied to a particular differential equation has small RTC errors that are quasi-periodic along the trajectory, and even though the amplitude may be small, drawing a “tight” accuracy conclusion is difficult in the presence of these oscillations. Comparing the forward and reverse integrations with each other to form a relative state error permits us to infer the amplitude and frequency content of these relative errors. This in turn suggests that a small ensemble of final states associated with the final times spanning the typically small period of the domain error oscillation should be used. The maximum error associated with the worst RTC closure error has been found to be a conservative estimate of the solution accuracy.

Method of Manufactured Solutions

MMS, [30, 43–45], computes, or manufactures, an analytical function near 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 system has an analytical solution, which when compared to the numerical solution allows the numerical accuracy of the integrator to be measured.

Consider the nonlinear differential equation, with specified initial conditions:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t)), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \quad t_0 \le t \le t_f. \tag{38}$$

Suppose that the differential equation of Eq. 38 does not have an analytical solution. Furthermore, suppose that an approximate solution $\boldsymbol{x}_r(t)$ is available that does not satisfy Eq. 38 exactly but is believed to satisfy it with “small” but unknown errors. Suppose that $\boldsymbol{x}_r(t)$ is smooth and at least once differentiable. In some methods, $\boldsymbol{x}_r(t)$ is directly available as an algebraic linear combination of differentiable basis functions, in other *step-by-step* methods, a finite sequence of states at known times must be interpolated to produce $\boldsymbol{x}_r(t)$. On substituting $\boldsymbol{x}_r(t)$ into Eq. 38, we can obtain an explicit algebraic solution for the error as

$$\boldsymbol{d}_r(t) = \dot{\boldsymbol{x}}_r(t) - \boldsymbol{f}(t, \boldsymbol{x}_r(t)) \tag{39}$$

or

$$\dot{\boldsymbol{x}}_r(t) = \boldsymbol{f}(t, \boldsymbol{x}_r(t)) + \boldsymbol{d}_r(t). \tag{40}$$

We can compute the norm of $\boldsymbol{d}_r(t)$ to see if it is sufficiently small to consider a good starting approximation. As a rule of thumb for orbit problems, if double precision accuracy is sought for the state trajectory, then at least single precision accuracy is desired for $\boldsymbol{x}_r(t)$, and we would typically require $\|\boldsymbol{d}_r(t)\|/\|\boldsymbol{f}(t,\boldsymbol{x}_r(t))\|$ to be smaller than 10^{-7}. Frequently, we find when testing high fidelity methods that our starting solution is more accurate with much smaller $\boldsymbol{d}_r(t)$. Comparing Eqs. 38 and 40 and reflecting for a moment, it is clear that $\boldsymbol{x}_r(t)$ *is the exact analytical solution* of the slightly disturbed differential equation

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(t, \boldsymbol{x}(t)) + \boldsymbol{d}_r(t), \quad \boldsymbol{x}(t_0) = \boldsymbol{x}_0, \quad t_0 \le t \le t_f. \tag{41}$$

Since we have a candidate solution with a small $\|\boldsymbol{d}_r(t)\|$, Eq. 41 can be considered a very close neighboring problem to the original one of Eq. 38, but with the important advantage that *we know the exact analytical solution* $\boldsymbol{x}_r(t)$. One can argue that whatever numerical method is under evaluation for solving Eq. 38, can be evaluated on the perturbed system of Eq. 41, which should prove slightly more difficult for the numerical solver than solving the original unforced system of Eq. 38. Any numerical integration process of interest can be used to solve the perturbed system of Eq. 41 and obtain an approximate solution $\tilde{\boldsymbol{x}}(t)$, however we know the exact solution of Eq. 41 is $\boldsymbol{x}_r(t)$, so we can compute the exact solution error $e(t) = \tilde{\boldsymbol{x}}(t) - \boldsymbol{x}_r(t)$ of this perturbed problem at any/all times. If the numerical method of interest in Eq. 38 gives, for example, a 15 digit solution *for the more difficult perturbed problem* of Eq. 41, then we can be justifiably optimistic that it will solve Eq. 38 with similar precision. In particular, if $\frac{\|\boldsymbol{d}_r(t)\|}{\|\boldsymbol{f}(t,\boldsymbol{x}_r(t))\|} < \varepsilon$, say 10^{-7}, then our experience indicates that the numerical method used (with the same tuning used to generate $\tilde{\boldsymbol{x}}(t)$), when applied to Eq. 38, will always (at least in extensive tests with various initial states for several nonlinear systems that have known analytical solutions) produce a solution with 14 significant figures.

Traditionally, the main weakness with the MMS test is that the acceleration, $\ddot{\boldsymbol{x}}_r(t)$, is 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-by-step* integrators, but for APC the coefficients of the acceleration fit are already available due to the *path approximation* nature of the algorithm, so no differentiation of the velocity approximation is required. To avoid differentiation of the velocity approximation for the *step-by-step* integrators, in this paper we pre-compute a reference trajectory (and corresponding coefficients) using APC. During the MMS iterations the *step-by-step* integrators interpolate the reference trajectory (position, velocity, acceleration) at every time step in order to generate MMS acceleration. Thus no differentiation of the state trajectory approximation of the velocity is necessary and MMS can be used to honestly test the accuracy of the integrators without the necessity of introducing other approximations.

Integrator Comparison

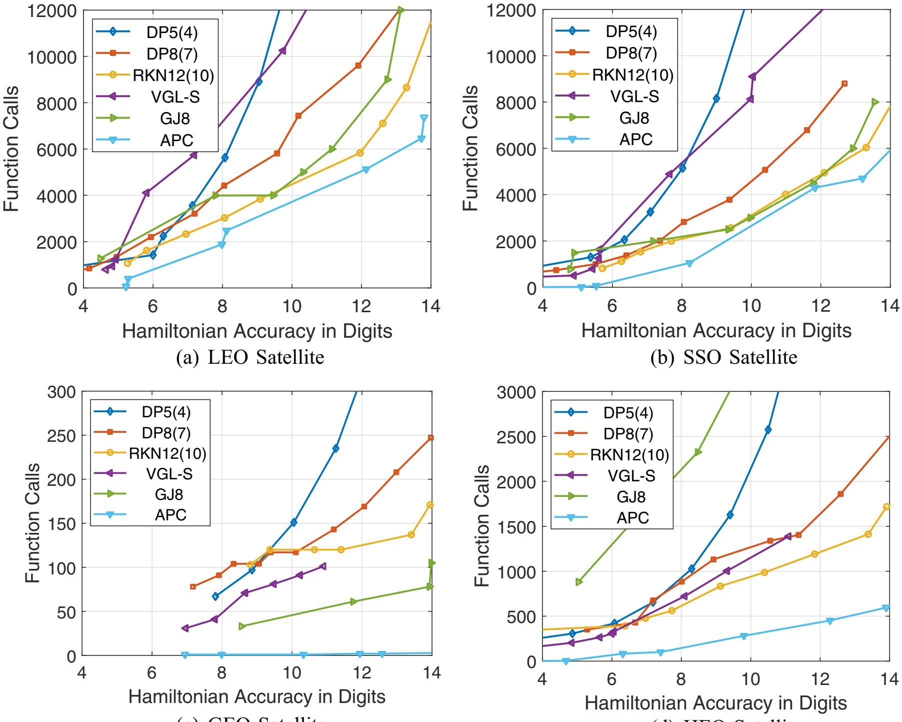
In this section we discuss the results of the integrator comparison for both conservative and non-conservative systems using the four example test cases: LEO, SSO, GEO and Molniya (HEO). Table 3 outline their orbital elements. The methods for determining the integrator accuracy are *conservation of the Hamiltonian*,

Table 3 Study satellites

Category	Semimajor Axis (km)	Eccentricity	Inclination (degree)	Right Ascension of Ascending Node (degree)	Argument of Perigee (deg)	True Anomaly (degree)
LEO	6745.592	0.01	7.81	100.21	152.83	0.0
SSO	7153.12	0.0	98.4469	212.741	0.0	0.0
GEO	42,164.0	0.001	0.0	0.0	0.0	0.0
HEO	26,164	.74	63.4	0.0	0.0	0.0

round-trip-closure and *method of manufactured solutions*. The spherical harmonics gravity model is given by, [36]:

$$V\left(r, \phi, \lambda\right) = \frac{\mu}{r}\left[1 + \sum_{n=2}^{\infty}\sum_{m=0}^{n}\left(\frac{R_{eq}}{r}\right)^{n} P_{nm}\sin\phi\left(C_{nm}\cos(m\lambda) + S_{nm}\sin(m\lambda)\right)\right], \qquad (42)$$



(a) LEO Satellite

(b) SSO Satellite

(c) GEO Satellite

(d) HEO Satellite

Fig. 2 Work-precision diagram for conservation of the Hamiltonian with spherical harmonic gravity perturbations (70 × 70 degree and order) propagated over 12 hours

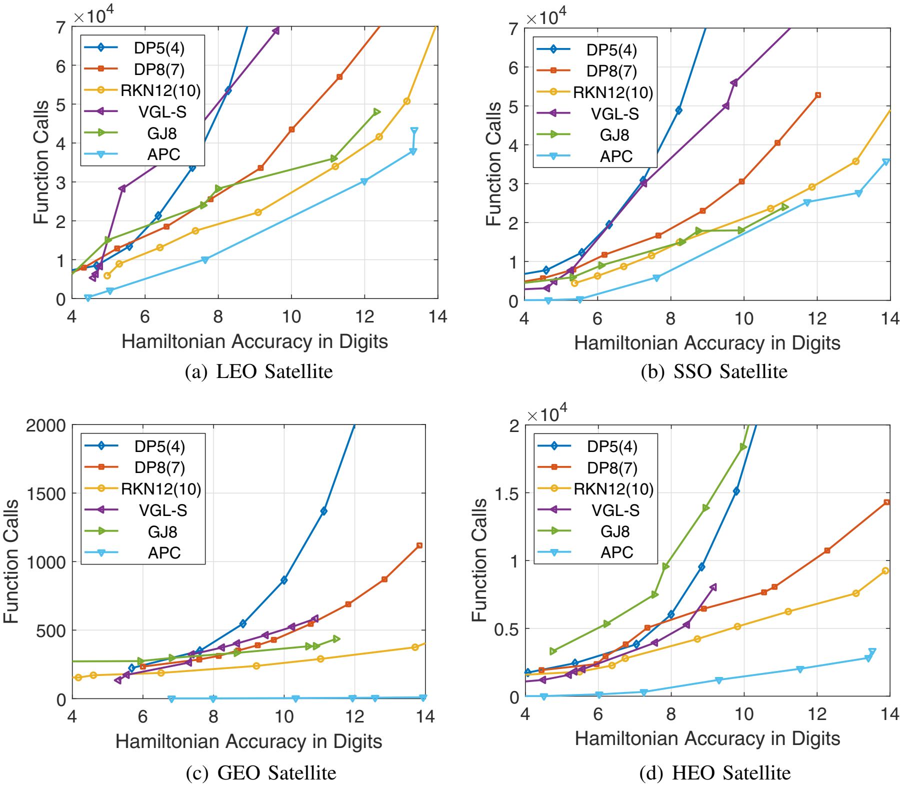
where $\{r, \phi, \lambda\}$ are the spherical coordinates of the satellite with r describing the distance, λ the longitude , ϕ the latitude, the functions $P_{nm}(x)$ are the associated Legendre polynomials, C_{nm} and S_{nm} are constants determined from experimental data following Earth Gravitational Model 2008 (EGM 2008). The inertial acceleration can then be obtained from the gradient of the potential given in Eq. 42 as,

$$\mathbf{a} = \nabla V = \frac{\partial V}{\partial r}\nabla r + \frac{\partial V}{\partial \phi}\nabla \phi + \frac{\partial V}{\partial \lambda}\nabla \lambda \tag{43}$$

The perturbing acceleration due to the cannonball drag model is given by, [37]:

$$\mathbf{a} = -\beta\rho|\mathbf{v}_{rel}|\mathbf{v}_{rel}, \tag{44}$$

where, the relative velocity, $\mathbf{v}_{rel} = \mathbf{v} - \boldsymbol{\omega} \times \mathbf{r}$, $\boldsymbol{\omega}$ is Earth's angular velocity, the density, $\rho = \rho_0 \exp\left[(h - h_0)/H\right]$, h is the height above Earth's surface, h_0 is a reference



(a) LEO Satellite

(b) SSO Satellite

(c) GEO Satellite

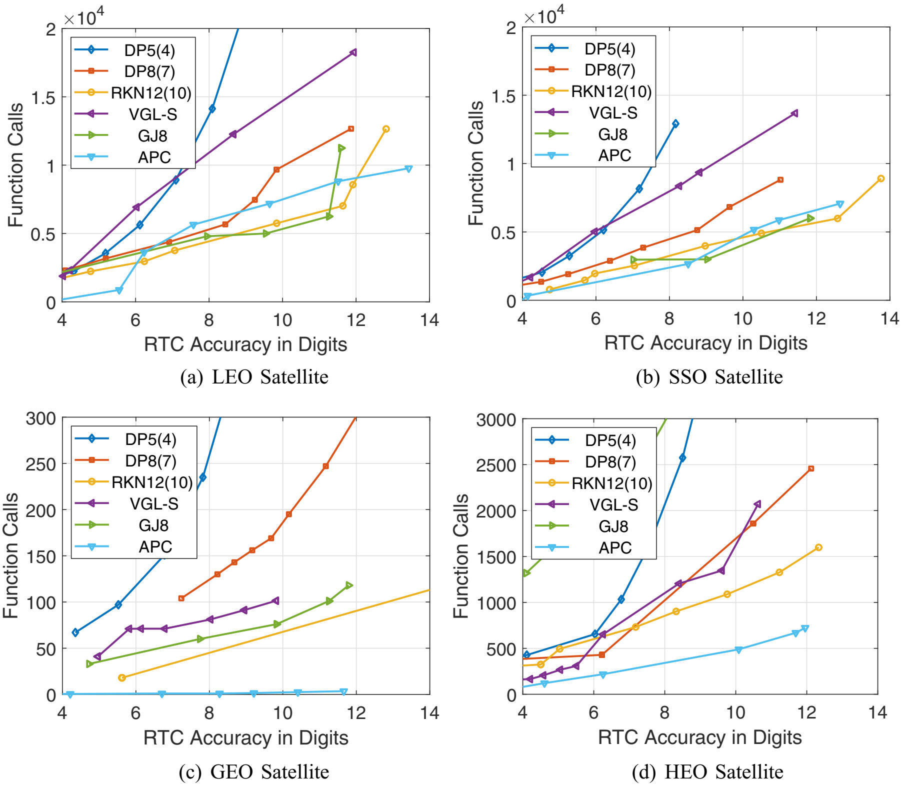
(d) HEO Satellite

Fig. 3 Work-precision diagram for conservation of the Hamiltonian with spherical harmonic gravity perturbations (70 × 70 degree and order) propagated over 72 hours

height, H is the scale height, and β is the ballistic coefficient, $\beta = C_d A/m$, where C_d is the drag coefficient, A is the drag area and m is the mass of the satellite.

Conservative Cases: no Atmospheric Drag

Figure 2 shows the Hamiltonian results for the four test cases where the propagation was done with a 70 × 70 degree and order spherical harmonic gravity model over 12 hours. In all cases, APC exhibits the best performance, that is the least number of function evaluations required to achieve a specified accuracy. Not surprisingly, the low order integrator, DP5(4), struggles to produce trajectories that are accurate to 12 digits. GJ8 performs reasonably well except for the highly eccentric HEO case where the fixed step-size leads to many unnecessary function evaluations over apogee where the gravity gradients are the least nonlinear. The high order integrators (DP8(7), RKN12(10), APC) are able to achieve solutions that are accurate to 12-14 digits for



(a) LEO Satellite

(b) SSO Satellite

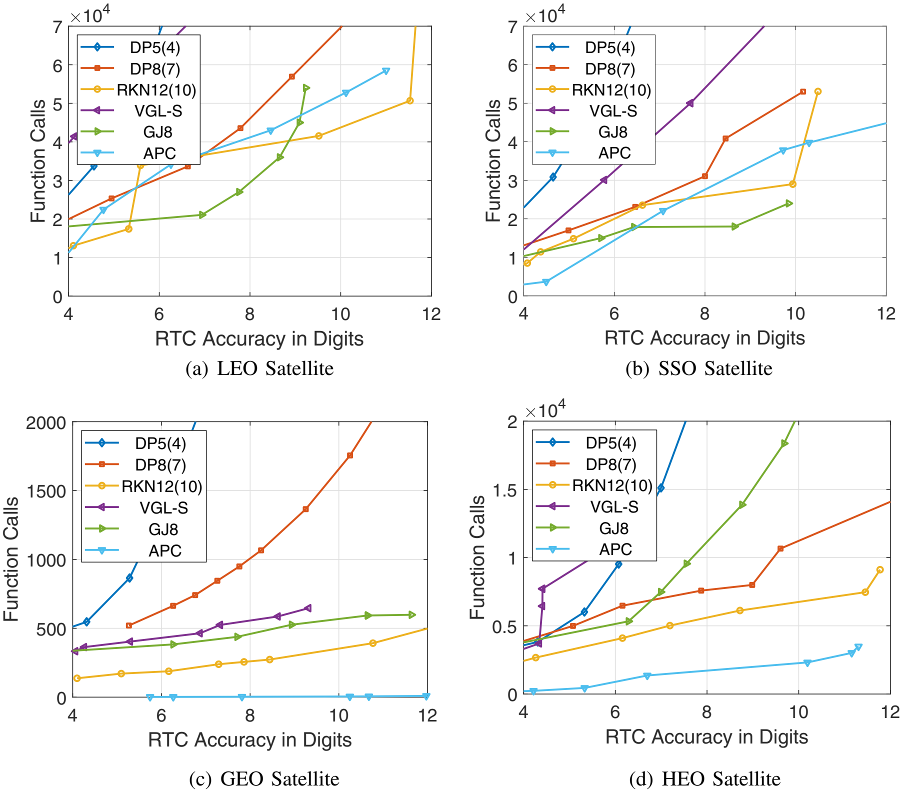
(c) GEO Satellite

(d) HEO Satellite

Fig. 4 Work-precision diagram for RTC with spherical harmonic gravity perturbations (70 × 70 degree and order) propagated over 12 hours

all the test cases. VGL-s performed very similarly to DP8(7) for the GEO and HEO cases. For the 72 hours propagation, Fig. 3, the number of function calls increased by an order of magnitude across all methods. APC, RKN12(10) and GJ8 had similar performance in the LEO, SSO and GEO cases. GJ8, as in the 12 hours test, struggled in the HEO case. Overall, APC maintained higher efficiency across all cases. VGL-s and DP8(7) had very similar results in GEO and HEO; however, in the LEO and the SSO cases DP8(7) outperformed VGL-s. The low order integrator, DP5(4), produced even less accurate solutions. This is not surprising as round-off and truncation errors accumulate over a longer propagation time.

Figure 4 shows the RTC results for the four test cases where the propagation was done considering a 70 × 70 degree and order spherical harmonic gravity model over 12 hours. As with the Hamiltonian, the higher order integrators performed the best. The most notable difference is that for the LEO and HEO cases the integrators do not achieve the 14 digit accuracy that was portrayed in Fig. 2 with the Hamiltonian error metric. For the 72 hours propagation shown in Fig. 5 the trends are similar



(a) LEO Satellite

(b) SSO Satellite

(c) GEO Satellite

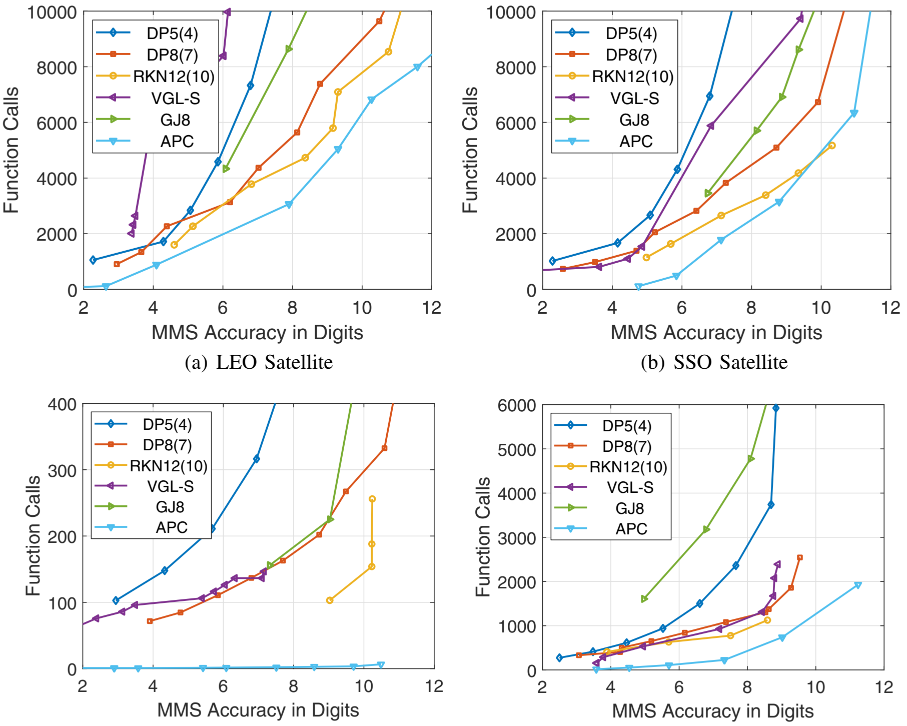
(d) HEO Satellite

Fig. 5 Work-precision diagram for RTC with spherical harmonic gravity perturbations (70 × 70 degree and order) propagated over 72 hours

to the 12 hours propagation case, however it is clear that the absolute achievable accuracy is reduced by at least an order of magnitude. This is again consistent with the fact that numerical error accumulates during the orbit propagation and highlights the optimistic nature of the Hamiltonian error metric.

Figure 6 shows the MMS results for the four test cases where the propagation was done considering a 70 × 70 degree and order spherical harmonic gravity model over 12 hours. As with the Hamiltonian and RTC tests, the higher order integrators show better performance. We see that for all test cases the achievable accuracy is considerably less than that demonstrated by the Hamiltonian and RTC performance metrics for all integrators. For the 72 hours propagation, Fig. 7, similar trends are evident but with an overall reduction in the final achievable tolerance.

It is important to note that neither of these tests is perfect and one should evaluate the performance of an integrator by taking into account the results from all three



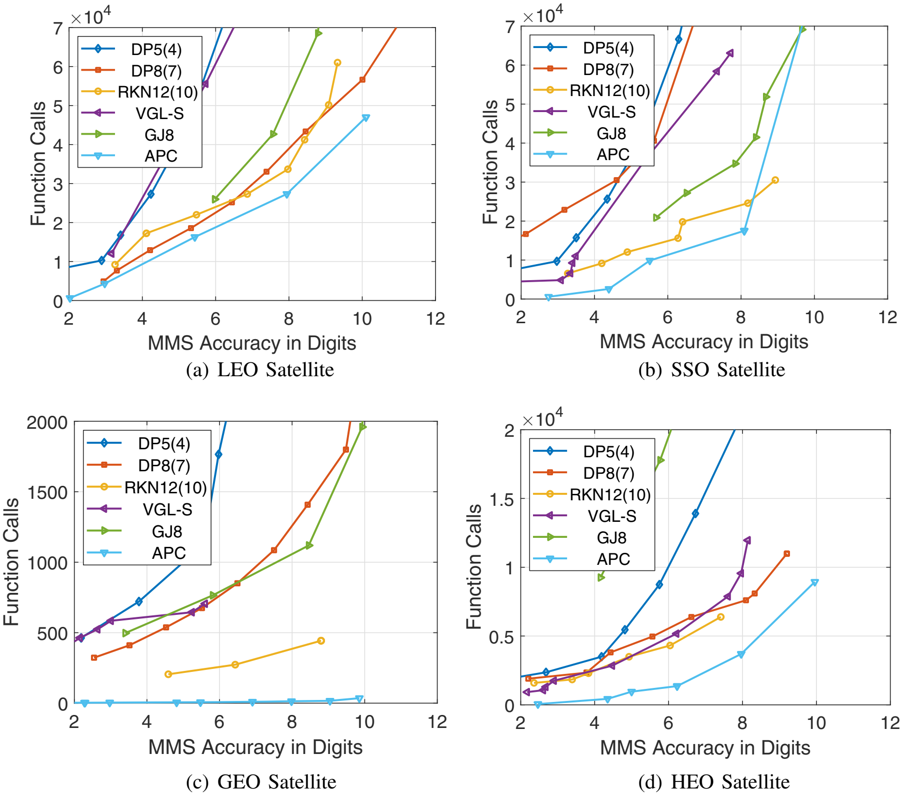
(a) LEO Satellite

(b) SSO Satellite

(c) GEO Satellite

(d) HEO Satellite

Fig. 6 Work-precision diagram for MMS with spherical harmonic gravity perturbations (70 × 70 degree and order) propagated over 12 hours



(a) LEO Satellite (b) SSO Satellite

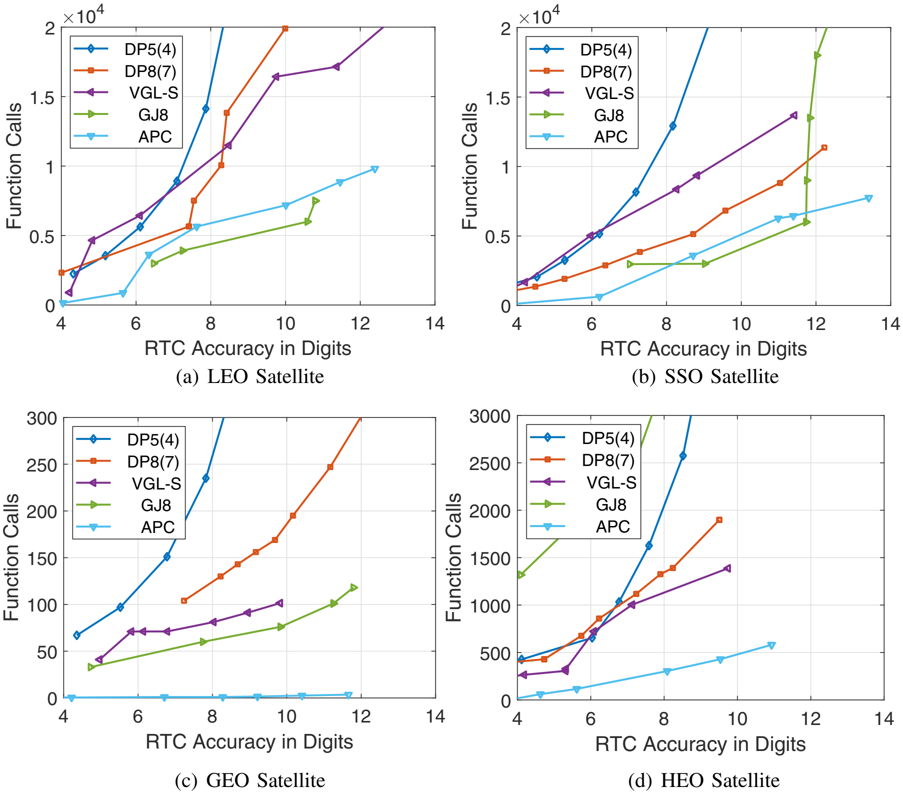
(c) GEO Satellite (d) HEO Satellite

Fig. 7 Work-precision diagram for MMS with spherical harmonic gravity perturbations (70 × 70 degree and order) propagated over 72 hours

tests. One obvious conclusion is that APC performs better than the other integrators for almost all the test case orbits.

Nonconservative Cases: Exponential Drag Model

In this section we include the exponential atmospheric drag model, Eq. 44, along with the 70 × 70 degree and order spherical harmonic gravity model in the orbit propagation. We assume that the drag coefficient is $C_d = 2.2$, the surface area is $A = 1\ \text{m}^2$ and the mass of the spacecraft is $m = 25$ kg. Note that conservation of the Hamiltonian is only valid for conservative systems, and thus only the RTC and MMS error metrics are used for examining the integrator accuracy for these non-conservative test cases. Also, the RKN(12)10 algorithm, in its original form, is



(a) LEO Satellite

(b) SSO Satellite

(c) GEO Satellite

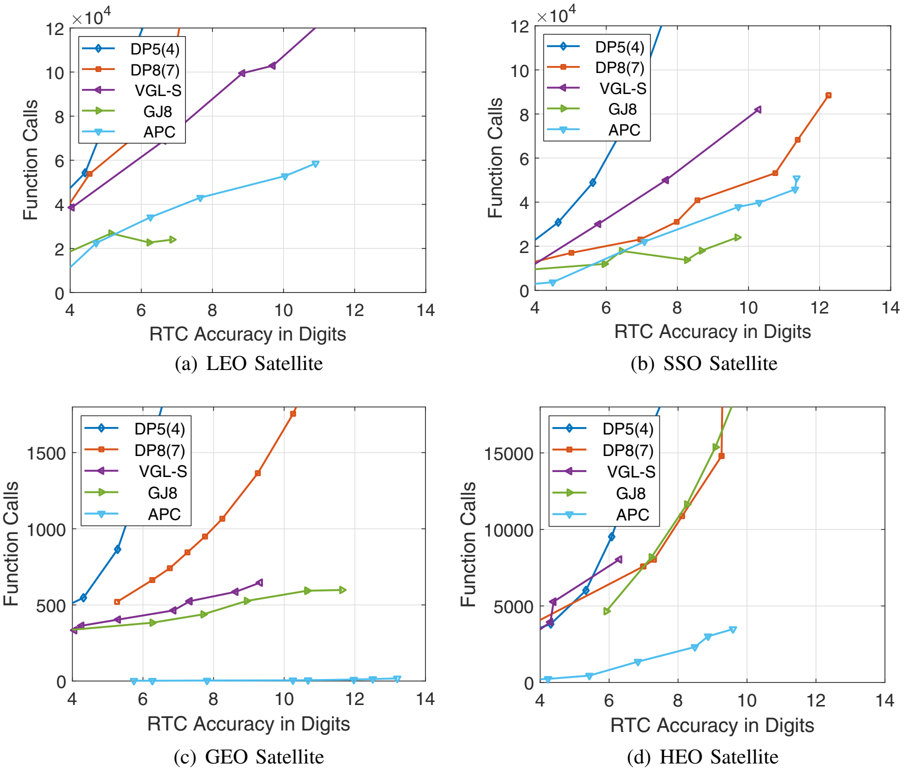
(d) HEO Satellite

Fig. 8 Work-precision diagram for RTC, with spherical harmonic gravity perturbations (70 × 70 degree and order) and an exponential drag model, propagated over 12 hours

limited to integrating second order systems that are a function of position only. The drag force model is velocity dependent and thus the combined force model (drag plus gravity) is a function of both position and velocity, thus preventing RKN(12)10 from being used in the comparison.[3]

Figures 8 and 9 show the RTC results for the four test case orbits for 12 and 72 hours, respectively. For the LEO, SSO and GEO test cases, APC and GJ8 outperformed the other methods for both propagation times. In the HEO 12 hours propagation, VGL-s and DP8(7) performed better than GJ8; however, for the 72

[3] RKN(12)10 was not tested with drag due to software limitations.



(a) LEO Satellite

(b) SSO Satellite

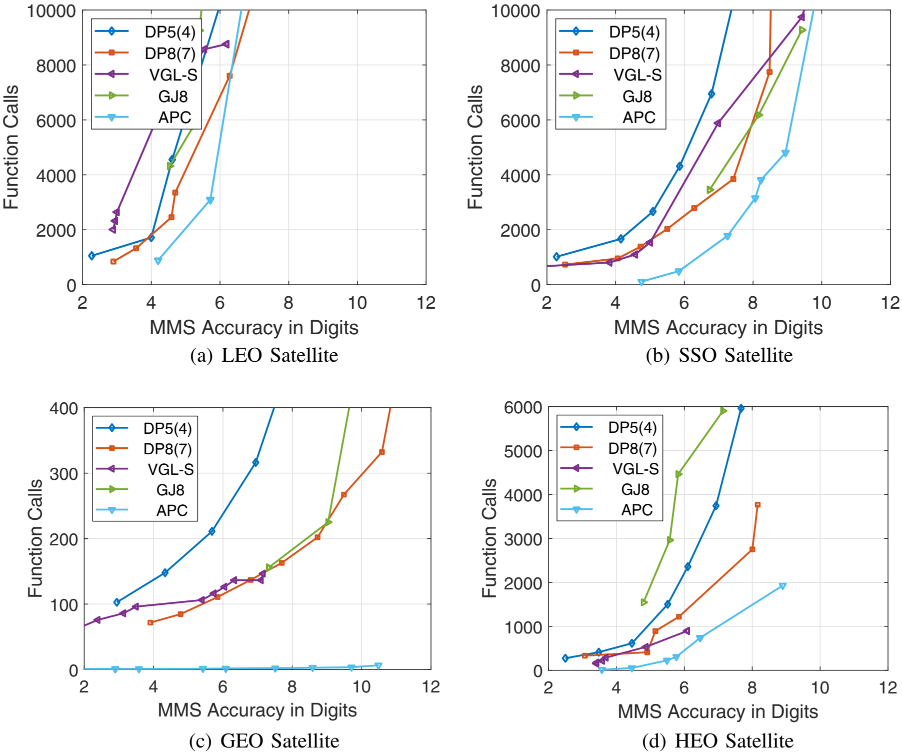
(c) GEO Satellite

(d) HEO Satellite

Fig. 9 Work-precision diagram for RTC, with spherical harmonic gravity perturbations (70 × 70 degree and order) and an exponential drag model, propagated over 72 hours

hours case APC outperformed all methods by approximately an oder of magnitude infunction calls for 10 digits of accuracy. In particular, the RTC results demonstrate that the integrators perform at least two orders of magnitude worse then they did for the no-drag simulations. This is an important point for consideration when making an integrator selection.

Figures 10 and 11 show the MMS results for the four test case for 12 and 72 hours, respectively. APC generally outperforms the other methods. When compared to RTC all methods produced similar results. However, GJ8 produced much worse results in the MMS test when compared to RTC. This is due to the propagation of back-points that affected the MMS results. For the 72 hours propagation, and except for APC in the GEO case, the achievable accuracy was reduced by an order of magnitude as a result of the longer intergation period.



(a) LEO Satellite

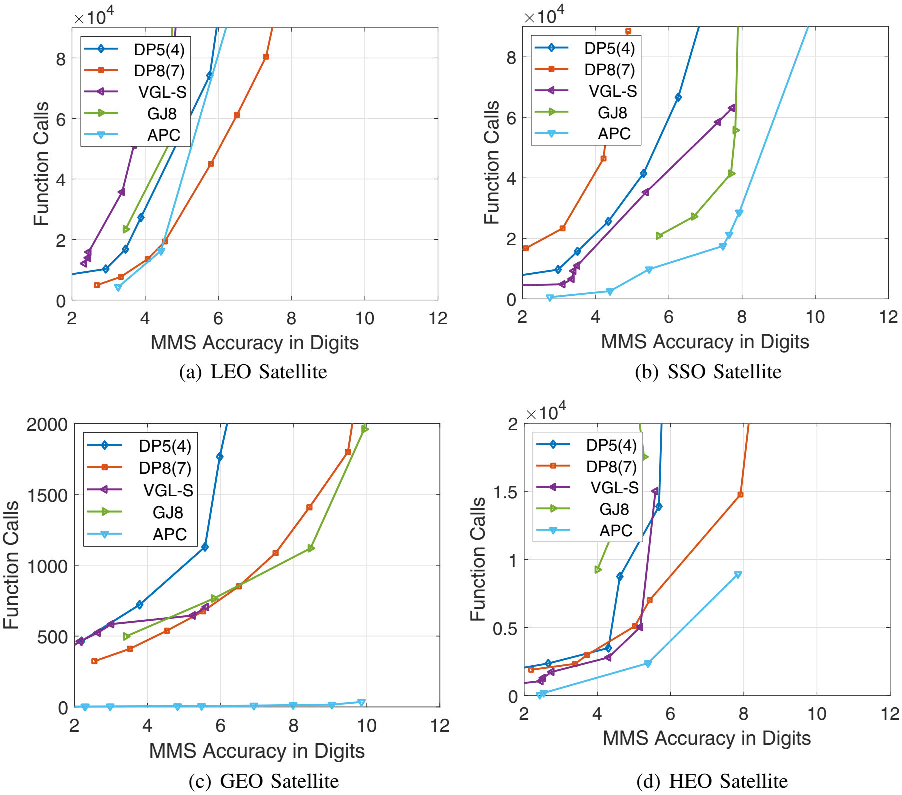
(b) SSO Satellite

(c) GEO Satellite

(d) HEO Satellite

Fig. 10 Work-precision diagram for MMS, with spherical harmonic gravity perturbations (70 × 70 degree and order) and an exponential drag model, propagated over 12 hours

The RTC errors depend on the numerical integrators and dynamical system under consideration: there is no known way to reduce the errors, beyond step size selection and similar integrator tuning, unless the integrators themselves are modified. Our testing shows that the space-time spectral content of the acceleration experienced along the particular trajectory interplays with the particular integrator to produce errors that oscillate in a difficult to predict manner. Perhaps, one positive feature of the RTC method is that it reveals these error oscillations, but heretofore, there has not been a method for dealing with these oscillations. Our recommended approach is to compute an ensemble of RTC errors, that flow an ensemble of final states at final times beyond the declared final time of the forward integration (and spanning the longest period evident in the oscillations observed in trial integrations), and then use the largest norm of the resulting ensemble of round trip closure errors as the conservative RTC measure of convergence.



(a) LEO Satellite

(b) SSO Satellite

(c) GEO Satellite

(d) HEO Satellite

Fig. 11 Work-precision diagram for MMS, with spherical harmonic gravity perturbations (70 × 70 degree and order) and an exponential drag model, propagated over 72 hours

Conclusion

In this paper we present the results of a comprehensive study in which the precision and efficiency of six numerical integration techniques, both implicit and explicit, are compared for solving the perturbed two-body problem in astrodynamics. The integrators used in the study are $5^{th}/4^{th}$ and $8^{th}/7^{th}$ order Dormand-Prince, 8^{th} order Gauss-Jackson, $12^{th}/10^{th}$ order Runga-Kutta-Nystrom, Variable-step Gauss Legendre integrator, and the Adaptive-Picard-Chebyshev methods. Four orbit test cases were considered, low Earth orbit, Sun-synchronous orbit, geosynchronous orbit and a Molniya orbit. A set of tests were done using a high fidelity spherical-harmonic gravity (70 × 70) model with and without an exponential cannonball drag model. We present three metrics for quantifying the solution precision achieved by each integration method. These are round-trip-closure (RTC), the method of manufactured solutions (MMS), and conservation of the Hamiltonian for conservative systems. The efficiency of each integrator is determined by the number of function evaluations required for convergence to a solution. We found that the higher order integrator, Adaptive Picard-Chebyshev, outperformed the lower order integrators with regard to both accuracy and efficiency for all the orbit test cases, except for the LEO cases, where it is competitive with several other excellent algorithms, especially Gauss-Jackson 8th order and RKN12(10). The 72 hour integrations, for MMS and all 4 test orbits showed the Adaptive Picard-Chebyshev method to be very stable and the most efficient for a given accuracy. We also confirmed that the Hamiltonian frequently gives an overly optimistic accuracy measure compared with round-trip-closure and the method of manufactured solutions. Based on the shown results, we highly recommend the use of all three error metrics when validating a new algorithm or when comparing the relative merits of several algorithms. All the results from our study are concisely presented in several figures that are intended to provide the reader with useful information for selecting integrators for their purposes and to illustrate a systematic way to evaluate integrators in astrodynamics.

Acknowledgements The authors would like to thank Abhay Masher and Austin Probe for their help and technical advice. Additionally, we would like to thank Dr. Julie Moses, Dr. Stacie Williams (AFOSR) and Dr. Alok Das (AFRL) for their financial support. Finally, we would like thank the Egyptian Cultural and Educational Bureau in Washington, D.C. for supporting the first author's visit to Texas A&M University in Summer 2016.

Disclaimer This work was done as a private venture and not in Woollands' capacity as an employee of the Jet Propulsion Laboratory, California Institute of Technology.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Atallah, A.: An Implementation and Verification of Of a High-Precision Orbit Propagator. Master's Thesis, Cairo University, Cairo (2018)
2. Atallah, A., Bani Younes, A.: Parallel integration of perturbed orbital motion. In: AIAA Scitech 2019 (2019)
3. Atallah, A., Woollands, R.M., Bani Younes, A., Junkins, J.: Tuning orthogonal polynomial degree and segment interval length to achieve prescribed precision approximation of irregular functions. In: 2018 Space Flight Mechanics Meeting, pp 2225. https://doi.org/10.2514/6.2018-2225 (2018)
4. Bai, X.: Modified Chebyshev-Picard Iteration for Solution of Initial Value and Boundary Value Problems. Ph.D. Thesis, Texas A&M, College Station (2010)
5. Bai, X., Junkins, J.: Modified chebyshev-Picard iteration methods for solution of initial value problems. Adv. Astronaut. Sci. **139**, 345–362 (2011)
6. Bani-Younes, A.: Orthogal Polynomial Approximation in Higher Dimensions: Applications in Astrodynamics. Ph.D. Thesis, Texas A&M University, College Station (2013)
7. Berry, M.: A Variable-step Double-integration Multi-step Integrator. Ph.D. Thesis, Virginia Polytechnic Institute and State University, Blacksburg (2004)
8. Berry, M., Healy, L.: The generalized sundman transformation for propagation of high-eccentricity elliptical orbits. Adv. Astronaut. Sci. **112**, 127–146 (2002)
9. Berry, M., Healy, L.: Comparison of accuracy assessment techniques for numerical integration. In: 13th AAS/AIAA Space Flight Mechanics Meeting (2003)

10. Berry, M.M., Healy, L.M.: Implementation of gauss-Jackson integration for orbit propagation. J. Astronaut. Sci. **52**(3), 331–357 (2004)
11. Butcher, J.C.: Coefficients for the study of Runge-Kutta integration processes. J. Aust. Math. Soc. **3**(2), 185–201 (1963). https://doi.org/10.1017/S1446788700027932
12. Butcher, J.C.: Implicit Runge-Kutta processes. Math. Comput. **18**(85), 50–64 (1964)
13. Butcher, J.C.: Numerical Methods for Ordinary Differential Equations. Wiley, Hoboken (2008)
14. Dormand, J., Elmikkawy, M., Prince, P.: High-order embedded Runge-Kutta-Nystrom formulae. IMA J. Numer. Anal. **7**, 423–430 (1987)
15. Dormand, J., Prince, P.: A family of embedded Runge-Kutta formulae. J. Comput. Appl. Math. **6**(1), 19–26 (1980)
16. Dormand, J., Prince, P.: A reconsideration of some embedded Runge-Kutta formulae. J. Comput. Appl. Math. **15**(2), 203–211 (1986)
17. Elgohary, T.A.: Novel Computational and Analytic Techniques for Nonlinear Systems Applied to Structural and Celestial Mechanics. Ph.D. thesis, Texas a&m University, College Station (2015)
18. Elgohary, T.A., Dong, L., Junkins, J.L., Atluri, S.N.: A simple, fast, and accurate time-integrator for strongly nonlinear dynamical systems. CMES: Comput. Model. Eng. Sci. **100**(3), 249–275 (2014)
19. Elgohary, T.A., Junkins, J.L., Atluri, S.N.: An RBF-collocation algorithm for orbit propagation. In: Advances in Astronautical Sciences: AAS/AIAA Space Flight Mechanics Meeting (2015)
20. Fehlberg, E.: Classical eighth-and lower-order Runge-Kutta-Nystrom formulas with stepsize control for special second-order differential equations. Tech. Rep. NASA-TR-r-381 NASA (1972)
21. Gladwell, L., Shampine, L., Brankin, R.: Automatic selection of the initial step size for an ODE solver. J. Guid. Control. Dyn. **18**, 175–192 (1987)
22. Hadjifotinou, K., Gousidou-Koutita, M.: Comparison of numerical methods for the integration of natural satellite systems. Celest. Mech. Dyn. Astron. **70**, 99–113 (1998)
23. Hatten, N., Russell, R.P.: Parallel implicit Runge-Kutta methods applied to coupled orbit/attitude propagation. J. Astronaut. Sci. **64**(4), 333–360 (2017)
24. Iserles, A. A First Course in Numerical Analysis of Differential Equations, 2nd edn. Cambridge University Press, Cambridge (2009)
25. Jackson, J.: Note on the numerical integration of $d^2x/dt^2 = f(x,t)$. Mon. Not. R. Astron. Soc. **84**, 602–606 (1924)
26. Jones, B., Anderson, R.: A survey of symplectic and collocation methods for orbit propagation. 22nd Annual AAS/AIAA Space Flight Mechanics Meeting, Charleston (2012)
27. Jones, B.A.: Orbit propagation using gauss-legendre collocation. In: Proceedings of the 2012 AIAA/AAS Astrodynamics Specialist Conference, AIAA, vol. 4967. pp. 1–16 (2012)
28. Junkins, J., Woollands, R.: Nonlinear differential equation solvers via adaptive picard-Chebyshev iteration: Applications in astrodynamics. In: AAS/AIAA Astrodynamics Specialist Conference (2017)
29. Macomber, B.: Enhancements of Chebyshev-Picard Iteration Efficiency for Generally Perturbed Orbits and Constrained Dynamics Systems. PhD. Dissertation, Texas A&M University, College Station (2015)
30. Miele, A., Iyer, R.: General technique for solving nonlinear, two-point boundary value problems via the method of particular solutions. J. Optim. Theory Appl. **5**(5), 392–399 (1970)
31. Montenbruck, O., Gill, E.: Satellite Orbits: Models, Methods and Applications. Springer Science & Business Media, Berlin (2012)
32. Nielsen, P., Alfriend, K., Bloomfield, M., et al.: J.T., E. Assesing air force space command's astrodynamics standards. The National Academics Press, Continuing Kepler's quest (2012)
33. Nystrom, E.J.: UBer Die Numerische Integration Von Differentialgleichungen. Druckerei der finnischen Literaturgesellschaft, Helsingfors (1925)
34. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P. Numerical Recipes 3rd Edition: The Art of Scientific Computing, 3rd edn. Cambridge University Press, New York (2007)
35. Probe, A., Read, J., Macomber, B., Junkins, J.: Massively parallel implementation of modified chebyshev picard iteration for perturbed orbit propagation. In: Advances in Astronautical Sciences: AAS/AIAA Space Flight Mechanics Meeting (2015)
36. Schaub, H., Junkins, J.: Analytical Mechanics of Aerospace Systems. 3rd edn, AIAA Education Series (2014)
37. Vallado, D.: Fundamentals of astrodynamics and applications. Space technology library (2013)
38. Van Der Houwen, P.J., Sommeijer, B.P.: Parallel iteration of high-order Runge-Kutta methods with stepsize control. J. Comput. Appl. Math. **29**(1), 111–127 (1990)

39. Wang, X., Yue, X., Dai, H., Atluri, S.: Feedback-Accelerated Iteration for Orbit Propagation and Lambert's Problem. Journal Guidance, Control and Dynamics (2017)
40. Watts, H.: Starting step size for an ode solver. J. Comput. Appl. Math. **9**(2), 177–191 (1983)
41. Woollands, R., Junkins, J.: Nonlinear Differential Equation Solvers via Adaptive Picard-Chebyshev Iteration: Applications in Astrodynamics. Journal of Guidance, Control and Dynamics (2018)
42. Wright, K.: Some relationships between implicit Runge-Kutta, collocation, and lanczos τ methods, and their stability properties. BIT Numer. Math. **10**, 217–227 (1970)
43. Zadunaisky, P.: A Method for the Estimation of Errors Propagated in the Numerical Solution of a System of Ordinary Differential Equations, pp. 281–287. International Astronomical Union, Academic Press, Paris (1966)
44. Zadunaisky, P.: On the Accuracy in the Numerical Computation of Orbits, pp. 216–227. D. Reidel Publishing Company, Dordrecht (1970)
45. Zadunaisky, P.: On the estimation of errors propagated in the numerical integration of ordinary differential equations. Numer. Math. **27**, 21–39 (1976)

Affiliations

Ahmed M. Atallah[1,2] · Robyn M. Woollands[3] · Tarek A. Elgohary[4] · John L. Junkins[3]

Tarek A. Elgohary
elgohary@ucf.edu

[1] Department of Aerospace Engineering, San Diego State University, San Diego, CA 92182, USA

[2] Mechanical and Aerospace Department, University of California San Diego, La Jolla, CA 92093, USA

[3] Department of Aerospace Engineering, Texas A, M University, TAMU 3141, College Station, TX 77843-3141, USA

[4] Department of Mechanical and Aerospace Engieering, University of Central Florida, Orlando, FL 32816, USA