

# Parallel Implicit Runge-Kutta Methods Applied to Coupled Orbit/Attitude Propagation

Noble Hatten<sup>1</sup>  · Ryan P. Russell<sup>1</sup>

Published online: 28 December 2016  
© American Astronautical Society 2016

**Abstract** A variable-step Gauss-Legendre implicit Runge-Kutta (GLIRK) propagator is applied to coupled orbit/attitude propagation. Concepts previously shown to improve efficiency in 3DOF propagation are modified and extended to the 6DOF problem, including the use of variable-fidelity dynamics models. The impact of computing the stage dynamics of a single step in parallel is examined using up to 23 threads and 22 associated GLIRK stages; one thread is reserved for an extra dynamics function evaluation used in the estimation of the local truncation error. Efficiency is found to peak for typical examples when using approximately 8 to 12 stages for both serial and parallel implementations. Accuracy and efficiency compare favorably to explicit Runge-Kutta and linear-multistep solvers for representative scenarios. However, linear-multistep methods are found to be more efficient for some applications, particularly in a serial computing environment, or when parallelism can be applied across multiple trajectories.

**Keywords** Implicit Runge-Kutta · Six degrees of freedom · Attitude · Propagation · Parallel computing

---

Presented as Paper AAS 16-395 at the 26th AAS/AIAA Space Flight Mechanics Meeting, Feb. 14–18, 2016, Napa, CA.

---

✉ Noble Hatten  
noble.hatten@utexas.edu  
Ryan P. Russell  
ryan.russell@utexas.edu

<sup>1</sup> Department of Aerospace Engineering and Engineering Mechanics,  
University of Texas at Austin, Austin TX, USA

## Introduction

The efficient propagation of space object (SO) trajectories is of vital importance to applications including mission design, conjunction analysis, and trajectory estimation and optimization. In recent years, the astrodynamics community has revisited the viability of implicit Runge-Kutta (IRK) ordinary differential equation (ODE) solvers as accurate, efficient alternatives to the widely used linear multistep and explicit Runge-Kutta (ERK) solvers [6, 14, 21, 26]. Equivalent but alternatively formulated implicit solvers like Modified Chebyshev-Picard Iteration (MCPI) have also been investigated with success [6, 11]. The dynamics model evaluations at each node within an implicit propagation step are independent, allowing such methods to take advantage of modern parallel computing paradigms within a single step – unlike linear multistep and ERK solvers. Additional benefits depend on the specific implementation, but may include, for example, high order, strong stability properties, continuous solutions, symmetry, and symplecticity [6, 11, 19, 20, 26].

In this paper, concepts for propagating three-degree-of-freedom (3DOF) SO trajectories using IRK methods are modified and extended to the propagation of coupled SO trajectory and attitude – the 6DOF problem. Efficient 6DOF propagation is important for a variety of applications, including, perhaps most obviously, determining the orientation evolution of controllable SOs with pointing requirements. In addition, body forces like aerodynamic drag and solar radiation pressure (SRP) are dependent on SO orientation. Thus, high-accuracy orbit propagation necessitates attitude knowledge, particularly for SOs that significantly violate the “cannonball” shape assumption commonly used to model body forces in the 3DOF domain [16]. In the context of space situational awareness (SSA), 3DOF propagation may be insufficient to maintain custody of and perform conjunction analysis on such SOs.

Unfortunately, 6DOF propagation is significantly less efficient than 3DOF propagation due to factors including (1) the increased size of the state vector, (2) the use of potentially complicated SO shape models, and (3) differences between the characteristic time scales of the translational and rotational dynamics. As a result, 6DOF numerical propagation generally uses much smaller time steps than 3DOF propagation, and the advancement of the state at a single time step is more computationally intensive for 6DOF propagation than for 3DOF propagation.

Several methods have been shown to modestly ease the computational burden of 6DOF propagation, such as partially decoupling the rotational and translational dynamics [16, 30, 42]. However, numerical propagation remains fundamental to these methods, and the use of more efficient ODE solvers benefits both fully coupled and semi-coupled propagation approaches. In this paper, a variable-step-size Gauss-Legendre IRK (GLIRK) ODE solver is applied to the 6DOF problem. Inspired by previous work on the 3DOF problem, customizations are introduced that greatly improve the efficiency of the GLIRK solver without sacrificing accuracy for certain classes of SOs and dynamic environments. Two example scenarios are given, and the applicability of the customizations – and the corresponding effects on propagator efficiency – are discussed. In addition, the effects of parallelizing the dynamics model evaluations of the GLIRK solver using a multicore CPU are examined, and accuracy and efficiency are compared to linear multistep and ERK solvers.

### Fully Coupled Orbit and Attitude Propagation

The 6DOF state of an SO in Earth orbit may be written as

$$x = \begin{pmatrix} r \\ v \\ \bar{q} \\ \omega \end{pmatrix}_{13 \times 1}, \tag{1}$$

where  $r$  and  $v$  are the SO’s Earth-centered inertial (ECI) position and velocity vectors, respectively,  $\bar{q}$  is a quaternion representing the orientation of a reference frame fixed to the SO body with respect to the ECI frame, and  $\omega$  is the angular velocity vector of the body-fixed frame with respect to the ECI frame, expressed in the body-fixed frame.<sup>1</sup> While the attitude may be expressed in several ways, the quaternion and angular velocity combination is chosen due to the robustness and precision of this formulation [17]. The state equations are [33]

$$\dot{x} = f(t, x) = \begin{pmatrix} v \\ \ddot{r} \\ \dot{\bar{q}} \\ \dot{\omega} \end{pmatrix}, \tag{2}$$

where

$$\ddot{r} = -\frac{\mu}{r^3}r + a_p \tag{3}$$

$$\dot{\bar{q}} = \frac{1}{2}\Omega\bar{q} \tag{4}$$

$$\dot{\omega} = J_0^{-1} [T - \omega \times (J_0\omega)]. \tag{5}$$

In Eqs. 3–5,  $\mu$  is the two-body gravitational parameter of the Earth,  $r$  is the magnitude of  $r$ ,  $a_p$  is the sum of all non-two-body accelerations acting on the SO in the ECI frame,  $J_0$  is the inertia tensor of the SO in the body-fixed frame (assumed constant), and  $T$  is the sum of external torques acting on the SO in the body-fixed frame. (Note that the time derivative of  $\omega$ ,  $\dot{\omega}$ , is expressed in the body-fixed frame.)  $\Omega$  is a matrix given by

$$\Omega = \begin{pmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ -\omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{pmatrix}, \tag{6}$$

where  $\omega = [\omega_1, \omega_2, \omega_3]^T$ .

The vectors  $a_p$  and  $T$  are sums of individual perturbing accelerations and torques, respectively. The elements of the summation are problem-specific and depend primarily upon the dynamical regime (e.g., low Earth orbit (LEO) vs. geosynchronous

<sup>1</sup>The quaternion is defined here such that  $\bar{q}_{4 \times 1} = [e^T \sin(\phi/2), \cos(\phi/2)]^T$ , where  $e_{3 \times 1}$  is the rotation axis and  $\phi$  is the rotation angle.

orbit (GEO)) and the desired model fidelity. For the purposes of the test cases presented in this paper, the perturbing accelerations considered are the non-two-body component of the geopotential, two-body gravitational accelerations caused by the Sun and Moon, aerodynamic acceleration (i.e., drag), and SRP acceleration. Meanwhile,  $\mathbf{T}$  consists of the two-body approximation of the gravity-gradient torque, aerodynamic torque, and SRP torque. In the current model, the gravitational accelerations depend only on the 3DOF state of the SO, and the gravity-gradient torque depends on the 6DOF state and  $\mathbf{J}_0$ . On the other hand, the aerodynamic and SRP accelerations and torques act on the SO external body surfaces, and are functions of the SO geometry in addition to the 6DOF state. The external SO surfaces are modeled as a set of single-sided flat panels, each of which may be given unique physical properties. The position, orientation, and surface area of each panel is fixed in the body-fixed reference frame. In this model, the SRP acceleration (in km/s<sup>2</sup>) on a single panel  $i$  is given by Wetterer et al. [41] and Früh and Jah [16]

$$\mathbf{a}_{SRP,i} = -\frac{F_{sun}AU^2}{mcD^2}H_1H_{2,i}A_i\cos\phi_i\left\{(1-C_{s,i})\mathbf{u}_{sun} + 2\left[\frac{1}{3}C_{d,i} + C_{s,i}\cos\phi_i\right]\mathbf{n}_i\right\}, \quad (7)$$

where  $F_{sun}$  is the total solar flux over all wavelengths (W/m<sup>2</sup>),  $AU$  is one astronomical unit (km),  $m$  is the mass of the SO (kg),  $c$  is the speed of light (km/s),  $D$  is the distance from the SO to the Sun (km),  $H_1$  is an Earth shadowing function,<sup>2</sup>  $H_{2,i}$  is the fraction of panel  $i$  facing the Sun,  $A_i$  is the area of panel  $i$  (km<sup>2</sup>),  $C_{s,i}$  is the specular reflectivity coefficient for panel  $i$ ,  $C_{d,i}$  is the diffusive reflectivity coefficient for panel  $i$ ,  $\mathbf{u}_{sun}$  is a unit vector pointing from the SO to the sun,  $\mathbf{n}_i$  is a unit vector in the outward normal direction of panel  $i$ , and  $\cos\phi_i = \mathbf{u}_{sun} \cdot \mathbf{n}_i$ . The total SRP acceleration experienced by the SO is the sum of the  $\mathbf{a}_{SRP,i}$ . The torque due to SRP on panel  $i$  is then calculated in the body-fixed frame by

$$\mathbf{T}_{SRP,i} = m\left[\mathbf{d}_{cm,i} \times \left(\mathbf{R}^{I \rightarrow B}\mathbf{a}_{SRP,i}\right)\right], \quad (8)$$

where  $\mathbf{d}_{cm,i}$  is the vector from the center of mass of the SO to the center of pressure of panel  $i$  and  $\mathbf{R}^{I \rightarrow B}$  is the transformation matrix from the ECI frame to the body-fixed frame. As with the acceleration, the total SRP torque experienced by the SO is the sum of the  $\mathbf{T}_{SRP,i}$ .

The aerodynamic acceleration and torque are also calculated for each panel individually following the method of Doornbos [15]. In the interest of brevity, the full details of the method are not presented here; at the most basic level, the method calculates the vector aerodynamic coefficient  $\mathbf{C}_{a,i}$  for each panel  $i$  based on atmospheric properties. Then, the aerodynamic acceleration on panel  $i$  is given by

$$\mathbf{a}_{aero,i} = \frac{1}{2}\frac{A_{ref}\rho v_r^2}{m}\mathbf{C}_{a,i}, \quad (9)$$

<sup>2</sup>A cylindrical Earth shadowing function is assumed in this paper.

where  $A_{ref}$  is a constant reference area,<sup>3</sup>  $\rho$  is the atmospheric density, and  $v_r$  is the magnitude of the relative velocity between the SO and the atmosphere. Aerodynamic torque is then found following Eq. 8.

### Implicit Runge-Kutta ODE Solver

A GLIRK ODE solver is written to propagate the 6DOF equations of motion. A description of the fundamental GLIRK method and IRK methods in general is given by Hairer et al. [19], while descriptions within the context of 3DOF SO propagation are found in Jones [26], Aristoff et al. [6], and Aristoff and Poore [5]. The referenced works also discuss the motivations for selecting GLIRK amongst the range of available IRK methods for astrodynamics applications. For example, the GLIRK method is superconvergent: An implementation using  $s$  internal stages per step produces a solution accuracy of order  $2s$ , the highest possible for an RK method. Additionally, the GLIRK method is both symmetric and symplectic, and exhibits both  $A$  and  $B$  stability.

Any explicit or implicit RK method with  $s$  internal nodes (or stages) may be summarized as advancing an approximation of the state vector  $\mathbf{x}$  from time  $t_n$  to time  $t_{n+1}$  via the equations

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h \sum_{j=1}^s b_j \mathbf{k}_j \tag{10}$$

$$\mathbf{k}_i = \mathbf{f} \left( t_n + hc_i, \mathbf{x}_n + h \sum_{j=1}^s a_{i,j} \mathbf{k}_j \right), \quad i = 1, \dots, s. \tag{11}$$

$$= \mathbf{f} (t_i, \mathbf{x}_i), \quad i = 1, \dots, s. \tag{12}$$

The arrays  $\mathbf{A}_{s \times s}$ ,  $\mathbf{b}_{s \times 1}$ , and  $\mathbf{c}_{s \times 1}$  – whose elements are the  $a_{i,j}$ ,  $b_j$ , and  $c_j$ , respectively – define the RK method.<sup>4</sup> In ERK methods, the  $a_{i,j}$  are zero for  $j \geq i$ , which allows for the  $\mathbf{k}_i$  to be calculated explicitly in a sequential fashion. On the other hand, IRK methods violate this restriction, and the  $\mathbf{k}_i$  that satisfy Eq. 11 must be determined by simultaneously solving a system of algebraic equations. For nonlinear systems of ODEs, solving Eq. 11 requires an iterative method such as fixed-point iteration, the Newton-Raphson method, or the Gauss-Seidel method [5]. In the current work, fixed-point iteration is used because the method is parallelizable (unlike Gauss-Seidel) and avoids the computation of the Jacobian (unlike Newton-Raphson). It is noted that a Newton or quasi-Newton method (as opposed to a fixed-point method) must be used

<sup>3</sup>The area and orientation of each individual panel  $i$  is taken into account in the calculation of  $C_{a,i}$ .

<sup>4</sup>The defining arrays of the GLIRK method are calculable for arbitrary  $s$ , and may be obtained to high precision using, for example, the Mathematica function `NDSolve`ImplicitRungeKuttaGaussCoefficients` [3].

in order to retain the  $A$  stability property of a GLIRK solver and guarantee convergence of Eq. 11 for large step sizes [5]. In practice, however, convergence of the fixed-point iterations may be monitored numerically at each step, and the step size may be reduced if divergence is predicted. For reasonably high-precision 3DOF or 6DOF astrodynamics applications, step size reductions of this type are rare enough that any associated cost is significantly less than that of the Jacobian calculations required by Newton-based methods [8].

The alternative implicit ODE solver MCPI is most commonly used as a high-order method (i.e., large  $s$ ) that takes large, predetermined step sizes on the order of one-to-five steps per orbital period for 3DOF applications [11, 32]. However, previous work has shown variable-step-size implementations of GLIRK to propagate 3DOF dynamics more efficiently than an untuned MCPI method for certain example cases [6]. Additionally, MCPI performance is dependent on the predetermination of a suitable step size because no general, truncation-error-based adaptive-step-size implementation has been published. For the 6DOF problem, this goal of a general variable-step implementation is important because of the multiple driving frequencies of the combined rotational and translational dynamics.

The need for a predetermined step size also holds for published implementations of bandlimited collocation IRK (BLC-IRK), an ODE solver that bases its quadrature scheme on a weighted sum of exponentials rather than on a weighted sum of polynomials, as is done by both GLIRK and MCPI [14]. Like MCPI, BLC-IRK is designed to work with large  $s$  and may outperform GLIRK if a near-optimal fixed step size is known *a priori* [6]. On the other hand, a propagator with a mechanism for adapting the step size to meet a user-defined local truncation error (LTE) tolerance avoids the difficulty of determining an appropriate fixed step size. Variable-step-size mechanisms exist for GLIRK, and such methods are the focus of this study [6, 26, 28].

## Variable-Fidelity Dynamics Models

Though the necessity of iterative methods to solve Eq. 11 is generally a disadvantage in terms of computational efficiency for IRK methods, problem-specific customizations of the iterative solver may greatly reduce the computational burden of the solution process. For example, a high-fidelity 3DOF SO dynamics model is computationally intensive. However, a lower-fidelity approximation of the dynamics model may be obtained for a small fraction of the computational price. Under the assumptions that the lower-fidelity dynamical model suitably approximates the full dynamics model and that the  $k_i$  do not undergo large changes between iterations, the lower-fidelity model may be evaluated instead of the full model at some iterations without compromising the accuracy of the converged solution of Eq. 11 [5, 6, 14, 26, 32].

In this paper, a similar approach is taken for fully coupled 6DOF propagation, as described in Algorithm 1. The forces and torques considered in the low-fidelity model are specified in the discussion of each example simulation.

**Algorithm 1** Iterative solution of IRK step (Eq. (11)) using variable-fidelity dynamics model

```

Set convergence tolerances:  $\epsilon_1 > \epsilon_2 > \epsilon_3 > 0$ 
 $\mathbf{x}_i \leftarrow \mathbf{x}_{i,guess}, i = 1, \dots, s$                                 ▷ Set initial guess for  $\mathbf{x}_i$ 
 $j \leftarrow 1$                                                     ▷ Start with first iteration tolerance
evaluation fidelity  $\leftarrow$  low                                    ▷ Start with low-fidelity dynamics model
 $\Delta \mathbf{f}_i \leftarrow \mathbf{0}, i = 1, \dots, s$ 
while  $j \leq 3$  do                                             ▷ Loop through iteration tolerances
    if evaluation fidelity == low then
         $\mathbf{f}_i \leftarrow \mathbf{f}_{low}(t_i, \mathbf{x}_i), i = 1, \dots, s$     ▷ Low-fidelity function evaluation
         $\mathbf{f}_i \leftarrow \mathbf{f}_i + \Delta \mathbf{f}_i, i = 1, \dots, s$     ▷ Update full function approximation
    else
         $\mathbf{f}_i \leftarrow \mathbf{f}_{high}(t_i, \mathbf{x}_i), i = 1, \dots, s$     ▷ High-fidelity function evaluation
         $\Delta \mathbf{f}_i \leftarrow \mathbf{f}_{low}(t_i, \mathbf{x}_i), i = 1, \dots, s$     ▷ Low-fidelity function evaluation
         $\Delta \mathbf{f}_i \leftarrow \mathbf{f}_i - \Delta \mathbf{f}_i, i = 1, \dots, s$     ▷ High-fidelity contribution to model
    end if
    Update  $\mathbf{x}_i, i = 1, \dots, s$ , using fixed-point iteration update equations
     $\delta \leftarrow$  norm of change over all  $\mathbf{x}_i, i = 1, \dots, s$ 
    if  $\delta < \epsilon_j$  then                                       ▷ Converged to within current iteration tolerance
         $j \leftarrow j + 1$                                        ▷ Proceed to next iteration tolerance
        evaluation fidelity  $\leftarrow$  high    ▷ Perform high-fidelity function evaluation at
next iteration
    else
        evaluation fidelity  $\leftarrow$  low    ▷ Perform low-fidelity function evaluation at
next iteration
    end if
end while

```

When using Algorithm 1, the high-fidelity dynamical model is evaluated at only two iterations per propagation step, regardless of the number of total iterations required for convergence. The three iteration convergence tolerances  $\epsilon_1, \epsilon_2$ , and  $\epsilon_3$  are user-defined parameters that may be used to control the precision of the solution. Though similar in spirit, this approach differs from previously published GLIRK variable-fidelity models used for 3DOF propagation. For example, both Jones [26] and Aristoff and Poore [5] utilize one or more low-fidelity dynamics model iterations before switching to the high-fidelity model at all subsequent iterations. Alternatively, as is done in Algorithm 1, Bradley et al. [14] use two high-fidelity iterations per step. However, in the Bradley algorithm, the two high-fidelity iterations are separated by a fixed number of low-fidelity iterations, and no subsequent low-fidelity iterations are performed following the second high-fidelity iteration. In the context of MCPI, Macomber [32] implements an approach similar to Algorithm 1, but the validity of the approach is only investigated for a 3DOF dynamical model consisting of the geopotential alone. In the present work, numerical examples show that this

variable-fidelity dynamics model approach is valid for the 6DOF propagation of an SO in the presence of realistic forces and torques – provided that the low-fidelity dynamics model is selected appropriately.

All tests presented in this paper use  $\epsilon_1 = 10^{-5}$ ,  $\epsilon_2 = 10^{-12}$ , and  $\epsilon_3 = 10^{-15}$ . Divergence of the iterative solver is handled by placing a maximum value on the number of iterations: If the solver does not converge to within  $\epsilon_3$  in 50 iterations, the step size is halved, and the step is rerun.

Initial guesses for the  $x_i$  are provided using approximate analytical propagation methods: An elliptical orbit is assumed for the translational propagation [12], and constant-angular-velocity rotation is assumed for the rotational propagation [43]. Analytical solutions to more physically representative models, such as torque-free rotation [43], exist, but are more computationally demanding and are not found to improve overall efficiency when combined with the variable-fidelity dynamics model strategy for the scenarios presented in this paper. At early iterations, the fixed-point solver converges toward an approximate solution because only the inexpensive, low-fidelity dynamics model is evaluated. Therefore, the improvement in the initial guess between the constant-angular-velocity solution and the torque-free-rotation solution is unnecessary. If the full, high-fidelity dynamics model is evaluated at every iteration, a more accurate initial guess becomes more valuable.

### Variable-Step-Size Propagation

Variable-step-size ODE solvers adaptively vary the step size of the independent variable by comparing the estimated LTE of the method to a user-prescribed tolerance. In this way, variable-step-size methods attempt to maximize efficiency by taking the minimum number of propagation steps needed to meet the user's accuracy requirements. ERK methods commonly use embedded methods to estimate the LTE; that is, each propagation step is performed by two methods, each of unique order, that share the same  $A$  and  $c$  arrays but are defined by different  $b$  arrays. This strategy allows for the efficient calculation of state estimates of different order. The LTE is then estimated based on the difference between the state estimates produced by the two methods and knowledge of the order of the LTE of each method [37].

For the GLIRK ODE solver, an embedded method of nearly the same order as the propagation method is difficult to achieve because an  $s$ -stage GLIRK method gives a solution of order  $2s$ , the highest order achievable by an RK method [6]. For this reason, previous authors have utilized alternative methods for creating a variable-step-size GLIRK ODE solver for 3DOF SO propagation. For example, Jones [26] employs a method of van der Houwen and Sommeijer [23] that uses the differences between the solutions produced at consecutive iterations of the iterative solution process to approximate the LTE. Aristoff [4], on the other hand, uses a second, non-embedded IRK propagation of order near  $2s$  (such as a Radau method) to produce a high-order LTE estimate. However, to the authors' knowledge, the precise methodology has not been specified in a published work [4–8]. The computational cost of the second IRK propagation is reduced by using the collocation polynomial produced by the original GLIRK propagation to obtain an initial guess (of order  $s$ ) to Eq. 11 [19].



In the current work, a lower-order (order  $s$ ), nearly embedded<sup>5</sup> solution is used to estimate the LTE [18, 28]. A solution  $\hat{\mathbf{x}}$  of order  $s$  is given by

$$\hat{\mathbf{x}}_{n+1} = \mathbf{x}_n + h \left( \gamma_0 \mathbf{f}(t_n, \mathbf{x}_n) + \sum_{j=1}^s \hat{\mathbf{b}}_j \mathbf{k}_j \right). \tag{13}$$

The elements  $\hat{\mathbf{b}}_j$  make up the array  $\hat{\mathbf{b}}_{s \times 1} = \mathbf{V}_{s \times s}^{-1} \mathbf{u}_{s \times 1}$ , where

$$\mathbf{V} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ c_1 & c_2 & \dots & c_s \\ \vdots & \vdots & \ddots & \vdots \\ c_1^{s-1} & c_2^{s-1} & \dots & c_s^{s-1} \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} 1 - \gamma_0 \\ \frac{1}{2} \\ \vdots \\ \frac{1}{j} \\ \vdots \\ \frac{1}{s} \end{pmatrix}. \tag{14}$$

The scalar  $\gamma_0$  is a user-selected parameter. Following Kouya [28], the method implemented in this paper uses  $\gamma_0 = 1/8$ . Once  $\mathbf{x}_{n+1}$  and  $\hat{\mathbf{x}}_{n+1}$  are known, the procedure for adapting the step size proceeds identically to an ERK method:

$$h_{new} = h_{old} \cdot \min \left[ \Delta h_{max}, c \left( \frac{1}{\delta} \right)^p \right] \tag{15}$$

$$\delta = \sqrt{\frac{1}{N} \sum_{i=1}^N \left( \frac{\mathbf{x}_{i,n+1} - \hat{\mathbf{x}}_{i,n+1}}{\tau_i} \right)^2} \tag{16}$$

$$\tau_i = atol + rtol \cdot \max(|\mathbf{x}_{i,n}|, 1) \tag{17}$$

$$p = \frac{1}{s + 1} \tag{18}$$

$$c = 0.8 \tag{19}$$

$$\Delta h_{max} = 2, \tag{20}$$

where subscript  $i$  indicates element  $i$  of the  $N$ -dimensional state vector,  $atol$  and  $rtol$  are user-defined absolute and relative tolerance parameters, respectively, and  $0 < c < 1$  is set to increase the likelihood that the subsequent step is accepted. In the event  $\delta = 0$  to double precision, the step size is increased by a factor of 10. Note that if  $h_{new} > h_{old}$ , then the current step is accepted and  $h_{new}$  is used as the step size for the next time step, while, if  $h_{new} < h_{old}$ , the current step is rejected and retried with  $h_{new}$  as the step size.

The low computational cost of the LTE estimation procedure is offset by the low order of the nearly embedded solution (order  $s$ ) relative to the GLIRK solution (order  $2s$ ). In other words, for step size adaptation purposes, the GLIRK method is treated as an order- $(s + 1)$  method. The result is that the LTE estimate  $\delta$  is likely to be larger

<sup>5</sup>The embedded solution requires a single additional high-fidelity dynamics model evaluation at the initial time and state of the step.

than the actual LTE of the GLIRK method. Thus,  $h_{new}$  is likely to be underestimated, and some steps that should be acceptable may be rejected. One method for addressing large differences between the orders of the propagated and embedded methods is Jay's "internal tolerance" scheme [25]. However, use of the internal tolerance is not found to reliably improve performance in the current application; an illustration of this finding is given in Appendix A. The test cases presented in this paper use the more conservative step size selection scheme given by Eqs. 13–20.

A disadvantage of variable-step-size methods is the costly re-propagation of a step if the estimate of the LTE is found to be larger than the user-specified tolerance. In order to proactively seek out steps that are likely to fail, the current GLIRK implementation estimates the LTE following the first evaluation of the high-fidelity dynamics model (following convergence of the fixed-point iteration to within  $\epsilon_1$ ). If the estimate of the LTE is greater than the tolerance (i.e.,  $\delta > \tau$ ), the fixed-point iteration is aborted, and the step is rerun using the step size computed using Eq. 15. Thus, the second evaluation of the high-fidelity dynamics model at each stage – which would have been wasted due to the rejection of the step – is avoided. While it is possible for false detections to cause unnecessary recomputations, empirical evidence suggests that false detections are rare relative to true detections because of the proximity of the partially converged solution to the fully converged solution.

An estimate for the step size to be used at the first propagation step is calculated using the initial value of the state and its time derivative, as described in Algorithm 2 [39].

---

**Algorithm 2** Calculation of estimate for initial step size  $h_0$

---

Set minimum and maximum step sizes  $h_{min}$  and  $h_{max}$

$$h_0 \leftarrow h_{max}$$

$$\alpha \leftarrow \frac{atol}{\max(rtol, \epsilon_m)}$$

$\triangleright \epsilon_m$  is machine epsilon

$$\beta \leftarrow \frac{\maxval(|f(t_0, x_0)|)}{\max(\maxval(|x_0|), \alpha, \epsilon_m)}$$

$$h_g \leftarrow \frac{\beta}{c \cdot rtol^p}$$

$\triangleright p$  is defined in Eq. 18;  $c$  is defined in Eq. 19

**if**  $h_g == 0$  **then**

$$h_0 \leftarrow h_{min}$$

**else if**  $h_g \cdot h_0 > 1$  **then**

$$h_0 \leftarrow 1/h_g$$

**end if**

$$h_0 \leftarrow \max(h_0, h_{min})$$

$\triangleright$  Ensure  $h_0 \geq h_{min}$

---

## Parallelization

Parallel computation within a single propagation is a trait of IRK methods not shared by ERK methods or linear multistep methods.<sup>6</sup> The opportunity for parallelization

---

<sup>6</sup>It is noted that, for any ODE solver, the propagation of state transition tensors may be parallelized within a single propagation step over the dimension of the state.

arises from the fact that, when solving Eq. 11 for an IRK method, each  $k_j$  evaluation is independent. Thus, each of the  $s$   $k_j$  evaluations required at each iteration of the implicit solver may be performed in parallel when using fixed-point iteration. Such parallelization has the potential to yield strong benefits for both 3DOF and 6DOF SO propagation because the high-fidelity dynamics model evaluations required to compute the  $k_j$  generally dominate propagation runtime. While the possibility of parallelization of 3DOF GLIRK methods has been suggested previously [5, 6, 14, 26], analysis has been limited to the assumption of linear speedups – i.e., the use of  $s$  parallel threads reduces total runtime by a factor of  $s$ . However, this assumption provides only a best-case approximation of efficiency gains. Linear speedups are not typical for realistic implementations because the CPU time required to evaluate a typical dynamical model may not be large enough to hide the overhead of parallelization.

In this paper, the effect of parallelization on the efficiency of the GLIRK method applied to 6DOF SO propagation is studied using the OpenMP library with up to 23 threads [2]. Dynamical models of varying complexity are implemented to examine the consequences of parallelization overhead. At each iteration of the solution process of Eq. 11 at which high-fidelity dynamics model evaluations are required, the  $s$   $k_j$  are evaluated in parallel using an `!$omp parallel do` loop. Static scheduling is used due to the very similar workload of each loop iteration. Parallelization of the evaluations of the low-fidelity dynamics model and the initial guesses for the  $x_j$  is also possible. However, because of the short runtime of these routines, serial evaluation is sometimes preferred because of the overhead associated with parallelization. Further discussion is given for the two example scenarios described in the “Results” section.

## Results

The accuracy and efficiency of the GLIRK propagator are compared to a standard implementation of the explicit Runge-Kutta-Dormand-Prince (RKDP) 8(7) propagator and to the public-domain linear multistep (Adams-Moulton) propagator LSODE [37, 38].<sup>7</sup> Results are presented for two scenarios: A rectangular prism SO in LEO and a high-area-to-mass-ratio (HAMR) SO in GEO. The 6DOF equations of motion are propagated with each ODE solver for three orbital periods unless otherwise specified. The quaternion unit norm constraint is enforced by renormalizing  $\bar{q}$  after each propagation step for all propagators. Accuracy is assessed via a “truth” model calculated using a quadruple-precision implementation of RKDP8(7) with a relative LTE tolerance of  $10^{-25}$ . For GLIRK parallelization, the number of available threads is set to  $(s + 1)$  so that each of the dynamical model evaluations – including the extra evaluation used in the estimation of the LTE – is provided a single thread. All code is written in Fortran and compiled using the Intel Visual Fortran Compiler

---

<sup>7</sup>All input options for LSODE are set to default values (with method flag = 10) except for the relative and absolute tolerance parameters: The absolute tolerance is set uniformly to machine epsilon, and the relative tolerance is varied.

**Table 1** Initial state for SO in LEO

(a) Translational state in classical orbital elements		(b) Rotational state	
$a$	6,745.592 km	$q_1$	0
$e$	0.01	$q_2$	0
$i$	7.81 deg	$q_3$	0
$\Omega$	100.21 deg	$q_4$	1
$\omega$	152.83 deg	$\omega_1$	0.573 deg/s
$\nu$	0 deg	$\omega_2$	0.573 deg/s
		$\omega_3$	0.573 deg/s

XE 14.0.0.103 (64-bit) using the `-O2` optimization flag. All computations are performed on a 64-bit Windows 7 Enterprise workstation with two 12-core Intel Xeon E5-2680 v3 processors (clock speed 2.50 GHz) and 64 GB of RAM. Hyperthreading is disabled.

It is noted that, in the presentation of results, discrete data points (indicated by shapes in figures) are connected by lines. This convention is *not* intended to convey information regarding trends between data points, but merely to aid the reader in differentiating between data sets.

### Object in Low Earth Orbit

The first test scenario is a tumbling rectangular prism SO in LEO. The initial state is given in Table 1, and the SO's physical characteristics are described in Tables 2 and 3. Note that the SO's panels are given non-uniform reflectance properties in order to induce an SRP torque. Tables 4 and 5 give the high-fidelity and low-fidelity dynamics models, respectively. In all results given for this scenario, parallelization is limited to the high-fidelity dynamics model; parallelization of the low-fidelity model and initial guess generation is found not to improve efficiency for this scenario.

The differing frequencies of the translational and rotational motions are demonstrated in the state evolutions shown in Fig. 1; for this scenario, the rotational dynamics evolve much more rapidly than the translational dynamics. The performance of the GLIRK propagator as a function of the number of stages  $s$  is shown

**Table 2** SO basic physical characteristics

Property	Value
Shape	Rectangular prism
Dimensions	1.5 m $\times$ 2 m $\times$ 3 m
Mass	175 kg
Number of panels	6

**Table 3** SO panel reflectance properties

Property	Panel dimensions (m)					
	2 × 3	2 × 3	1.5 × 3	1.5 × 3	1.5 × 2	1.5 × 2
$C_d$	0.3	0.35	0.25	0.25	0.15	0.2
$C_s$	0.2	0.15	0.25	0.25	0.35	0.3

in Fig. 2 for a three-orbit propagation.<sup>8</sup> Figure 2a confirms that the use of the variable-fidelity models does not degrade propagation accuracy. Figure 2b depicts the performance gains that may be realized by using variable-fidelity dynamics models in the fixed-point iteration. The optimal number of stages for a serial implementation for this scenario is seen to be in the range of 8–12, which is found to be typical for this application. For a parallel implementation, the optimal number of stages depends on the dynamical model: As the CPU time required by the dynamical model increases, larger relative efficiency gains are available from parallelization using a high number of stages and threads. For this scenario, Fig. 2b shows that efficiency gains using more than eight stages and nine threads are minimal for either the high-fidelity-only or variable-fidelity implementations.

Figure 2c explicitly shows the efficiency gains achieved by parallelizing the dynamical model evaluations at each propagation step. When the high-fidelity dynamics model is evaluated at every iteration, these evaluations consume a greater percentage of propagator runtime than when the variable-fidelity model is used. Thus, greater relative efficiency gains are observed for the high-fidelity-only propagator. As expected, the efficiency gains from parallelization lag the number of parallel threads employed. For the variable-fidelity propagator, speedups between 3× and 4× compared to the serial performance are achieved for  $s \geq 8$ , while, for the high-fidelity-only propagator,  $s \geq 16$  delivers relative speedups between 7× and 8×. However, despite the superior *relative* efficiency gains of the high-fidelity-only propagator, greater *absolute* performance is still achieved via the variable-fidelity propagator for this scenario (Fig. 2b).

Figure 2d displays the relative efficiency improvements for the eight-stage GLIRK propagator as a function of the number of parallel threads available. Efficiency jumps correspond to threadcount increases that decrease the number of loop iterations required. For example, the high-fidelity-only propagator sees a large efficiency gain between seven and eight threads because  $s = 8$ . As expected, the availability of more than  $(s + 1)$  threads provides no further efficiency gains.

The impact of the variable-fidelity dynamics model strategy is further underscored in Fig. 3, which displays the relative CPU time requirements of each of the constituents of the high-fidelity dynamics model. The low-fidelity dynamics model evaluation time is also shown for comparison. The benefits of eliminating evaluations

<sup>8</sup>Each subfigure displays results corresponding to a single identical value of relative LTE tolerance,  $rtol = 10^{-15}$ .

**Table 4** High-fidelity dynamics model for SO in LEO

Source	Force model	Torque model
Gravity	70×70 geopotential (spherical harmonics or interpolated [10]); Sun and Moon point mass (cubic spline ephemerides [9])	Earth gravity gradient
SRP	Direct (panel model)	Direct (panel model)
Drag	Doornbos model (panel-based) [15]; modified Harris-Priester atmospheric density [31]	Doornbos model (panel-based)

of the high-fidelity geopotential are clear – even though the runtime of the interpolated 70×70 field is approximately equivalent to that of a 15×15 field if the more common spherical harmonics formulation is used.

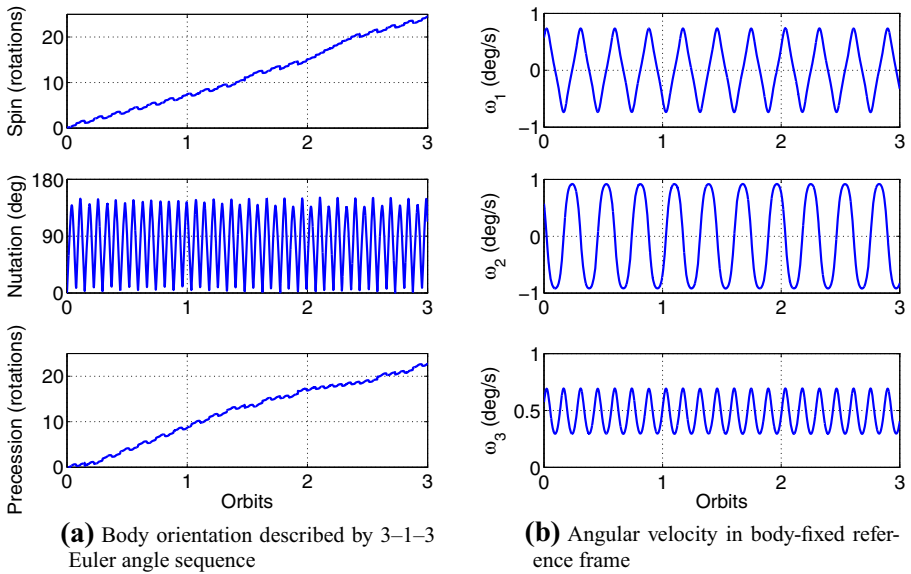
Figure 4 compares root-mean-square (RMS) accuracy and efficiency of the eight-stage GLIRK propagator to RKDP8(7) and LSODE for relative LTE tolerances ranging from  $10^{-5}$  to  $10^{-15}$ . Among serial implementations, the linear multistep solver LSODE is more efficient than either the IRK or ERK solvers, particularly for stringent LTE tolerances. This result supports the substantial heritage of linear multistep solvers for serial astrodynamics applications [35]. For example, the Gauss-Jackson method is used by United States Air Force Space Command to propagate the SO catalog using special perturbation techniques [1, 13, 24, 27]. In addition, the DIVA propagator used by the Jet Propulsion Laboratory is a linear multistep integrator [29], and LSODE is the default propagator in the Copernicus mission design software [36].

The poor performance of the serially implemented, high-fidelity-only GLIRK propagator demonstrates why the method has not been traditionally popular for astrodynamics applications. On the other hand, the variable-fidelity GLIRK propagator gives comparable or superior performance compared to the RKDP8(7) propagator, even when implemented serially. When parallelized, variable-fidelity GLIRK also outperforms LSODE, showing improved efficiency for a given accuracy. For a 30-orbit propagation, the parallelized variable-fidelity GLIRK propagator achieves sub-centimeter-level accuracy in 31 percent less CPU time than LSODE and 61 percent less CPU time than RKDP8(7) (Fig. 4c).

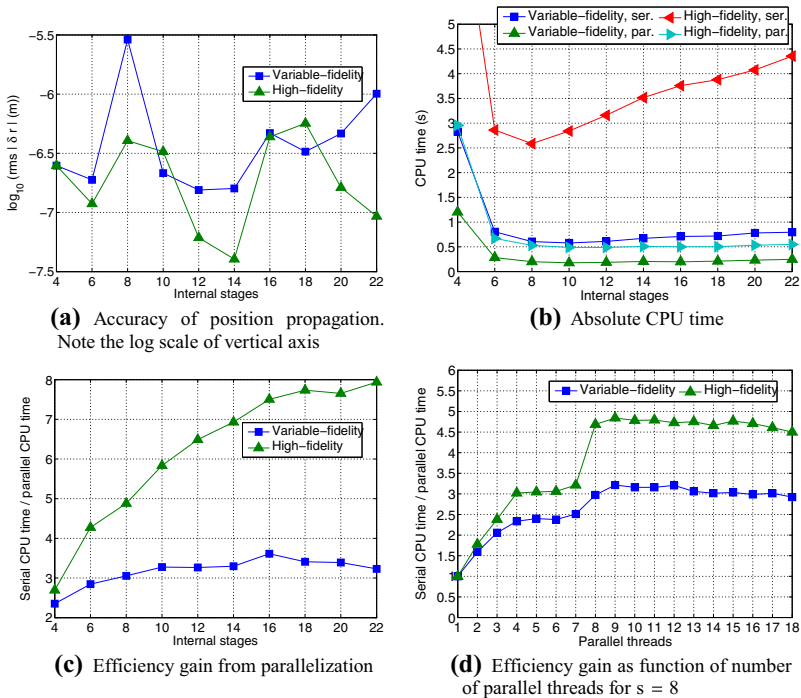
As the effects of parallelization and the variable-fidelity dynamics model increase with increasing dynamics model complexity, even further relative speed gains are possible. For example, higher-fidelity atmospheric density models can be an order of magnitude more computationally expensive than the modified Harris-Priester model

**Table 5** Low-fidelity dynamics model for SO in LEO

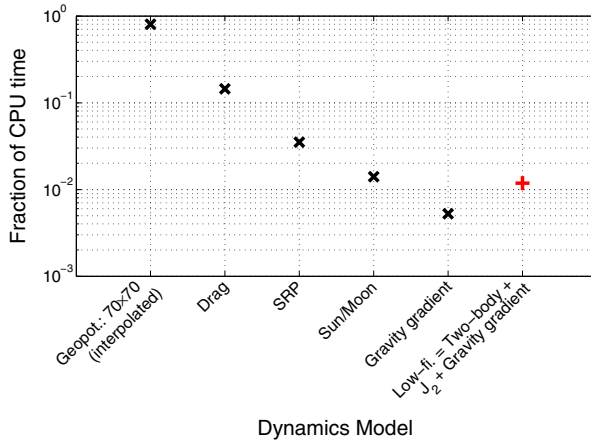
Source	Force model	Torque model
Gravity	Earth point-mass and $J_2$ terms	Earth gravity gradient
SRP	None	None
Drag	None	None



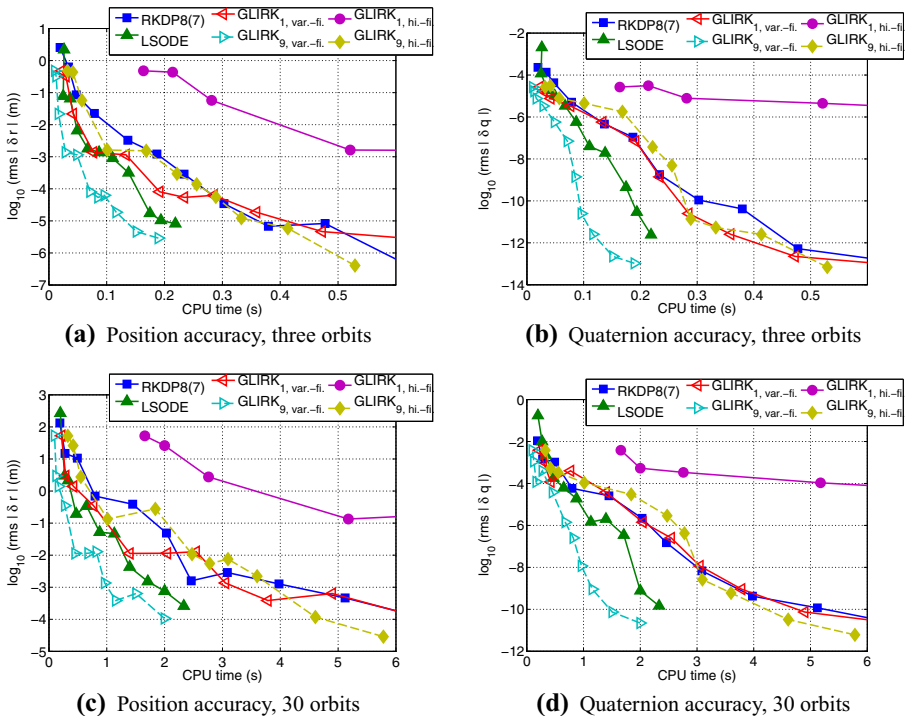
**Fig. 1** Rotational state evolution for three-orbit LEO scenario



**Fig. 2** Performance of GLIRK propagator as a function of number of stages  $s$  using both variable-fidelity and high-fidelity-only dynamics models. Unless otherwise specified, the number of threads made available for parallelization is equal to  $(s + 1)$



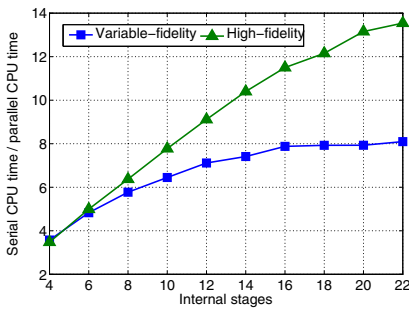
**Fig. 3** Fraction of total high-fidelity dynamics model CPU time spent evaluating each model constituent for three-orbit LEO scenario; alternative low-fidelity dynamics model time shown for comparison. (Low-fidelity model time is not included in summation used to compute total time). Note logarithmic scale of vertical axis



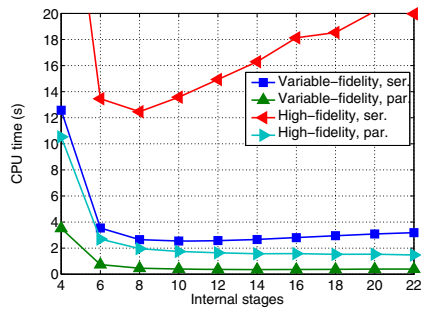
**Fig. 4** Accuracy vs. CPU time for propagation of tumbling SO in LEO; relative LTE tolerance of propagators is varied from  $10^{-5}$  –  $10^{-15}$ . Eight-stage GLIRK propagator is used; numeric subscript in legend represents number of parallel threads used in computation; “var.-fi.” indicates use of variable-fidelity dynamics model, while “hi.-fi.” indicates use of high-fidelity dynamics model only. Note the log scale of vertical axes



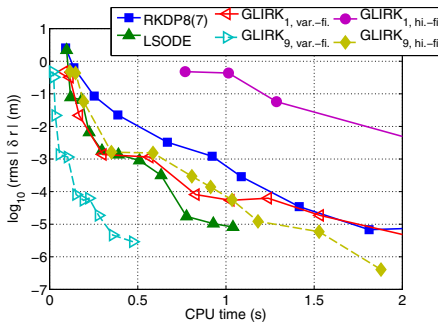
implemented in this example [22]. Many more than six panels may be required to adequately represent an SO body. Additionally, the propagation of state transition tensors via the variational equations is often important – and significantly increases the computational burden of the dynamical model. This phenomenon is clearly demonstrated in Fig. 5, in which the  $70 \times 70$  geopotential is calculated using spherical harmonics instead of the significantly faster interpolated gravity model. The peak relative effect of parallelization increases significantly for both the high-fidelity-only and variable-fidelity GLIRK propagators (Fig. 2c vs. Fig. 5a), and the parallelized variable-fidelity GLIRK is now four times faster than LSODE for sub-centimeter-level position error after three orbital periods (Fig. 5c). Additionally, comparing Fig. 5b to Fig. 2b, larger relative efficiency increases from parallelization allow for runtime improvements



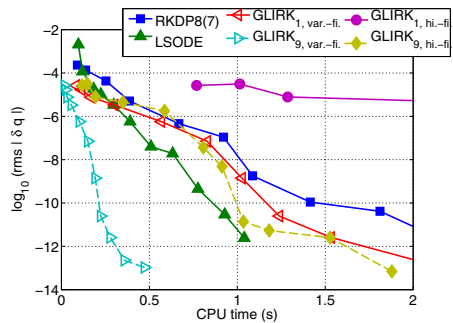
(a) Efficiency gain from parallelization. The number of threads made available for parallelization is equal to  $(s + 1)$



(b) Absolute CPU time

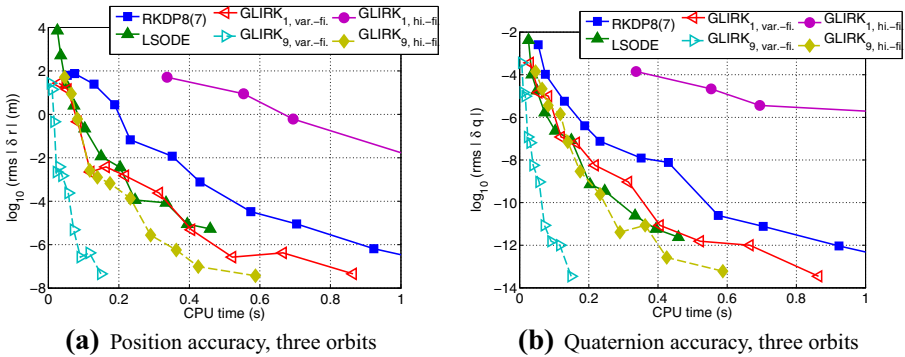


(c) Position accuracy, three orbits; relative LTE tolerance of propagators is varied from  $10^{-5}$ – $10^{-15}$ . Eight-stage GLIRK propagator is used. Note the log scale of vertical axis



(d) Quaternion accuracy, three orbits; relative LTE tolerance of propagators is varied from  $10^{-5}$ – $10^{-15}$ . Eight-stage GLIRK propagator is used. Note the log scale of vertical axis

**Fig. 5** Effect of more computationally expensive dynamics function. Geopotential is calculated using spherical harmonics, which is approximately an order of magnitude slower than the interpolated model used to produce the results of Figs. 2, 3, and 4 [10]



**Fig. 6** Accuracy vs. CPU time for propagation of non-tumbling SO in LEO; relative LTE tolerance of propagators is varied from  $10^{-5} - 10^{-15}$ . Eight-stage GLIRK propagator is used. Note the log scale of vertical axes

at larger  $s$ . However, gains achieved beyond 8–12 stages are minimal due to the inefficient serial performance of the high-order methods.

Further efficiency improvements for GLIRK relative to RKDP8(7) and LSODE are possible when the solvers are applied to uncertainty propagation [8]. Sigma-point filters such as the unscented Kalman filter (UKF) require the propagation of a population of neighboring state vectors [34]. For relatively small initial uncertainties, the differences between the neighboring state vectors are also small. In this situation, one state vector (e.g., the mean) may be propagated using an unmodified GLIRK method, and the converged times and states of each internal stage of each step may be saved. Then, when propagating the remaining state vectors, the variable time steps produced by the initial propagation may be reused, eliminating costly rejected steps. While this modification is not unique to the GLIRK method, each individual step of an eight-stage GLIRK is more computationally expensive than popular ERK and linear multistep methods because the GLIRK method is single-step and high-order. Thus, the elimination of a rejected – and therefore recalculated – step is more meaningful for GLIRK than for competing methods. In addition, the saved states at each internal stage of each step may be used as accurate initial guesses for the internal stages of

**Table 6** Initial state for HAMR SO in GEO

	(a) Translational state in classical orbital elements	(b) Rotational state	
$a$	42,164 km	$q_1$	0
$e$	0.001	$q_2$	0
$i$	1 deg	$q_3$	0
$\Omega$	145.92 deg	$q_4$	1
$\omega$	266.13 deg	$\omega_1$	0 deg/s
$\nu$	0 deg	$\omega_2$	0 deg/s
		$\omega_3$	0 deg/s

**Table 7** SO physical characteristics

Property	Value
Shape	Two-sided flat plate
Dimensions	1 m × 1 m
Mass	0.038 kg
Number of panels	2
$C_d$ (uniform)	0.26
$C_s$ (uniform)	0.6
Center-of-mass-center-of-pressure offset	$1.414 \times 10^{-3}$ m

the neighboring propagations, potentially reducing the number of iterations required for convergence, depending on the proximity of the neighboring state vectors.

The spherical-harmonics-based LEO scenario is re-simulated assuming that the previously described SO is non-tumbling ( $\omega_0 = [0, 0, 0]^T$ ). Figure 5c and d are duplicated for this case in Fig. 6a and b, respectively. For the non-tumbling case, the absolute CPU time required by all ODE solvers decreases because the attitude state changes more slowly, allowing for larger propagation step sizes. On the other hand, the performances of the ODE solvers relative to one another are similar to the tumbling case (Fig. 5). In a serial environment, LSODE and the variable-fidelity GLIRK propagator display generally similar efficiency profiles, while the RKDP8(7) propagator requires more CPU time to achieve a given accuracy. When parallelization is used, the variable-fidelity GLIRK propagator once again provides the greatest efficiency among the options considered, here achieving an approximately 6× speedup over LSODE for approximately sub-centimeter-level position error.

### High-Area-to-Mass-Ratio Object in Geosynchronous Orbit

A second test scenario consists of a HAMR two-sided flat plate in GEO. An offset between the center of mass and center of pressure of the plate is assumed in order to induce an SRP torque. For the primary analysis of this example, the SO is not tumbling, and instead begins with zero angular velocity. The complete initial state and SO physical characteristics are given in Tables 6 and 7, respectively. In all results given for this scenario, both the low-fidelity and high-fidelity dynamics models are parallelized because of the relative expense of the low-fidelity model, as discussed below.

**Table 8** High-fidelity dynamics model for SO in GEO

Source	Force model	Torque model
Gravity	8×8 geopotential (spherical harmonics); Sun and Moon point mass (cubic spline ephemerides)	Earth gravity gradient
SRP	Direct (panel model)	Direct (panel model)
Drag	None	None

**Table 9** Low-fidelity dynamics model for SO in GEO

Source	Force model	Torque model
Gravity	Earth point-mass term	Earth gravity gradient
SRP	None	Direct (panel model)
Drag	None	None

The dynamical models differ from the LEO example, as shown in Tables 8 and 9.<sup>9</sup> Of particular note, the low-fidelity torque model is identical to its high-fidelity counterpart. The reason is that the low-fidelity model is only useful if the approximation of the high-fidelity model is adequate: Referencing Eq. 5, for a HAMR SO with little or no angular velocity, the torque contribution to  $\dot{\omega}$  dwarfs the inertia contribution. Further, in the current example scenario, SRP is the dominant source of torque, and therefore the largest driver of  $\dot{\omega}$ . Thus, failure to take into account SRP in the evaluation of the low-fidelity model results in an inaccurate propagation. This contrasts with the 3DOF problem, in which the two-body gravitational acceleration term is orders of magnitude greater than all disturbing accelerations,<sup>10</sup> even for a HAMR SO in GEO.

The evolution of the rotational states is shown in Fig. 7. Despite the small center-of-mass-center-of-pressure offset, the HAMR nature of the SO results in an SRP torque that significantly affects the attitude. Additionally, the small rotational angular momentum of the SO results in slower, less regular rotational motion than was observed in the tumbling LEO scenario, exemplified by the decreased number of complete rotations of the spin and precession angles.

The computational expense of the low-fidelity dynamics model relative to the high-fidelity model degrades the efficiency of the variable-fidelity GLIRK propagator, as shown in Fig. 8 for a three-orbit propagation. This result is confirmed by Fig. 9, as well. The effect of parallelization is also diminished in this scenario because the high-fidelity dynamics model is significantly less expensive to evaluate than that of the LEO scenario due to the truncated geopotential and absence of aerodynamic forces. Thus, for the GEO scenario, a smaller fraction of the runtime is spent evaluating the dynamical model, and the detrimental impact of parallelization overhead is more pronounced.

The consequences of the SO physical characteristics and dynamics model on the performance of the eight-stage GLIRK propagator relative to RKDP8(7) and LSODE are shown in Fig. 10. Unlike in the LEO scenario, the variable-fidelity GLIRK propagator is only competitive with the other propagators for very high-accuracy applications, even when implemented in parallel. Also unlike the LEO scenario, RKDP8(7) is nearly as efficient as LSODE, due to two factors: (1) the ratio of the number of steps taken by RKDP8(7) to the number of steps taken by LSODE decreases from the LEO scenario to the GEO scenario; and (2) a lower percentage

<sup>9</sup>Note that the spherical harmonics formulation of the geopotential is used because it is more efficient than the interpolation model for the low degree and order used for GEO propagation.

<sup>10</sup>Excepting possibly the effect of aerodynamic acceleration on an SO in an extremely low orbit.

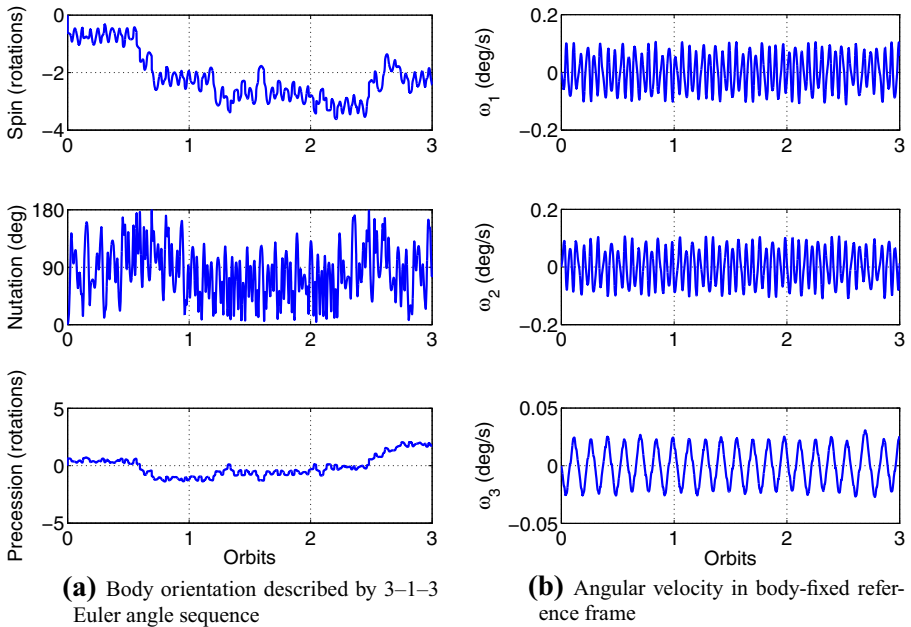


Fig. 7 Rotational state evolution for three-orbit GEO scenario

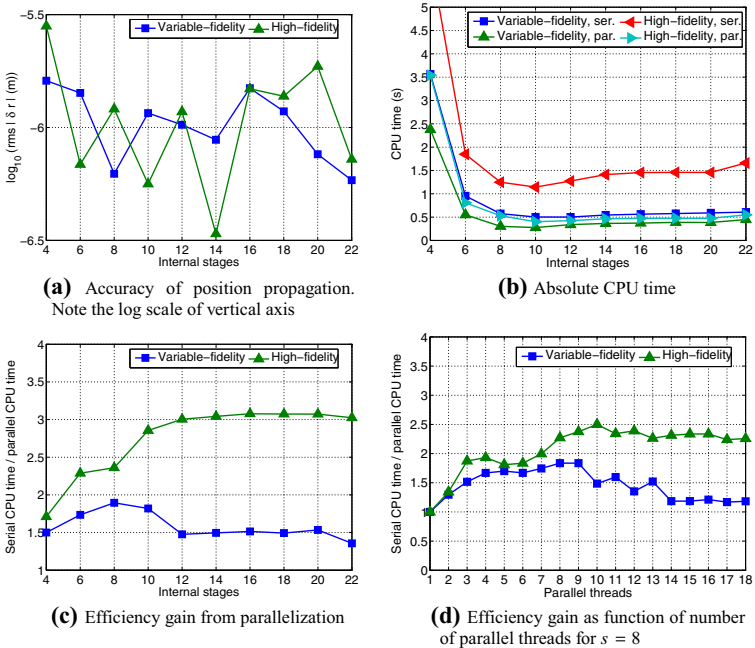
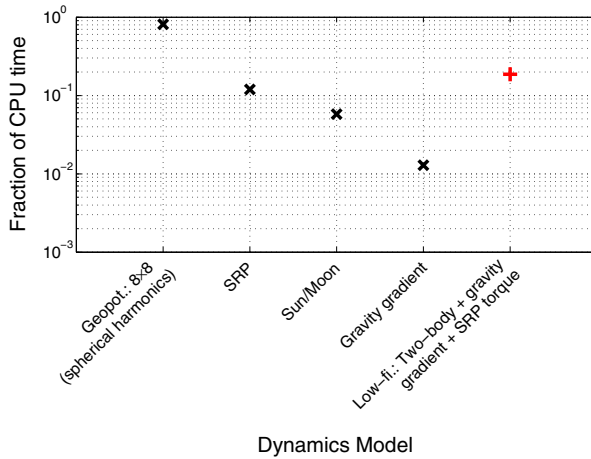
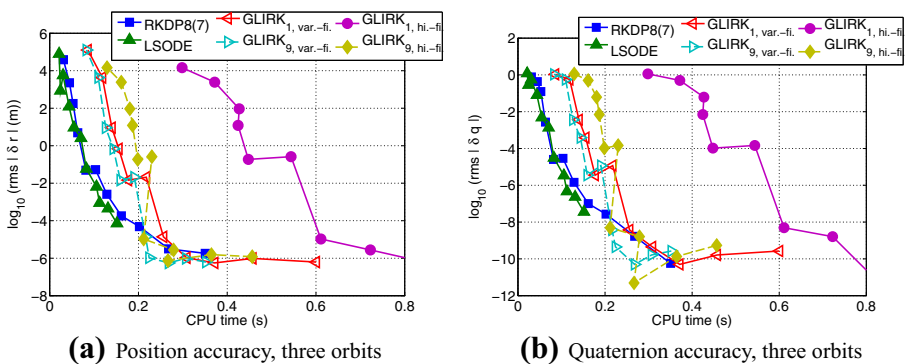


Fig. 8 Performance of GLIRK propagator as a function of number of stages  $s$  using both variable-fidelity and high-fidelity-only dynamics models. Unless otherwise specified, the number of threads made available for parallelization is equal to  $(s + 1)$

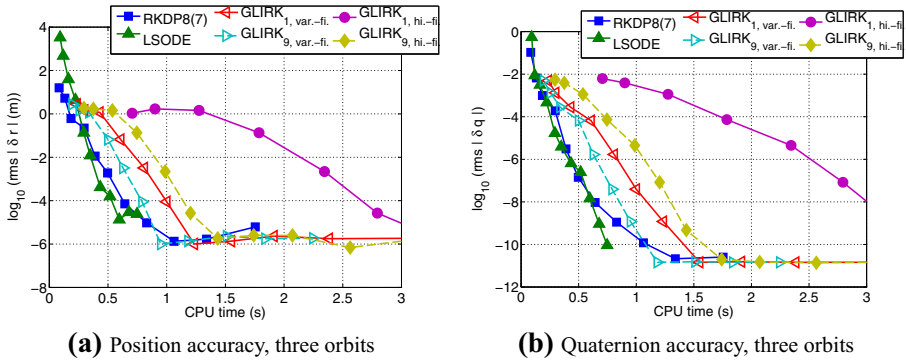


**Fig. 9** Fraction of total high-fidelity dynamics model CPU time spent evaluating each model constituent for three-orbit GEO scenario; alternative low-fidelity dynamics model time shown for comparison. (Low-fidelity model time is not included in summation used to compute total time). Note logarithmic scale of vertical axis

of propagation steps are rejected for the GEO scenario, especially for RKDP8(7) at tight tolerances. Rejected steps are more costly for RK methods than for linear multi-step methods when dynamics function evaluations dominate overall runtime because RK methods generally require significantly more dynamical function evaluations per step (in this case, 13 for RKDP8(7) vs. 1–2 for LSODE). The smaller percentage of rejected steps is likely due to the more uniformly changing dynamic environment of the slowly rotating GEO scenario. For example, there is no torque due to aerodynamic drag, and the torques caused by the gravity gradient and SRP change more slowly



**Fig. 10** Accuracy vs. CPU time for propagation of non-tumbling HAMR SO in GEO; relative LTE tolerance of propagators is varied from  $10^{-5}$  –  $10^{-15}$ . Eight-stage GLIRK propagator is used; numeric subscript in legend represents number of parallel threads used in computation; “var.-fi.” indicates use of variable-fidelity dynamics model, while “hi.-fi.” indicates use of high-fidelity dynamics model only. Note the log scale of vertical axes



**Fig. 11** Accuracy vs. CPU time for propagation of tumbling HAMR SO in GEO; relative LTE tolerance of propagators is varied from  $10^{-5} - 10^{-15}$ . Eight-stage GLIRK propagator is used; numeric subscript in legend represents number of parallel threads used in computation; “var.-fi.” indicates use of variable-fidelity dynamics model, while “hi.-fi.” indicates use of high-fidelity dynamics model only. Note the log scale of vertical axes

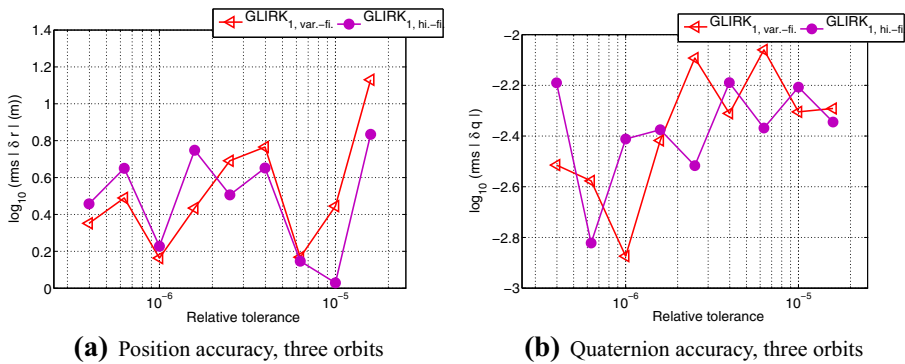
than in the LEO environment due to the slower rotation rate and increased orbital period of the SO. Thus, a step size  $h_{i+1}$  is more likely to differ significantly from the step size for the previous step  $h_i$  in the LEO propagation than in the GEO propagation. When a large difference between consecutive step sizes occurs, the predicted value for  $h_{i+1}$  may be inaccurate, resulting in either (1) a failed step (if the predicted  $h_{i+1}$  is too large) or (2) an inefficient step (if the predicted  $h_{i+1}$  is too small).

The scenario is also simulated assuming an initially tumbling plate, with  $\omega_0 = [0.573, 0.573, 0.573]^T$  deg/s; results are shown in Fig. 11. As in the LEO scenario, propagation of a tumbling body requires more absolute CPU time than an initially non-rotating body due to the higher frequencies of the rotational motion. The performance of the GLIRK propagator is improved in the tumbling scenario (compared to the non-tumbling GEO scenario) at loose LTE tolerances, though the efficiency still does not significantly exceed that of either LSODE or RKDP8(7) for a given RMS error. At more stringent tolerances, the relative performances of the propagators are similar to those observed in the non-tumbling case.

For any propagator, increasing the LTE tolerance beyond a limiting value<sup>11</sup> results in erratic behavior in an accuracy vs. CPU time plot.<sup>12</sup> This behavior can produce ranges of tolerance values in which the change in tolerance (and the resulting change in dynamics function evaluations) does not reliably correlate with a change in the accuracy of the propagation [40]. For this particular scenario, such a region exists for the GLIRK propagator for relative tolerances greater than approximately  $10^{-6.2}$ . As a result, the RMS state errors do not vary predictably at loose tolerances for GLIRK in Fig. 11, and errors observed for a tolerance of  $10^{-5}$  are similar to those observed for

<sup>11</sup>This value is problem- and integrator-dependent.

<sup>12</sup>It is noted that erratic behavior for an implicit method can also be caused by frequent divergence of the iterative procedure used to solve the RK update equations, but divergence does not occur over the range of tolerances displayed in the figures in this paper.



**Fig. 12** Accuracy vs. relative LTE tolerance for propagation of tumbling HAMR SO in GEO; relative LTE tolerance of propagators is varied from  $10^{-4.8}$  –  $10^{-6.4}$ . Eight-stage GLIRK propagator is used; numeric subscript in legend represents number of parallel threads used in computation; “var.-fi.” indicates use of variable-fidelity dynamics model, while “hi.-fi.” indicates use of high-fidelity dynamics model only. Note the log scale of horizontal and vertical axes

a tolerance of  $10^{-7}$  – even though the former case requires fewer dynamics function evaluations. Figure 12 details this result by directly displaying RMS errors as functions of relative LTE tolerance for serial GLIRK propagations only, with the relative tolerance varied from  $10^{-4.8}$  –  $10^{-6.4}$ .

Just as for the LEO scenario, the relative performance of the GLIRK propagator would improve if the dynamics were more strenuous: For example, SRP may be calculated using bidirectional reflectance distribution functions, SO self-shadowing effects may be relevant, and Earth-albedo and Earth-infrared radiation pressure and thermal radiation pressure may also be taken into account. The addition of any of these factors increases the performance gains available from parallelization. Also, because of the dominance of SRP torque on HAMR SOs in GEO, Earth and thermal radiation pressure torque may likely be ignored in the low-fidelity dynamics model,<sup>13</sup> further improving efficiency.

## Conclusion

A variable-step-size Gauss-Legendre implicit Runge-Kutta (GLIRK) ODE solver is applied to the fully coupled six-degree-of-freedom (6DOF) propagation of a space object in Earth orbit. Local truncation error is estimated using an inexpensive, nearly embedded lower-order method. The majority of dynamics model evaluations are replaced with fast, low-fidelity alternatives, decreasing CPU runtime without compromising propagation accuracy. Additional unnecessary high-fidelity evaluations are eliminated by estimating the local truncation error prior to full convergence of the iterative solver at each propagation step. As these customizations suggest, GLIRK

<sup>13</sup>Unless the SO is in the Earth’s shadow.



propagators – for both 3DOF and 6DOF astrodynamics applications – require careful tuning and problem-specific insight to maximize efficiency. However, the benefits of the GLIRK method are shown to be strong for certain common dynamical regimes, particularly given the growing ubiquity of parallel computing resources.

The performance of the GLIRK propagator is examined using 4–22 internal stages, evaluated either serially or in parallel using a multicore CPU with up to 23 threads and OpenMP. (One thread is reserved for an extra dynamics function evaluation used in the estimation of the local truncation error.) Eight-to-twelve-stage methods are found to produce greatest efficiency for a serial implementation. Speed improvements realizable from a parallel implementation depend on parallelization overhead and dynamical model complexity; for typical dynamics models, the efficiency gains from using more than 8–12 stages and threads are observed to be marginal.

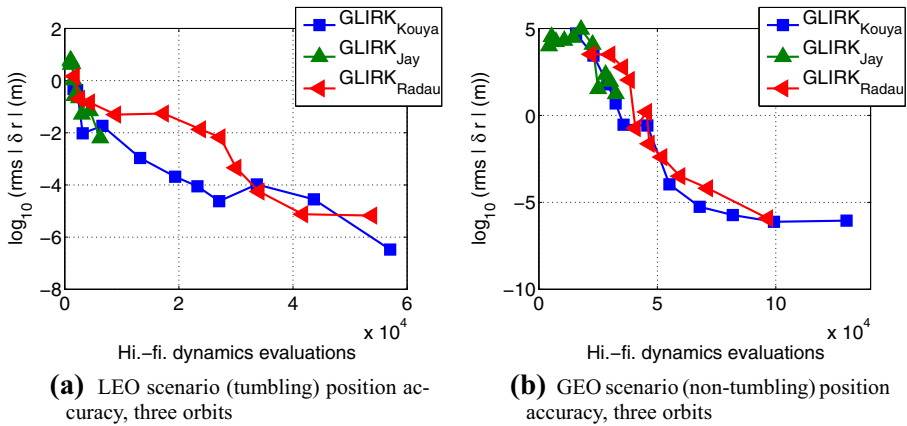
The use of variable-fidelity dynamics combined with the parallelizability of IRK methods means that the propagator is most efficient when the high-fidelity dynamics model is computationally intensive and a much less expensive, yet moderately accurate, low-fidelity model is available. For example, for a sample space object in low Earth orbit, the serially implemented, variable-fidelity, eight-stage GLIRK propagator is competitive with standard linear multistep and explicit Runge-Kutta propagators, though the linear multistep method may be the most efficient of the three tested options. However, when parallelized, the GLIRK method is shown to significantly outperform the serial alternatives.

On the other hand, the efficiency of the GLIRK method is degraded if the high-fidelity dynamics model is inexpensive to evaluate, and/or the cost of the low-fidelity model relative to the high-fidelity model is high. These conditions lessen the impact of variable-fidelity dynamics models and increase the visibility of parallelization overhead: For the propagation of a high-area-to-mass-ratio space object in geosynchronous orbit, the parallelized GLIRK method is only competitive with the linear multistep propagator at tight integration tolerances. In such scenarios – in which the effectiveness of GLIRK customizations is limited – a variable-step linear multistep method is likely to outperform the GLIRK method, particularly in a serial computing environment. The example simulations presented in this study are representative of the two ends of the performance spectrum for GLIRK methods applied to 6DOF propagation. However, when approaching a specific propagation scenario, a practitioner should carefully evaluate the factors discussed here to determine the best choice of integration method.

**Acknowledgments** This work was funded, in part, by a Phase II SBIR from the Air Force Research Laboratory, contract FA9453-14-C-0295, under a subcontract from Emergent Space Technologies, Inc.

## Appendix A: Local Truncation Error Estimation

Position accuracy as a function of high-fidelity dynamics function evaluations (a proxy for CPU time) for three methods of calculating the “comparison solution” for estimating the local truncation error (LTE) is shown in Fig. 13. The figure displays results for three-orbit propagations of the tumbling LEO scenario (Fig. 13a) and



**Fig. 13** Position accuracy as a function of high-fidelity dynamics function evaluations for three LTE calculation methods; relative LTE tolerance of propagators is varied from  $10^{-5}$  –  $10^{-15}$ . The Kouya method is used to produce the results given in this paper

non-tumbling GEO scenario (Fig. 13b) introduced in this paper. The variable-fidelity dynamics strategy is used, and the GLIRK solution uses eight stages. The “Kouya” method, which is based on a nearly embedded solution of order  $s$ , is the strategy used to produce the results given in this paper [28]. The “Jay” method uses an internal tolerance parameter to generate a less conservative estimate of the LTE based on the order- $s$  comparison solution [25]. The “Radau” method produces a comparison solution using a non-embedded  $s$ -stage Radau-IA propagation, which produces a solution of order  $(2s - 1)$  [19]. For the results presented in Fig. 13, it is assumed that the initial guess for the Radau-IA solution is accurate enough that only one high-fidelity dynamics function evaluation per stage is required to achieve convergence.

The inexpensiveness of the nearly embedded Kouya method generally produces a more efficient propagation than the Radau method for a given accuracy when using variable-fidelity dynamics models, even though the Radau method generally uses larger step sizes. On the other hand, the Jay method uses very few function evaluations, but the internal tolerance tends to produce such large step sizes that global error control is poor compared to the other two methods.

## References

1. Continuing kepler’s quest: assessing air force space command’s astrodynamics standards. Tech. rep., national research council of the national academies, Washington, D. C., pp. 19–43 (2012)
2. OpenMP application program interface, version 4.0. OpenMP architecture review board (2013)
3. Wolfram language & system documentation center: Implicitrungekutta method for NDSolve. <https://reference.wolfram.com/language/tutorial/NDSolveImplicitRungeKutta.html>. Accessed 18 Nov. 2015 (2015)
4. Aristoff, J.: Method and system for propagating the state of an object and its uncertainty. US Patent 8,909, 588 (2014)
5. Aristoff, J.M., Poore, A.B.: Implicit Runge-Kutta methods for orbit propagation. In: AAS/AIAA Astrodynamics Specialist Conference, Minneapolis, MN (2012)

6. Aristoff, J.M., Horwood, J.T., Poore, A.B.: Orbit and uncertainty propagation: a comparison of Gauss-Legendre-, dormand-Prince-, and Chebyshev-Picard-based approaches. *Celest. Mech. Dyn. Astr.* **118**(1), 13–28 (2014a)
7. Aristoff, J.M., Horwood, J.T., Singh, N., Poore, A.B.: Error estimation and control for efficient and reliable orbit (and uncertainty) propagation. In: *AAS/AIAA Space Flight Mechanics Meeting*, Santa Fe, NM (2014b)
8. Aristoff, J.M., Horwood, J.T., Poore, A.B.: Implicit-Runge-Kutta-based methods for fast, precise, and scalable uncertainty propagation. *Celest. Mech. Dyn. Astr.* **122**(2), 169–182 (2015)
9. Arora, N., Russell, R.P.: A fast, accurate, and smooth planetary ephemeris retrieval system. *Celest. Mech. Dyn. Astr.* **108**(2), 107–124 (2010)
10. Arora, N., Vittaldev, V., Russell, R.P.: Parallel computation of trajectories using graphics processing units and interpolated gravity models. *J. Guid. Control. Dynam.* **38**(8), 1345–1355 (2015)
11. Bai, X., Junkins, J.L.: Modified chebyshev-Picard iteration methods for orbit propagation. *J. Astronaut. Sci.* **58**(4), 583–613 (2011)
12. Bate, R.R., Mueller, D.D., White, J.E.: *Fundamentals of Astrodynamics*, pp. 177–212. Dover Publications, Inc, New York (1971)
13. Berry, M., Healy, L.: Implementation of gauss-Jackson integration for orbit propagation. *J. Astronaut. Sci.* **52**(3), 331–357 (2004)
14. Bradley, B.K., Jones, B.A., Beylkin, G., Sandberg, K., Axelrad, P.: Bandlimited implicit Runge-Kutta integration for astrodynamics. *Celest. Mech. Dyn. Astr.* **119**(2), 143–168 (2014)
15. Doornbos, E.: *Thermospheric Density and Wind Determination from Satellite Dynamics*. Ph.D Dissertation, TU Delft (2011)
16. Früh, C., Jah, M.K.: Attitude and orbit propagation of high area-to-mass ratio (HAMR) objects using a semi-coupled approach. *J. Astronaut. Sci.* **60**(1), 32–50 (2014)
17. Fukushima, T.: Simple, regular, and efficient numerical integration of rotational motion. *Astron. J.* **135**(6), 2298–2322 (2008)
18. Hairer, E., Wanner, G.: Stiff differential equations solved by Radau methods. *J. Comput. Appl. Math.* **111**, 93–111 (1999)
19. Hairer, E., Lubich, C., Wanner, G.: *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, 2nd edn. No. 31 in Springer Series in Computational Mathematics, pp. 30–35, 101, 147, 330–332. Springer, Berlin (2006)
20. Hairer, E., Norsett, S.P., Wanner, G.: *Solving Ordinary Differential Equations 1: Nonstiff Problems*, 3rd edn. No. 8 in Springer Series in Computational Mathematics, pp. 196–200. Springer, Berlin (2008)
21. Hatten, N., Russell, R.P.: Parallel implicit Runge-Kutta methods applied to coupled orbit/attitude propagation. In: *26th AAS/AIAA Space Flight Mechanics Meeting*, Napa, CA (2016a)
22. Hatten, N., Russell, R.P.: A smooth and robust Harris-Priester atmospheric density model. In: *26th AAS/AIAA Space Flight Mechanics Meeting*, Napa, CA (2016b)
23. van der Houwen, P.J., Sommeijer, D.P.: Parallel iteration of high-order Runge-Kutta methods with stepsize control. *J. Comput. Appl. Math.* **29**(1), 111–127 (1990)
24. Jackson, J.: Note on the numerical integration of  $d^2x/dt^2 = f(x, t)$ . *Mon. Not. R. Astron. Soc.* **84**, 602–606 (1924)
25. Jay, L.O.: Structure preservation for constrained dynamics with super partitioned additive Runge-Kutta methods. *SIAM J. Sci. Comput.* **20**(2), 416–446 (1998)
26. Jones, B.A.: Orbit propagation using Gauss-Legendre collocation. In: *AAS/AIAA Astrodynamics Specialist Conference*, Minneapolis, MN (2012)
27. Jones, B.A., Anderson, R.L.: A survey of symplectic and collocation integration methods for orbit propagation. In: *22nd Annual AIAA/AAS Space Flight Mechanics Meeting*, Charleston, SC (2012)
28. Kouya, T.: Practical implementation of high-order multiple precision fully implicit runge-Kutta methods with step size control using embedded formula. *Int. J. Numer. Meth. Applic.* **9**(2), 85–108 (2013)
29. Krogh, F.T.: On testing a subroutine for the numerical integration of ordinary differential equations. *J. ACM* **20**(4), 545–562 (1973)
30. Le Fevre, C., Morand, V., Delpéch, M., Gazzino, C., Henriquel, Y.: Integration of coupled orbit and attitude dynamics and impact on orbital evolution of space debris. In: *AAS/AIAA Space Flight Mechanics Meeting*, Williamsburg, VA (2015)
31. Long, A.C., Cappellari Jr., J.O., Velez, C.E., Fuchs, A.J.: *Goddard Trajectory Determination System (GTDS) Mathematical Theory Revision 1*. NASA, Goddard Space Flight Center, MD (1989)

32. Macomber, B.D.: Enhancements to the Chebyshev-Picard Iteration Efficiency for Generally Perturbed Orbits and Constrained Dynamical Systems. Ph.D. Dissertation, Texas A&M University (2015)
33. Markley, F.L.: *Spacecraft Attitude Determination and Control*, pp. 510–522. Kluwer Academic Publishers, Dordrecht (1978)
34. van der Merwe, R.: *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. Ph.D. Dissertation, Oregon Health & Science University (2004)
35. Montenbruck, O.: Numerical integration methods for orbital motion. *Celest. Mech. Dyn. Astron.* **53**, 59–69 (1992)
36. Ocampo, C.A.: An architecture for a generalized spacecraft trajectory design and optimization system. In: *International Conference on Libration Point Missions and Applications*, Girona, Spain (2002)
37. Prince, P.J., Dormand, J.R.: High order embedded runge-Kutta formulae. *J. Comput. Appl. Math.* **7**(1), 67–75 (1981)
38. Radhakrishnan, K., Hindmarsh, A.: Description and use of LSODE, the Livermore Solver for Ordinary Differential Equations. Tech. Rep. UCRL-ID-113855 Lawrence Livermore National Laboratory, Livermore, CA (1993)
39. Shampine, L.F., Reichelt, M.W.: The MATLAB ODE suite. *SIAM J. Sci. Comput.* **18**(1), 1–22 (1997)
40. Urrutxua, H., Bombardelli, C., Roa, J.: Gonzalo JL. In: *AAS/AIAA Space Flight Mechanics Conference*, Napa, CA. Quantification of the performance of numerical orbit propagators (2016)
41. Wetterer, C.J., Crassidis, J.L., Linares, R., Kelecy, T.M., Ziebart, M.K., Jah, M.K., Cefola, P.J.: Refining space object radiation pressure modeling with bidirectional reflectance distribution functions. *J. Guid. Control. Dynam.* **37**(1), 185–196 (2014)
42. Woodburn, J., Tanygin, S.: Efficient numerical integration of coupled orbit and attitude trajectories using an Encke type correction algorithm. In: *AAS/AIAA Astrodynamics Specialist Conference*, Quebec City (2001)
43. van Zon, R., Schofield, J.: Numerical implementation of the exact dynamics of free rigid bodies. *J. Comput. Phys.* **225**, 145–164 (2007)