This paper describes a new method of numerical integration and compares its efficiency in propagating orbits to existing techniques commonly used in astrodynamics. By using generalized Gaussian quadratures for bandlimited functions, the implicit Runge-Kutta scheme (a collocation method) allows us to use significantly fewer force function evaluations than other integrators. The new method computes the solution on a large time interval, leading to a different approach to force evaluation. In particular, it is sufficient to use a low-fidelity force model for most of the iterations, thus minimizing the use of a high-fidelity force model. Our goal is to develop a numerical integration technique that is faster than current methods in an effort to address the expected increase of the space catalog due to improvements in tracking capabilities.

INTRODUCTION

This paper presents a new numerical integration technique, developed by Beylkin and Sandberg at the University of Colorado, and compares its efficiency in propagating orbits to existing techniques commonly used in astrodynamics. The new scheme, dubbed Bandlimited Collocation Implicit Runge-Kutta (BLC-IRK) method, is an Implicit Runge-Kutta (IRK) method based on collocation, where generalized Gaussian quadratures for bandlimited functions are used instead of the traditional orthogonal polynomials. Implicit Runge-Kutta methods have been constructed for a variety of polynomial based quadratures, such as Gauss-Legendre, Gauss-Lobatto, and Chebyshev quadratures. Using quadratures based on exponentials instead of polynomials significantly affects the resulting scheme.

While a Runge-Kutta scheme with the Gauss-Legendre nodes provides an excellent discretization of a system of ODEs, using a large number of nodes per time interval is not advisable. The reason is that the nodes of the Gauss-Legendre quadratures (as well as any other polynomial-based Gaussian quadratures) accumulate rapidly towards the end points of the interval. For such quadratures, the ratio of the distances between the nodes near the end of the interval and those in the middle, is asymptotically inversely proportionate to their number. On the other hand, the node accumulation for the generalized Gaussian quadratures for bandlimited functions is moderate and the ratio of distances is asymptotically a constant that depends only on the desired accuracy.

*Graduate Assistant, Aerospace Engineering Sciences, University of Colorado at Boulder, 431 UCB, Boulder, CO, 80309. E-mail: ben.bradley@colorado.edu
†Research Associate, Colorado Center for Astrodynamics Research, University of Colorado at Boulder, 431 UCB, Boulder, CO, 80309.
‡Professor, Applied Mathematics, University of Colorado at Boulder, 526 UCB, Boulder, CO, 80309.
§Professor, Aerospace Engineering Sciences, University of Colorado at Boulder, 431 UCB, Boulder, CO, 80309.
**Reference 1 is currently in preprint. Please contact B.K. Bradley to request a draft of the manuscript.
The consequence of this fact is that the solution may be sought on a large time interval where we use a large number of nodes. This, in turn, changes the way forces are evaluated yielding an efficient algorithm in spite of the implicit and, hence, iterative nature of the scheme. We reduce the computational cost by employing a low-fidelity force model for a majority of the required force evaluations. Additionally, the use of generalized Gaussian nodes for bandlimited functions minimizes the total number of nodes required to achieve a given accuracy.\textsuperscript{1,2}

The reduced sampling requirements and significantly better resulting differentiation schemes have been successfully used in problems of wave propagation.\textsuperscript{8,9} For example, a time domain solver in Reference 8 for the wave equation uses bandlimited approximations and yields about 12 digits of accuracy with only 3 nodes per wavelength over large propagation distances. Unlike problems of wave propagation, where solutions are always well approximated via bandlimited functions, solutions of some ODEs may, in fact, be polynomials or other functions that do not have an efficient approximation via bandlimited exponentials. However, as long as the solution is well approximated by bandlimited functions, using appropriate generalized Gaussian quadratures is effective.

A basis for bandlimited functions, the so-called Prolate Spheroidal Wave Functions (PSWFs) of classical mathematical physics, was introduced by Slepian et al. in a series of seminal papers.\textsuperscript{10–16} Their goal was to optimize (simultaneously) the localization of functions in the space and Fourier domains and they constructed the eigenfunctions of time-limiting and band-limiting operators and demonstrated that the resulting eigenfunctions are the PSWFs. However, the quadratures for integrating and interpolating the bandlimited functions were constructed only recently.\textsuperscript{2,17} These quadratures are essential for using bandlimited functions in numerical analysis.

Unlike the classical Gaussian quadratures for polynomials which integrate exactly a subspace of polynomials up to a fixed degree, the Gaussian type quadratures for exponentials in Reference 2 use a finite set of nodes in an attempt to integrate an infinite set of functions, namely, $\{e^{ibx}\}_{|b| \leq c}$ on the interval $|x| \leq 1$. While there is no way to accomplish this exactly, these quadratures are constructed so that all exponentials with $|b| \leq c$ are integrated with accuracy of at least $\epsilon$, where $\epsilon$ is arbitrarily small but finite. We note that if the accuracy $\epsilon$ is chosen to be around $10^{-16}$, such quadratures are effectively exact within the double precision of machine arithmetic.

In this paper we compare the performance of the new scheme with the traditional methods used in astrodynamics. We are motivated by the need to improve the computational performance of existing schemes. The growing cloud of spent rocket bodies, defunct satellites, and other debris in Earth orbit is a serious threat to our use of space, particularly in densely populated low-Earth orbits and the orbits within the geosynchronous belt. In 2005, NORAD tracked about 10,000 objects and close approaches were already a common occurrence, taking place hundreds of times each week.\textsuperscript{18} Currently, the public space catalog is approximately 15,000, while the Joint Space Operations Center (JSpOC) maintains a catalog containing over 22,000 objects in Earth orbit that are at least 10 centimeters.\textsuperscript{*} Although conjunction assessment for the entire space catalog is manageable at this time, it will become extremely difficult in the near future. This expected difficulty is due to the planned Space Fence (several ground-based S-band radar sensors) and the JSpOC Mission System (JMS) High Accuracy Catalog (HAC). This new capability is anticipated to increase the space catalog to hundreds of thousands, making the current method for performing orbit determination and conjunction assessment very challenging. Since orbit determination and propagation take up a majority of the computation time, faster numerical integration techniques are considered necessary.

\textsuperscript{*}United States Strategic Command, http://www.stratcom.mil/factsheets/usstratcom_space_control_and_space_surveillance
The intent of this paper is to provide a mathematical overview of the new BLC-IRK integration scheme and compare its efficiency in orbit propagation with other more commonly used techniques. We start by outlining the framework of implicit Runge-Kutta collocation based methods and describe the details of the new scheme. We then consider the advantages of the new framework, the required input parameters, and then compare it to other integration techniques. Three orbit types are used to compare results from four numerical integration techniques (frequently used in the astrodynamics community): Runge-Kutta-Fehlberg 7(8), Dormand-Prince 8(7), Dormand-Prince 5(4), and an 8th-order Gauss-Jackson. A low-Earth orbit, Molniya orbit, and geostationary orbit are propagated for 3 revolutions using a 70x70 gravity field and lunisolar perturbations. We conclude with a summary of the results and recommended future work.

**MATHEMATICAL OVERVIEW**

This section details the mathematical techniques of the new BLC-IRK method as well as the basics of implicit Runge-Kutta and collocation methods to put the new scheme into context. We consider the initial value problem (IVP) for an ordinary differential equation (ODE)

\[ y' = f(t, y), \quad y(0) = y_0, \quad t \geq 0. \]  

(1)

The solution \( y \) at some time \( h \) can then be written as a Picard integral

\[ y(h) = y_0 + \int_0^h f(s, y(s)) \, ds. \]  

(2)

IRK methods are based on using Gaussian type quadratures for discretization of Eq. (2). In general, quadratures approximate integrals

\[ \int_{-1}^1 f(x)W(x) \, dx \approx \sum_{j=1}^M w_j f(\tau_j), \]  

(3)

where \( W(x) \geq 0 \) is the weight, \( \tau_j \) are quadrature nodes, and \( w_j \) are quadrature weights. Given a fixed number of nodes, \( M \), the classical Gaussian quadratures maximize the degree of polynomials for which Eq. (3) is exact. Gauss-Legendre quadratures correspond to the case when \( W(x) = 1 \).

**Implicit Runge-Kutta Schemes**

While explicit Runge-Kutta methods (ERK) are commonly used in astrodynamics problems, the use of implicit Runge-Kutta methods (IRK) is still infrequent. Runge-Kutta methods are single-step methods with \( M \) stages, or nodes, used to solve Eq. (1) and (2) above. The basic form of Runge-Kutta methods using quadratures integrate from time \( t = 0 \) to time \( t = h \)

\[ y(h) = y_0 + \sum_{j=1}^M w_j f(h\tau_j, y(h\tau_j)), \quad \tau \in [0, 1] \]  

(4)

with weights \( \{w_j\}_{j=1}^M \) and nodes \( \{\tau_j\}_{j=1}^M \). With traditional use of Runge-Kutta methods the time interval, \( h \), (or step-size) is small. In the new method, the time interval doesn’t have to be small since the number of nodes, \( M \), may be selected to be large. Ideally, Eq. (4) would be used to solve
for $y(h)$, however, the value of $y(h\tau_j)$ (i.e. value of $y$ at each node) is not known and must be approximated. We use $\xi_i$ as the approximation of $y(h\tau_j)$ and now solve for $y(h)$ using

$$\xi_i = y_0 + \sum_{j=1}^{M} S_{ij} f(h\tau_j, \xi_j)$$

and

$$y(h) = y_0 + \sum_{j=1}^{M} w_j f(h\tau_j, \xi_j)$$

where $S$ is the integration matrix. IRK methods are similar to ERK methods except that the vector functions, $\xi_i$, form a set of nonlinear equations which cannot be solved for explicitly. Thus, an iterative approach must be taken to solve Eq. (5). Several techniques are available, such as fixed-point iteration and Newton iteration. The advantages, disadvantages, and implementation of each method are discussed in Reference 5 and 7.

The quadrature nodes $\{\tau_j\}_{j=1}^{M}$, weights $\{w_j\}_{j=1}^{M}$, and values in the integration matrix $S_{ij}$ are typically displayed in a Butcher table

$$
\begin{array}{c|ccc}
\tau & S & w^T \\
\hline
\tau_1 & S_{1,1} & \cdots & S_{1,M} \\
\tau_2 & S_{2,1} & \cdots & S_{2,M} \\
\vdots & \vdots & \ddots & \vdots \\
\tau_M & S_{M,1} & \cdots & S_{M,M} \\
\end{array}
$$

The implicit property is due to a full integration matrix of size $M \times M$. Explicit methods, in contrast, implement a lower triangular integration matrix with components $S_{ij} = 0$ for $j \geq i$. Note that although we use $\tau$, $w$, and $S$ the variables $c$, $b$, and $A$ have a long tradition of use for representing nodes, weights, and the integration matrix, respectively.

IRK methods have been used sparingly in astrodynamics due to the additional computations required to iteratively solve for the vector functions and the fact that ERK methods are simple to code and are well-documented. Advances in computational power, however, has evened out the implementation of explicit and implicit schemes. IRK methods lend themselves to multi-core computers since the force model evaluation, $f$, at a particular node, is independent of other nodes. Reference 5 contains a summary of methods and references on this topic.

**Collocation**

We now consider another algorithm suited for solving ODEs called collocation. Traditional collocation methods define nodes to be located at the zeros or extrema of a chosen polynomial. It turns
out that the collocation method may be expressed as an IRK method, however, not all Runge-Kutta methods are collocation methods. The benefit of this technique lies in the fact that a continuous solution is described inherently because a continuous polynomial is used to describe the function and node locations. Explicit methods, on the other hand, yield solutions at discrete points in time and require a separate interpolation scheme to compute intermediate solutions. Consider the collocation polynomial, $u(t)$, with constraints

$$
\begin{align*}
  u(0) &= y_0 \\
  \dot{u}(h\tau_j) &= f(h\tau_j, u(h\tau_j))
\end{align*}
$$

where $y(t) = u(t)$. The most commonly used polynomial-based quadratures are Gauss-Legendre and Gauss-Lobatto, although the use of Chebyshev has captured some attention in astrodynamics recently. Gaussian quadrature using polynomials has a long history of use due to tradition, ease of use, and node/order optimality. We introduce interpolating basis functions $\{R_j(t)\}$ with nodes $\{\tau_j\}_{j=1}^M$ to approximate the derivative function, $f$. We choose nodes such that we approximate $f$ to a given accuracy $\epsilon$ on $[0, h]$,

$$
\|f(h\tau, y(h\tau)) - \sum_{j=1}^M f(h\tau_j, y(h\tau_j))R_j(\tau)\| \leq \epsilon, \quad \tau \in [0, 1].
$$

Equation (2) is then rewritten using Eq. (10) as

$$
y(h\tau_i) = y_0 + \sum_{j=1}^M f(h\tau_j, y(h\tau_j)) \int_0^{\tau_i} R_j(s) ds, \quad i = 1, \ldots, M
$$

which can be simplified to

$$
y(h\tau_i) = y_0 + \sum_{j=1}^M S_{ij} f(h\tau_j, y(h\tau_j))
$$

where $S_{ij} = \int_0^{\tau_i} R_j(s) ds$ is the integration matrix. We use $M$ quadrature nodes such that

$$
y(h) = y_0 + \sum_{j=1}^M w_j f(h\tau_j, y(h\tau_j))
$$

yields the solution at time $t = h$. Equations (12) and (13) now form an IRK scheme.

**New Scheme: BLC-IRK**

As stated previously, the new scheme analyzed in this paper is an IRK method (with collocation) that uses generalized Gaussian quadratures for bandlimited functions (exponentials) instead of polynomials. Consult References 2 and 17 for the development of generalized Gaussian quadratures for exponentials. As it is traditional, we construct generalized Gaussian quadratures on the interval $[-1,1]$ (although we use them on $[0,1]$),
\[
\left| \int_{-1}^{1} e^{2ictx} dt - \sum_{j=1}^{M} w_j e^{2ic\tau_j x} \right| < \epsilon^2, \quad x \in [-1, 1]
\] (14)

for an accuracy \(\epsilon > 0\), a bandlimit \(c > 0\), and weights \(w_j > 0\).\(^1\) The nodes \(\tau_j\) depend on the bandlimit and accuracy. These generalized Gaussian quadrature nodes correspond to the zeros of discrete prolate spheroidal wave functions (DPSWFs).\(^2\) Reference 2 shows that by finding quadrature nodes for exponentials with bandlimit \(2c\) and accuracy \(\epsilon^2\), we generate an interpolating basis for bandlimited functions with bandlimit \(c\) and accuracy \(\epsilon\). The interpolating basis functions for bandlimited functions are

\[
R_j(x) = \sum_{l=1}^{M} r_{jl} e^{ic\tau_l x}
\] (15)

for \(j = 1, \ldots, M\) and where

\[
r_{jl} = \sum_{k=1}^{M} w_j \Psi_k(\tau_j) \frac{1}{\eta_k} \Psi_k(\tau_l) w_l.
\] (16)

In Eq. (16), \(\eta_k\) are the eigenvalues and \(\Psi_k(\tau)\) are the eigenvectors of a discretized integral operator for the PSWFs (see References 1 and 2 for more details).

![Comparison of node spacing for 70 nodes of Chebyshev, Gauss-Legendre, and generalized Gaussian quadratures.](image)

The use of quadratures for exponentials has certain advantages over polynomial-based quadratures. It is well known that the nodes of polynomial-based quadrature cluster significantly towards
the ends of each interval as the number of nodes increases. This is to compensate for large interpolation errors that occur near the interval endpoints when using equally spaced nodes and high degree polynomial interpolants (known as the Runge Phenomenon). For this reason, a small number of nodes are typically used to avoid oversampling at the interval boundaries. Nodes of quadratures for exponentials, however, do not accumulate as rapidly at the endpoints as shown by Figure 1 above. Following Reference 8 we define a ratio

$$r(M, \epsilon) = \frac{\tau_2 - \tau_1}{\tau_{\lfloor M/2 \rfloor} - \tau_{\lfloor M/2 \rfloor - 1}}$$

(17)
to represent the extent of node accumulation near the interval endpoints. Since the distance between nodes decreases monotonically towards the end of the interval, Eq. (17) yields a quantitative comparison of quadrature methods. The ratio is the distance between two nodes closest to the interval edge divided by the distance between two nodes in the middle of the interval. Figure 2 displays the behavior of the ratio as a function of the number of nodes for polynomial-based quadratures and quadrature for exponentials.

The ratio for Gauss-Legendre and Chebyshev quadrature nodes asymptotically approaches zero as the number of nodes increase. Again, this is why, traditionally, only a few nodes are used with polynomial-based quadratures. This ratio for nodes of quadratures for exponentials, however, approaches a finite limit. This asymptote is a function of the accuracy, $\epsilon$, to which the quadrature is constructed, as seen in Eq. (14). This characteristic of generalized Gaussian quadratures for bandlimited functions lends itself towards using larger time intervals with a large number of nodes per interval.

Figure 2. Comparison of node accumulation of exponential and polynomial-based quadratures. (a) Generalized Gaussian quadrature for bandlimited exponentials with different interpolation accuracies. Marker dots indicate values for quadratures currently available and used in this study. (b) Polynomial-based quadratures, Gauss-Legendre, Chebyshev, and Gauss-Lobatto. Ratios approach zero as $1/M$. 

7
IMPLEMENTATION AND ANALYSIS OF BLC-IRK

This section describes the input parameters necessary for the BLC-IRK method and demonstrates the effect these parameters have on the accuracy of orbit propagation around Earth. Figure 3 illustrates the concept of intervals and nodes to aid in our discussion.

![Figure 3. Example of nodes and intervals (for illustrative purposes only).](image_url)

The current version of the BLC-IRK method requires 5 parameters to be specified by the user in order to execute the integration. Each parameter is described in the list below. Future work will develop the ability for BLC-IRK to determine appropriate values of each parameter automatically based on the orbit and force model.

- **Accuracy** ($\epsilon$): Accuracy of interpolation to which the generalized Gaussian quadrature is constructed. In the current implementation $\epsilon \approx 10^{-13}$. Currently, the quadratures are computed offline and this is not a user selected parameter. It may be made available to the user in future implementations.

- **Bandlimit** ($c$): For a given accuracy $\epsilon$, the bandlimit determines the number of nodes per interval and vice versa. More nodes per interval equates to a higher bandlimit.

- **Number of Intervals** ($N_I$): A time interval $h$ is similar to a step-size in traditional integration schemes and $N_I = (t_f - t_0)/h$. Each interval contains the same number and placement of nodes. Choice of number of nodes, or bandlimit, will affect the number of intervals required to achieve a certain propagation accuracy, however, number of intervals $N_I$ is still a user input parameter. This is similar to choosing a step-size in other fixed-step integration schemes. As demonstrated later, there is a distinct, optimal $N_I$ for a given number of nodes per interval.

- **Number of Low-Fidelity Force Model Iterations** ($N_1$): The number of evaluations of the low-fidelity force model at each node before the high-fidelity force model is evaluated. Iteration is used to solve for each vector function, $\xi$, placing the solution at each node in a location that is close to its true location.

- **Number of Iterations After Accessing High-Fidelity Model** ($N_2$): The number of evaluations of the low-fidelity force model at each node after the high-fidelity force model has been evaluated once. Each iteration uses the same contribution from the high-fidelity model in combination with the updated low-fidelity information to refine the solution at each node.

Traditionally, evaluation of a high-fidelity force model dominates the computational load of any orbit propagation. The iteration process inside the current version of BLC-IRK has been modified...
from a traditional IRK method to make use of a low-fidelity and a high-fidelity force model to reduce the number of times the high-fidelity force model is evaluated. IRK methods require the use of iteration to solve the nonlinear equations for $\xi$, thus involving several calls to the force model, $f$, at each node. We first use a low-fidelity force model, $f_{\text{low}}$, containing 2-body and $J_2$ effects for the first few iterations to place the solution at each node close to the final value. The high-fidelity force model, $f_{\text{high}}$, is then evaluated once and the difference between the low and high-fidelity model, $\Delta f$, is stored. The high-fidelity force model used in this study is comprised of a 70x70 EGM96 gravity model and lunisolar forces. Drag and solar radiation pressure were omitted from this initial study to simplify the analysis. A second set of low-fidelity force model iterations is then used to finalize the iteration process. During this second set of iterations, $\Delta f$ is added to the low-fidelity evaluation. This improves the solution by using information from the high-fidelity force model without expending computation time evaluating it again. We rely on the assumption that the solution at each node is already close to its final value and that the high-fidelity perturbations do not vary much on this scale. The need for a second high-fidelity evaluation after this second round of iterations is under current investigation and is not used for results in this paper. Algorithm 1 describes the overall process further. Additionally, Reference 5 discusses the general use of iteration in IRK methods and provides a collection of references for more information.

Algorithm 1 Iteration Using Low and High-Fidelity Force Models

Inputs are number of iterations $N_1$ and $N_2$, number of nodes $M$, and low and high-fidelity force models $f_{\text{low}}$ and $f_{\text{high}}$.

Note: This algorithm is to be used for each interval

$$\text{for } i_1 = 1 \rightarrow N_1 \text{ do}$$
$$\text{for } m = 1 \rightarrow M \text{ do}$$
$$\text{Update } \xi_m \text{ by evaluating } f_{\text{low}}^m$$
$$\text{end for}$$
$$\text{end for}$$

$$\text{for } m = 1 \rightarrow M \text{ do}$$
$$\text{Evaluate } f_{\text{high}}^m \text{ and store } \Delta f_m = f_{\text{high}}^m - f_{\text{low}}^m$$
$$\text{end for}$$

$$\text{for } i_2 = 1 \rightarrow N_2 \text{ do}$$
$$\text{for } m = 1 \rightarrow M \text{ do}$$
$$\text{Evaluate } f_{\text{low}}^m$$
$$\text{Update } \xi_m \text{ with } f_{\text{low}}^m + \Delta f_m$$
$$\text{end for}$$
$$\text{end for}$$

The force model evaluation at each node is independent of other stages, allowing for heavy parallelization. Numerous computer processors can be devoted to solving each vector function iteratively, thus speeding up this technique even further. This is a property of all IRK methods, however, BLC-IRK will benefit the most from multiple computer processors due to the large number of nodes per interval. Bai, however, investigates the use of graphics processing units (GPUs) to parallelize a
Chebyshev based collocation method with tens to hundreds of nodes per interval. Future work will include optimizing BLC-IRK for use with multiple cores and comparing evaluation times with other integration techniques.

**Case Study**

This investigation uses three types of orbits to evaluate BLC-IRK and compare its performance to commonly used integrators in the astrodynamics community. A low-Earth orbit (LEO), geostationary orbit (GEO), and a Molniya orbit (MOL) were chosen to investigate different orbital regimes and eccentricities. Table 1 lists the Keplerian orbital elements at epoch (0 January 1st, 2011) for each of the three orbits and includes the perigee altitude, \( h_p \).

<table>
<thead>
<tr>
<th>Name</th>
<th>( a ) (m)</th>
<th>( e )</th>
<th>( i ) (deg)</th>
<th>( \Omega ) (deg)</th>
<th>( \omega ) (deg)</th>
<th>( \nu ) (deg)</th>
<th>( h_p ) (km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LEO</td>
<td>6,730,038.573</td>
<td>0.0008023</td>
<td>35.00002</td>
<td>4.99999</td>
<td>335.04742</td>
<td>19.95260</td>
<td>346.5</td>
</tr>
<tr>
<td>MOL</td>
<td>26,553,376.348</td>
<td>0.7409694</td>
<td>63.40000</td>
<td>330.21416</td>
<td>270.00000</td>
<td>0.0</td>
<td>500.0</td>
</tr>
<tr>
<td>GEO</td>
<td>42,164,118.245</td>
<td>0.0009997</td>
<td>0.01000</td>
<td>27.30363</td>
<td>9.99757</td>
<td>2.29880</td>
<td>35,743.8</td>
</tr>
</tbody>
</table>

For all analyses that follow, results are displayed for propagations lasting 3 orbital revolutions of the orbit in question. The truth trajectory is generated by an 8th-order Gauss-Jackson (GJ8) integration scheme using a 1-second time step. The Gauss-Jackson scheme is a multi-step predictor-corrector method that has been used by U.S. space surveillance centers for orbit propagation for over 50 years and is especially efficient at propagating near-circular orbits. The use of GJ8 with time steps from 1 to 10 seconds produce trajectories that are essentially equivalent, with RMS position errors below 1E-06 meters between them for each orbit. Therefore, we use a 1-second step-size for truth because it yields a dense reference trajectory to compare against.

Evaluating the performance of a numerical integration scheme requires careful consideration of two things: (1) How to generate the truth trajectory, and (2) interpolation of the solution. Berry and Healy investigated several techniques for measuring integration error, specifically, what to use for the truth trajectory when propagating orbits with perturbations. They conclude that step-size halving and higher-order integration both work well for generating truth trajectories when perturbations are present. As stated previously, we use truth trajectories generated by the GJ8 scheme with a fixed step-size of 1 second and compare integration accuracy only. The implementation of GJ8 follows that of Berry and Healy. The use of a small step-size for truth requires us to assume that the use of a small step-size yields a more accurate trajectory and that round-off error is not significantly affecting the solution. As the number of force model evaluations is increased, each integration method we are comparing approaches the generated reference trajectory with differences below 1E-06 meters. This indicates that round-off error is not affecting our results for the accuracy range we are considering, i.e. 1E-06 meters.
The interpolation strategy can have a notable impact on computing the error of an integration method. As depicted in Figure 4 above, we interpolate the truth trajectory at times where we have a solution from the method we are comparing. Interpolating at fixed 30-second intervals, however, has proven to introduce significant error. This is especially true with high-order variable-step integration schemes because they take larger time steps than a lower 4th-order method. Since we are limiting ourselves to only interpolating the truth trajectory (generated with a 1-second time step), error due to interpolation is essentially eliminated. The 3D position error at each data point is

$$\Delta r_i = \|X_{C_i} - X_{T, interp_i}\|$$

where the RMS error for the entire trajectory with $n$ data points is then

$$\varepsilon_{rms} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \Delta r_i^2}.$$ 

A range of values for each BLC-IRK input parameter are used to examine the full range of accuracies in this study. For each orbit type, BLC-IRK is implemented using 1 to 130 intervals, in steps of 2, over the duration of the propagation. We also use 1 to 5 iterations for $N_1$ and $N_2$, resulting in a total of about 2,450 orbit propagations for each orbit type. Additionally, each propagation performed by BLC-IRK is done using inertial Cartesian coordinates and Poincaré orbital elements.

Poincaré orbital elements are a canonical version of the equinoctial element set that retain the property of being non-singular for near-circular and low inclination orbits.\textsuperscript{31–36} Non-singular elements are ideal for orbit propagation since the majority of objects in the space catalog are near-circular and geostationary objects have very low inclinations. The canonical property means that Poincaré orbital elements preserve the symplectic nature of a Hamiltonian system, making them a good choice for use with a symplectic integrator. We use the equinoctial orbital elements as an intermediate step when transforming between inertial Cartesian coordinates and Poincaré orbital elements.\textsuperscript{3,31,33–35,37,38} For propagations performed using Poincaré orbital elements, conversion back to inertial Cartesian coordinates is done prior to the computation of error.
Intervals

First, we look at how the number of intervals affects propagation accuracy. Figure 5 shows the relationship between the number of intervals used per orbit and the RMS 3D position error for the LEO orbit. When a small number of intervals per orbit is used, adding intervals reduces the integration error significantly. There reaches a point, however, where adding more intervals does not reduce the integration error. This accuracy floor is due to $\epsilon$, as seen in Eq. (14), which is set prior to the computation of the integration matrix. A small amount of accuracy is sacrificed for faster evaluation time as $\epsilon$ is increased. This is acceptable when position accuracies below the micron or even centimeter level are not needed or even possible, due to imperfect force models. In operational use, an acceptable choice could be to aim for the knee in the curve, in terms of number of intervals, to ensure sufficiently accurate results while minimizing the number of force model calls. However, determining the location of this knee automatically requires additional analysis due to its dependence on the orbit, force model, and bandlimit.

![Figure 5. RMS values of position errors for propagations of the LEO orbit using a range of number of intervals per orbit. Propagations performed in Cartesian coordinates and a bandlimit of 20.](image)

Bandlimit

As mentioned previously, the bandlimit affects how many nodes are contained in each interval. Table 2 lists several bandlimits and their associated node count. The displayed bandlimits are those that have been used to compute and store integration matrices and are the only options available in the current version of the BLC-IRK software. A higher bandlimit forces the quadrature to better approximate the PSWF, thus requiring more nodes per interval.

Figure 6 illustrates the impact that bandlimit has on the relationship between number of function calls and accuracy. Note that when number of function calls is plotted for the BLC-IRK method, we are plotting the number of high-fidelity force model evaluations. This is justified by the fact that the high-fidelity force model requires several orders of magnitude more mathematical operations than
Table 2. Several bandlimits and the number of nodes per interval they generate.

<table>
<thead>
<tr>
<th>Bandlimit</th>
<th>Nodes Per Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>32</td>
</tr>
<tr>
<td>10</td>
<td>46</td>
</tr>
<tr>
<td>17</td>
<td>64</td>
</tr>
<tr>
<td>20</td>
<td>70</td>
</tr>
<tr>
<td>40</td>
<td>114</td>
</tr>
<tr>
<td>46</td>
<td>128</td>
</tr>
<tr>
<td>80</td>
<td>200</td>
</tr>
</tbody>
</table>

The low-fidelity force model. This is mainly due to the 70x70 spherical harmonic gravity model computation.

The results reveal that the choice of bandlimit does not affect how many high-fidelity force model evaluations it takes to achieve a certain accuracy. We propagated the LEO orbit using each bandlimit setting and a range of intervals from 1 to 130. At first, the fact that bandlimit does not affect the outcome of Figure 6 seems odd. However, the reason is that as the bandlimit increases, the number of intervals (thus total number of nodes) required to achieve a given level of accuracy is reduced, thereby lowering the number of force model evaluations. There may be computational advantages of using one bandlimit over another, however, that is currently on the list for future work.

Figure 6. 3-orbit LEO propagation comparing results using different bandlimits. Number of intervals used range from 1 to 130 and number of first and second sets of iterations range from 1 to 5.
Symplectic Property

As with Gauss-Legendre IRK methods,\textsuperscript{39} BLC-IRK can be formulated as a symplectic integrator.\textsuperscript{1} By imposing several constraints on the nodes and weights of the generalized Gaussian quadrature, the BLC-IRK method becomes symplectic, making it an excellent tool for long-term orbit propagation. The symplectic property is easily demonstrated using an energy-like integral analogous to the Jacobi integral of the Restricted Three-Body Problem. The Jacobi Constant, $K$, is computed by

$$\frac{V^2}{2} - \frac{\mu}{r} - \omega_\oplus \cdot r \times \dot{r} - U'(r, t) = K$$

(20)

where $r$ and $V$ are the orbital radius and inertial velocity of the satellite, respectively, $\omega_\oplus$ is the angular velocity vector of the Earth, and $U'(r, t)$ is the time varying gravitational potential of the Earth (without the point-mass contribution).\textsuperscript{40,41} The Jacobi Constant is an energy-like parameter that, in theory, remains constant over time when integrating a system involving a central gravity field with temporal variations. Numerically, this is not actually achieved though, due to the finite precision of a computer. The relative change in Jacobi Constant compared to its initial value is plotted below in Figure 7 for a 100-day propagation of the LEO orbit using BLC-IRK (top) and a Runge-Kutta-Fehlberg 7(8) method (bottom).\textsuperscript{42}

![Figure 7](image)

Figure 7. Relative change in Jacobi Constant during a 100-day LEO propagation using a 70x70 gravity field only. Top: BLC-IRK propagation using 5500 intervals and a bandlimit of 17. Bottom: Runge-Kutta-Fehlberg 7(8) propagation with a relative tolerance of 1e-14.

BLC-IRK maintains a bounded Jacobi Constant over 100 days while the Runge-Kutta-Fehlberg method fails to maintain the Jacobi Constant over long integration times. If the propagation continued, it is clear that the Jacobi Constant for RKF would continue to increase. It is known that
non-symplectic integrators (such as all ERK methods\textsuperscript{39}) do not maintain a bounded energy or Jacobi Constant due to the accumulation of roundoff error. The ability of the BLC-IRK method to be symplectic is of great benefit to long-term propagations where the accumulation of roundoff error becomes a problem. Integration matrices for BLC-IRK can be generated without the symplectic attribute as well, however, all results for BLC-IRK presented in this paper were generated using the symplectic version of the integration matrix.

**PERFORMANCE COMPARISON**

In this section, we compare the propagation efficiency of BLC-IRK to commonly used integration methods for the three orbits given in Table 1. Three of the four integration methods are explicit Runge-Kutta schemes with step-size control and the fourth is the 8\textsuperscript{th}-order Gauss-Jackson method.

- **Runge-Kutta-Fehlberg 7(8)** (RKF 7(8)13): a 13-stage explicit Runge-Kutta method of order 7 and an embedded method of order 8 used for step-size control developed by Erwin Fehlberg.\textsuperscript{42} The software package Satellite Tool Kit, by Analytical Graphics Inc., uses this as the default integrator (other options are available as well).

- **Dormand & Prince 8(7)** (DOPRI 8(7)13 or RK 8(7)13): similar to the 13-stage RKF 7(8), but uses an 8\textsuperscript{th}-order method for the solution and a 7\textsuperscript{th}-order method for step-size control.\textsuperscript{43}

- **Dormand & Prince 5(4)** (DOPRI 5(4)7 or RK 5(4)7): a 7-stage explicit Runge-Kutta method of order 5 and an embedded method of order 4 used for step-size control.\textsuperscript{44} This integration scheme is available in MATLAB where it is known as ode45.\textsuperscript{45} The integration matrix and weights of DOPRI 5(4) were designed with a beneficial feature called FSAL (first-same-as-last). This means that the final stage evaluation at time $t_n$ is equal to the first stage evaluation at the next time $t_{n+1}$, thus saving one evaluation of the force model per time step.

- **Gauss-Jackson 8\textsuperscript{th}-order** (GJ8): a multi-step predictor-corrector method of 8\textsuperscript{th}-order which uses a fixed step-size.\textsuperscript{4,26,27} This scheme has been used by U.S. Space Surveillance Centers since the 1960’s due to its highly efficient propagation of near-circular orbits.\textsuperscript{4,28}

We compare each integrator based on the number of force model evaluations (function calls) that were used to achieve various levels of RMS 3D position error. It is important to remember that this study evaluates integration error only. We are not considering the separate topic of force model errors. Also note that number of function calls for BLC-IRK is the number of high-fidelity force model evaluations only. Results for the BLC-IRK method are given for the integration using Cartesian inertial coordinates as well as Poincaré orbital elements.\textsuperscript{36,38}

In space surveillance and many other applications, we often choose to sacrifice a little accuracy for reduced computation time. Thus, we desire an integration scheme which achieves a necessary level of accuracy while minimizing the number for force model evaluations and computation time required. Figure 8 contains results for the GEO propagation (see Table 1 for orbital elements) after 3 orbital revolutions. The figure demonstrates the well-known observation that more evaluations of the force model yields more accurate propagations with conventional schemes (until some accuracy floor is reached). Results for RKF 7(8), DOPRI 8(7), and DOPRI 5(4) are shown for propagations using relative tolerances ranging from $1E-08$ to $1E-15$. Relative tolerance is used to adaptively control step-size for these three embedded ERK methods. Results for GJ8 were generated using a wide range of fixed step sizes, again, while a 1-second time step is used as truth.
For the GEO propagation, the BLC-IRK method integrating in Poincaré orbital elements requires 70 function calls while the GJ8 method uses 550 function calls to achieve centimeter level accuracy, nearly a factor of 8 reduction. Even the use of Cartesian coordinates with BLC-IRK performs similarly to the GJ8 and DOPRI 8(7) methods for sub-meter accuracies. The accuracy floor shown for both implementations of the BLC-IRK method is due to the finite precision to which the generalized Gaussian quadratures were constructed to approximate the PSWFs. While this floor is greater than the floor for the other methods, it is still well within force model errors. It is evident that the use of Poincaré elements yields a higher error floor than when Cartesian coordinates are used. This is caused by the linear transformation of accelerations from Cartesian space to Poincaré space where information is inherently lost.

Results of the LEO propagation, shown in Figure 9, demonstrate a significantly different distribution of integration schemes than Figure 8. Results for the other schemes now overlap slightly. This is due to the increased spatial variation in the disturbing gravity field at LEO. Each scheme is required to take small time steps to compensate for the increase in perturbations, resulting in similar propagation accuracies. This fact is the reason that BLC-IRK is able to use drastically fewer function calls, and at the same time, is the reason why the Cartesian and Poincaré results are so similar. In LEO, the difference between the low-fidelity and high-fidelity force model is greater than at GEO, allowing the low-fidelity force model to do most of the work and only require one evaluation of the high-fidelity model at each node. Since the force models are more similar in GEO, we saw in Figure 8 that the other schemes require a similar number of high-fidelity force evaluations as the BLC-IRK method.
Figure 9. Comparison of RMS errors over a 3-orbit LEO propagation.

For LEO, BLC-IRK uses 4 to 5 times fewer force model evaluations than GJ8 at an accuracy of 1E-05 meters. The drastic reduction in number of force model evaluations needed by BLC-IRK compared to the other methods in LEO is very encouraging since the vast majority of tracked objects are in low-Earth orbit. This has the potential to reduce the time needed to propagate the entire space catalog by a significant amount. Furthermore, BLC-IRK can be massively parallelized, using a separate computer processor for each node in an interval.

Figure 10. Comparison of RMS errors over a 3-orbit Molniya propagation.
We now consider the Molniya orbit test case shown above in Figure 10. Note that the Molniya orbit has a very high eccentricity of 0.74. In this regime the variable step-size methods, particularly RKF 7(8) and DOPRI 8(7), show a vast improvement over the GJ8 scheme. This makes intuitive sense since the variable step-size integrators are able to take very large steps near apogee and then shrink back down towards perigee. Alternatively, the fixed-step GJ8 is forced to use a small step-size for the duration of the propagation in order to deal with the high dynamics at perigee. This is also why the BLC-IRK method does not out perform the high-order variable step methods. However, it is interesting that the performance of BLC-IRK is comparable to them.

CONCLUSION AND DISCUSSION

This paper has introduced a new numerical integration scheme to astrodynamics and demonstrated that it is more efficient than commonly used integrators for near-circular orbits, including the 8th-order Gauss-Jackson scheme. The generalized Gaussian quadratures for bandlimited functions yield node spacing that is more efficient than traditional polynomial-based quadrature methods such as Gauss-Legendre, Gauss-Lobatto, and Chebyshev. This allows us to use large time intervals and a large number of nodes per interval without wasting computations near the clustered endpoints as with polynomial-based quadratures.

We also introduced the concept of using low and high-fidelity force models for iterating at each node. This implementation has helped to decrease the number of full force model evaluations. Furthermore, since BLC-IRK is an implicit Runge-Kutta method, it can be parallelized. Parallelization would make an even more significant improvement over the high-performing 8th-order Gauss-Jackson technique, thus becoming an ideal scheme for propagating the growing space catalog. While BLC-IRK performs only slightly better than GJ8 in some orbit regimes, BLC-IRK is a brand new technique, leaving room for additional research and improvement. In contrast, the Gauss-Jackson scheme has been around for many years and has essentially maximized it potential.

ACKNOWLEDGMENT

This research is funded by the National Defense Science and Engineering Graduate Fellowship (NDSEG) and partially by the Air Force Research Lab (AFRL). This research of G.B. was partially supported by AFOSR grants FA9550-07-1-0135 and STTR Phase I grant 1118-001-01. We thank George Born for his comments and advice on the research.

REFERENCES


