Celest Mech Dyn Astr (2014) 119:143–168
DOI 10.1007/s10569-014-9551-x

ORIGINAL ARTICLE

Bandlimited implicit Runge–Kutta integration for Astrodynamics

Ben K. Bradley · Brandon A. Jones · Gregory Beylkin · Kristian Sandberg · Penina Axelrad

Received: 24 May 2013 / Revised: 21 February 2014 / Accepted: 25 April 2014 /
Published online: 22 May 2014

Abstract We describe a new method for numerical integration, dubbed bandlimited collocation implicit Runge–Kutta (BLC-IRK), and compare its efficiency in propagating orbits to existing techniques commonly used in Astrodynamics. The BLC-IRK scheme uses generalized Gaussian quadratures for bandlimited functions. This new method allows us to use significantly fewer force function evaluations than explicit Runge–Kutta schemes. In particular, we use a low-fidelity force model for most of the iterations, thus minimizing the number of high-fidelity force model evaluations. We also investigate the dense output capability of the new scheme, quantifying its accuracy for Earth orbits. We demonstrate that this numerical integration technique is faster than explicit methods of Dormand and Prince 5(4) and 8(7), Runge–Kutta–Fehlberg 7(8), and approaches the efficiency of the 8th-order Gauss–Jackson multistep method. We anticipate a significant acceleration of the scheme in a multiprocessor environment.

Keywords Numerical integration · Implicit Runge–Kutta · Initial value problem · Orbit propagation · Symplectic property

Electronic supplementary material The online version of this article (doi:10.1007/s10569-014-9551-x) contains supplementary material, which is available to authorized users.

B. K. Bradley (✉) · B. A. Jones · P. Axelrad
Colorado Center for Astrodynamics Research, University of Colorado Boulder,
431 UCB, Boulder, CO 80309, USA
e-mail: ben.bradley@colorado.edu

G. Beylkin
Department of Applied Mathematics, University of Colorado Boulder,
526 UCB, Boulder, CO 80309, USA

K. Sandberg
Computational Solutions, Inc., Boulder, CO 80301, USA

1 Introduction

We present a new numerical integration technique, developed by Beylkin and Sandberg at the University of Colorado (Beylkin and Sandberg 2014; Beylkin and Monzón 2002), and compare its performance in propagating orbits to existing techniques commonly used in Astrodynamics. The new scheme, dubbed the bandlimited collocation implicit Runge–Kutta (BLC-IRK) method, is an Implicit Runge–Kutta (IRK) collocation scheme which uses generalized Gaussian quadratures for bandlimited exponentials rather than the classical quadratures for orthogonal polynomials. We note that IRK methods have been constructed for a variety of polynomial based quadratures, such as Gauss–Legendre, Gauss–Lobatto, and Chebyshev (e.g., see discussions in Jones and Anderson 2012; Iserles 2009; Hairer et al. 2002). Among polynomial based IRK collocation schemes, only the scheme based on Gauss–Legendre quadratures achieves the highest order of approximation, is A-stable, and symplectic. The new BLC-IRK scheme is also A-stable and symplectic, achieves any user-selected accuracy and, in addition, allows one to use a large number of nodes within each time interval without the penalty of excessive node concentration near the endpoints of the interval. The properties of BLC-IRK scheme significantly affect the approach to using it in Astrodynamics.

The intent of this paper is to provide a mathematical overview of the new BLC-IRK inte-

The quadrature nodes $\{\tau\}_{=1}$, weights $\{w\}_{=1}$, and entries of the integration matrix are typically displayed in a Butcher tableau,

$$\begin{array}{c|c} \boldsymbol{\tau} & \\ \hline & w \end{array} \tag{6}$$

which expands to

$$\begin{array}{c|ccc} \tau_1 & {}_{1,1} & \cdots & {}_{1,} \\ \tau_2 & {}_{2,1} & \cdots & {}_{2,} \\ \vdots & \vdots & & \vdots \\ \tau & {}_{,1} & \cdots & {}_{,} \\ \hline & w_1 & \cdots & w \end{array} \tag{7}$$

We use τ, w, and , for representing nodes, weights, and the integration matrix and note that the variables , , and have also been used for this purpose in the literature.

In ERK methods, the integration matrix is lower triangular, = 0 for ≥ , and, consequently, such methods are explicit. In IRK methods, the set of nonlinear equations in Eq. 4 has to be solved on each time interval. Several techniques are available, such as fixed-point or Newton iterations (Iserles 2009; Atkinson et al. 2009). The advantages, disadvantages, and implementation of each method are discussed in Jones and Anderson (2012), Hairer et al. (1993, 2002), and Hairer and Wanner (1996).

Historically, IRK methods have been used sparingly in Astrodynamics due to the additional computations required to iteratively solve for the values of the solution at the nodes $\mathbf{y}(\tau)$ and the fact that ERK methods are simple to code, well-documented, and include several adaptive step methods. Advances in computational power and changes in computer architecture, however, have evened out the computational cost of explicit and implicit schemes. IRK methods lend themselves to multi-core computers and graphics processing units (GPUs) since, within a single iteration, the force model evaluation $\boldsymbol{f}$ may be performed simultaneously at all nodes. We refer to Jones and Anderson (2012) for a summary of methods and references on this topic specific to Astrodynamics.

for the user-selected accuracy > 0, bandlimit $2 > 0$, and weights $w > 0$. The nodes τ and weights w depend on the bandlimit and accuracy. In the BLC-IRK method the weight $( ) = 1$ and the nodes correspond to the zeros of discrete prolate spheroidal wave functions (DPSWFs) (Slepian 1978). As it is traditional, generalized Gaussian quadratures are constructed on the interval $[-1, 1]$ although we use them on $[0, 1]$ (with the appropriate linear transformation).

Beylkin and Monzón (2002) show that by finding quadrature nodes for exponentials with bandlimit 2 and accuracy ^{2}, we can generate an interpolating basis for bandlimited functions with bandlimit and accuracy . These interpolating basis functions are defined as

$$( ) = \sum_{=1} \tau \tag{15}$$

for $= 1, \ldots,$ with

$$= \sum_{=1} w \quad (\tau) \frac{1}{\eta} \quad (\tau) w, \tag{16}$$

where the matrix (τ) is obtained by solving an algebraic eigenvalue problem,

$$\sum_{=1} w \quad \tau \tau \quad (\tau) = \eta \quad (\tau), \quad , \quad = 1, \ldots, \quad . \tag{17}$$

Following Beylkin and Monzón (2002), accurate approximations to the first PSWFs are then defined as

$$(\tau) = \frac{1}{\eta} \sum_{=1} w \quad (\tau) \quad \tau \tau, \quad = 1, \ldots, \quad . \tag{18}$$

Given interpolating basis functions (ξ), the elements of the integration matrix for BLC-IRK are then computed as

$$= \int_0^{\tau} (\xi) \mathrm{d}\xi. \tag{19}$$

We note that in Beylkin and Sandberg (2014), the construction of interpolating functions and integration matrix is modified in order to assure that the resulting BLC-IRK method is symplectic.

The quadratures for exponentials offer certain advantages over polynomial-based quadratures. It is well known that the nodes of polynomial-based quadratures cluster significantly towards the ends of each interval as the number of nodes increases (a simple heuristic explanation is that polynomials can grow rapidly toward the end points of an interval causing high node concentration). Nodes of quadratures for exponentials, however, do not accumulate as rapidly at the endpoints.

Typically only a small number of nodes of polynomial-based quadratures are used in IRK methods to avoid oversampling at the interval boundaries (e.g., 2–4 nodes). Following Beylkin and Sandberg (2005), we define a ratio

$$(,) = \frac{\tau_2 - \tau_1}{\tau \quad /2 \quad - \tau \quad /2 -1}, \tag{20}$$

- **Number of nodes per interval (bandlimit)** ( ): For a given accuracy , the number of nodes per interval determines bandlimit and vice versa. More nodes per interval equates to a higher bandlimit.
- **Number of Intervals** ( ): A time interval is similar to a step size in traditional integration schemes where $= (- _0)/$ and denotes the final time of the entire orbit propagation. Each interval contains the same number and placement of nodes (i.e., this is a fixed-step and fixed-order implementation). Choice of number of nodes, or bandlimit, will affect the number of intervals required to achieve a certain propagation accuracy, however, number of intervals is a user-defined input parameter. This is similar to choosing a step size in fixed-step integration schemes. As demonstrated later, there is a distinct, optimal for a given number of nodes per interval.
- **Number of Low-Fidelity Force Model Iterations** ($_1$): 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, $\boldsymbol{\xi}$, placing the solution at each node in a location that is close to its true location.
- **Number of Iterations After Accessing High-Fidelity Model** ($_2$): The number of evaluations of the low-fidelity force model at each node after the high-fidelity force

Algorithm 1 Iteration Using Low- and High-Fidelity Force Models

Inputs are number of iterations $_1$ and $_2$, number of nodes , and low- and high-fidelity force models f_{low} and f_{high}.

for $_1 = 1 \rightarrow \ _1$ **do**
  for $= 1 \rightarrow$ **do**
    Update ξ by evaluating f_{low}
  end for
end for

for $= 1 \rightarrow$ **do**
  Evaluate f_{high} and store $f = f_{\text{high}} - f_{\text{low}}$
end for

for $_2 = 1 \rightarrow \ _2$ **do**
  for $= 1 \rightarrow$ **do**
    Evaluate f_{low}
    Update ξ with $f_{\text{low}} + f$
  end for
end for

for $= 1 \rightarrow$ **do**
  Update ξ by evaluating f_{high}
end for

Hairer et al. 2002 and Jones 2012) as well as adaptive step explicit Runge–Kutta schemes (see e.g., Prince and Dormand 1981). This paper specifies each iteration count in an effort to illustrate the low-/high-fidelity force model use. As demonstrated in the results, this method proves sufficient, but a more user-friendly interface may be desirable.

The force model evaluation may be accelerated using multi-core processors. While this is a property of all IRK methods, BLC-IRK will benefit the most from parallelization due to the large number of nodes per interval. Future work will include optimizing BLC-IRK for use with multiple cores and comparing evaluation times with other integration techniques (see e.g., Bai (2010) and Bai and Junkins (2011a) investigating the use of GPUs to parallelize a Chebyshev-based collocation method (MCPI) with tens to hundreds of nodes per interval).

3.1 Case study description

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^{h} January 1st, 2011) for each of the three orbits and includes the perigee altitude, .

A range of values for each BLC-IRK input parameter are used to examine the full range of accuracies. For each orbit type, BLC-IRK is implemented using 1–130 intervals over the duration of the propagation as well as 1–3 iterations for both $_1$ and $_2$. For all analyses that follow, results are displayed for propagations lasting 3 orbital revolutions of the orbit in

Table 1 Initial osculating Keplerian orbital elements and perigee altitude of each orbit investigated in this study

Name	(m)		(°)	(°)	ω (°)	ν (°)	(km)
LEO	6,730,038.57	0.000802	35.00	5.00	335.05	19.95	346.5
MOL	26,553,376.35	0.740969	63.40	330.21	270.00	0.00	500.0
GEO	42,164,118.25	0.000999	0.01	27.30	10.00	2.30	35,743.8

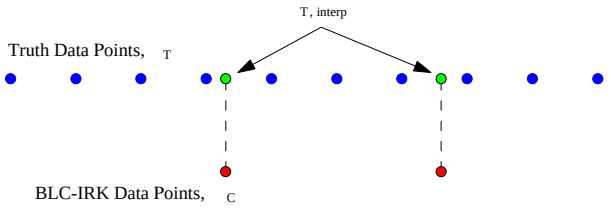

Fig. 3

Fig. 4 RMS values of position errors for propagations of the LEO, GEO, and MOL orbits

evaluations. This is justified by the fact that the high-fidelity force model requires several orders of magnitude more mathematical operations than the low-fidelity force model. This is mainly due to the high degree and order 70 × 70 spherical harmonic gravity model computation.

The results reveal that the choice of node count does not affect how many high-fidelity force model evaluations are necessary to achieve a given accuracy. At first, the fact that the number of nodes does not affect the outcome of Fig. 5a seems odd. However, this feature is actually a byproduct of the node accumulation ratio of the generalized Gaussian quadratures illustrated in Fig. 1a. Since the ratio asymptotically approaches a constant greater than zero, additional force model evaluations are not wasted towards the interval endpoints as with polynomial-based quadratures. As nodes are added, the number of intervals required to achieve a given level of accuracy is reduced, thereby lowering the number of force model evaluations. This point is illustrated in Fig. 5b. Jones (2012) demonstrates this weakness of polynomial-based quadrature schemes by showing the diminishing return of adding nodes in a GL-IRK scheme. As nodes are added, there comes a point when the number of force model evaluations necessary to achieve the certain precision starts increasing. Therefore, BLC-IRK will benefit from parallelization even more than a polynomial-based scheme such as GL-IRK since additional nodes (and thus processors) may be added without the same diminishing return.

3.4 Symplectic property

As with GL-IRK methods (Sanz-Serna 1988), the BLC-IRK method is symplectic (Beylkin and Sandberg 2014). By imposing constraints on the integration matrix and weights of the generalized Gaussian quadratures, the BLC-IRK method becomes symplectic, making it an excellent tool for long-term orbit propagation. Specifically, in order to be symplectic a Runge–Kutta method must satisfy the conditions (Sanz-Serna 1988)

$$w_{\quad} + w_{\quad} - w_{\quad} w_{\quad} = 0, \qquad , \quad = 1, \ldots , \quad . \tag{22}$$

We demonstrate the symplectic property of the BLC-IRK method by using an energy-like integral analogous to the Jacobi integral of the Restricted Three-Body Problem. The Jacobi constant, , is computed by

$$\frac{2}{2} - \frac{\mu}{-} - \quad (\boldsymbol{R}) = \quad = \quad \tag{23}$$

where μ is the gravitational parameter of the central body, and are the orbital radius and inertial velocity of the satellite, respectively, and (**R**) is the gravitational potential of the Earth (without the point-mass contribution) (

Fig. 6 Change in Jacobi constant during a 10-year GEO propagation using point-mass and zonals

– **Gauss–Jackson 8th-order (GJ 8):** a multi-step predictor-corrector method of 8th-order which uses a fixed step size (Jackson 1924; Fox 1984; Berry and Healy 2004). This scheme has been used by US Space Surveillance Centers since the 1960's due to its highly efficient propagation of near-circular orbits (SPADOC Computation Center 1982; Berry and Healy 2004).

In space surveillance and many other applications, we often have an option to sacrifice accuracy for reduced computation time. Thus, we desire an integration scheme which achieves a necessary level of accuracy while minimizing the number of force model evaluations and computation time required. We compare each integrator based on the number of force model evaluations (function calls) that are used to achieve various levels of position error. It is important to remember that this study evaluates integration error only and we are not considering the separate topic of force model errors. Note that the reported number of

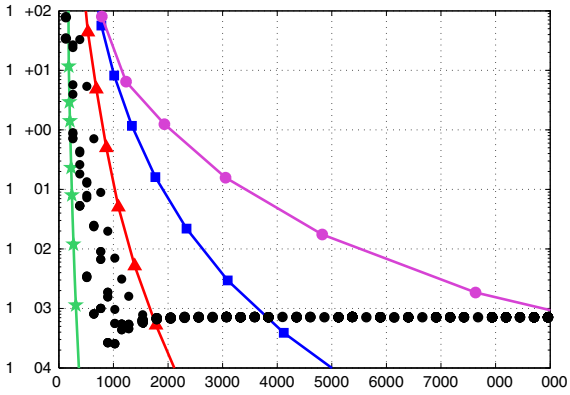

Fig. 7 Comparison of RMS position errors over a 3-orbit GEO (**a**) and LEO (**b**) propagation

Fig. 8 Comparison of low- and high-fidelity force models used in

Table 3 Performance summary of integration methods over three orbit periods

Orbit	Method	Function calls (< 1 m error)	Function calls (< 1 cm error)
LEO	RKF 7(8)	1500	2320
	DOPRI 8(7)	1230	2230
	DOPRI 5(4)	1350	3800
	GJ 8	370	600
	BLC-IRK	640	640
	BLC-IRK (2 processors)	320	320
	BLC-IRK (ideal parallel)	10	10
GEO	RKF 7(8)	1370	2630
	DOPRI 8(7)	815	1300
	DOPRI 5(4)	2050	5520
	GJ 8	210	270
	BLC-IRK	256	512
	BLC-IRK (2 processors)	128	256
	BLC-IRK (ideal parallel)	4	8
MOL	RKF 7(8)	2690	3860
	DOPRI 8(7)	2600	3470
	DOPRI 5(4)	3870	9350
	GJ 8	4930	8610
	BLC-IRK	4608	6140 (3 cm)
	BLC-IRK (2 processors)	2304	3070 (3 cm)
	BLC-IRK (ideal parallel)	72	96 (3 cm)

The approximate number of function calls required to reach a given level of accuracy for each orbit type and each numerical integration technique discussed in this study. Additional entries are given for a 2-processor parallelized and an ideally parallelized BLC-IRK implementation (i.e., 64 processors used and neglecting communication overhead). Note that GJ 8 was restricted to only use 1 iteration (i.e., force model evaluation) per step

operating both schemes in fixed-step mode and uses fixed-point iteration instead of low- and high-fidelity force models. While BLC-IRK and GL-IRK perform quite similarly for GEO orbits, BLC-IRK outperforms GL-IRK in both LEO and highly-eccentric orbits, which is due to the improved node spacing of the BLC-IRK nodes. Similarly, the benefit of BLC-IRK over GL-IRK is enhanced as more nodes are used. Note that all cases in Herman et al. (2013) were restricted to the use of a large number of nodes (i.e., 32 and 200). Figure 1 of this paper implies that GL-IRK and BLC-IRK may not exhibit much of a difference for cases with fewer nodes.

Future work will include developing an efficient step size control algorithm for BLC-IRK. Unlike the embedded ERK methods that exist, no such IRK method has been developed with a second, embedded method, to be used for step size control. However, a few algorithms to control step size for IRK methods do exist. Jones (2012) discusses the implementation of a variable-step algorithm from Houwen and Sommeijer (1990) with a GL-IRK scheme. Jones (2012) demonstrates that the variable-step algorithm, dubbed VGL-ξ, improves upon the fixed-step GJ 8 for highly-eccentric orbits, but recommends that further work be done to improve the efficiency of the algorithm. Aristoff et al. (2014) develops a variable-step GL-

IRK implementation, dubbed VGL-IRK, for orbit and uncertainty propagation and compares its performance against DOPRI 8(7), VGL-ϵ, and MCPI. Aristoff et al. (2014) shows that their VGL-IRK scheme outperforms the other integration methods in LEO, GEO, and highly-eccentric orbits, making their variable-step version of GL-IRK very attractive. As BLC-IRK contains more efficient node spacing than GL-IRK, a variable-step implementation of BLC-IRK should outperform VGL-IRK, in theory. As mentioned, this is an important part of our future work. Other recently developed integration schemes (mainly symplectic) and comparison studies of note include Hubaux et al. (2012), Blanes and Iserles (2012), Blanes et al. (2013), Farrés et al. (2013), Rose and Dullin (2013), and Nguyen-Ba et al. (2013).

4.2 Dense output

All collocation-based IRK schemes have built-in interpolation to evaluate solutions at arbitrary points. This section outlines and examines the dense output capability of the BLC-IRK method. We first describe how to interpolate a solution computed at the quadrature nodes $\{\tau_i\}_{i=1}^{s}$ to an arbitrary time τ. For clarity, we present the necessary equations for the case where the quadrature nodes τ_i lie on the interval $[-1, 1]$, noting that if the time $\tilde{\tau}$ is given in the interval $[\alpha, \beta]$, we can easily rescale it to the interval $[-1, 1]$ as $\tau = (2\tilde{\tau} - \alpha - \beta)/(\beta - \alpha)$. Data

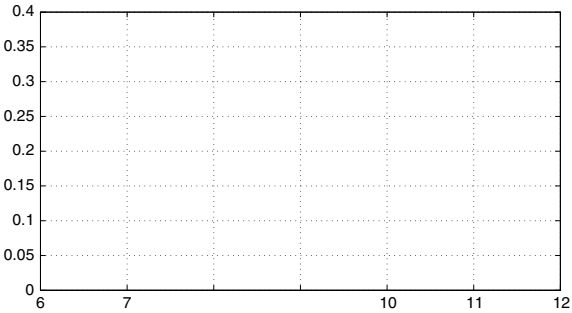

Fig. 10 BLC-IRK collocation interpolation error for a GEO propagation. BLC-IRK propagation performed using 4 intervals/orbit and 64 nodes/interval. Interpolation is performed every 5 seconds. Note that the plotted error is due to both interpolation and integration

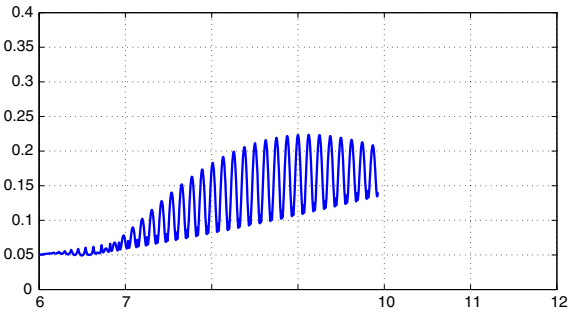

Fig. 11 Interpolation errors of BLC-IRK (**a**) and DOPRI 8(7) (**b**) trajectories using a 5th-order Lagrange scheme. Interpolations are performed every 5 seconds. A relative tolerance of 10^{-13} was used for step size control of the DOPRI 8(7) propagation

BLC-IRK is based, yield node spacing that is more efficient than traditional polynomial-based quadrature methods such as Gauss–Legendre, Gauss–Lobatto, and Chebyshev. This promotes the use of large time intervals and a large number of nodes per interval, reducing the computational load near the clustered endpoints as with polynomial-based quadratures. Additionally, the A-stable property of BLC-IRK makes its use appealing to solving stiff ODEs, including atmospheric entry.

We demonstrated superior performance of BLC-IRK over commonly used ERK methods for near circular orbits while closely matching GJ 8, even when operating in serial mode (no parallelization). Note that the GJ 8 results presented here were done with an implementation that uses one force model evaluation per step only. Ordinary versions of GJ 8 would likely contain iteration, resulting in several force model evaluations at each step. The presented BLC-IRK implementation of using both low- and high-fidelity force models is a major contributor to the efficiency. The specific execution can be tuned for each unique scenario, leaving room for improvement even on the implementation presented here. The low-fidelity model used here is just an example. Deep space and GEO scenarios may benefit from including a rough third-body contribution into the low-fidelity model. It should also be noted that this low-/high-fidelity implementation is applicable to any IRK method. While BLC-IRK is slightly less efficient than GJ 8, 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 its potential. Gauss–Jackson is also neither symplectic nor A-stable. When applicable, parallelization would result in a significant improvement in efficiency over the GJ 8 scheme.

This paper outlined the dense output algorithm for BLC-IRK as well. We demonstrated that interpolating a BLC-IRK trajectory using its collocation algorithm yields a high accuracy, smooth, and continuous solution. We also showed that the accuracy of Lagrange interpolation of a BLC-IRK trajectory is superior to that of a Dormand and Prince 8(7) propagated orbit. This is an appealing aspect of collocation methods, where the higher node density provides a better base for interpolation. This is especially important to conjunction assessment where solutions are required at various points in time along a trajectory.

Appendix: BLC-IRK data files

ASCII-files with quadrature data for $= 24$ nodes (corresponding to bandwidth $= 2.5\pi$) and $= 70$ nodes (corresponding to bandwidth $= 20\pi$) are provided as an online resource. For each bandwidth, we provide six files:

- nodesM.txt: Contains the quadrature nodes $\{\tau\ \}_{=1}$ for the interval $[-1, 1]$.
- weightsM.txt: Contains the quadrature weights $\{w\ \}_{=1}$ for the interval $[-1, 1]$.
- integrationMatrixM.txt: Contains the elements of the integration matrix 16Tf14010

Beylkin, G., Sandberg, K.: ODE solvers using band-limited approximations. J. Comput. Phys. **265**, 156–171 (2014)

Blanes, S., Iserles, A.: Explicit adaptive symplectic integrators for solving Hamiltonian systems. Celest. Mech. Dyn. Astron. **114**, 297–317 (2012)

Blanes, S., Casas, F., Farrés, A., Laskar, J., Makazaga, J., Murua, A.: New families of symplectic splitting methods for numerical integration in dynamical astronomy. Appl. Numer. Math. **68**, 58–72 (2013)

Landau, H.J., Pollak, H.O.: Prolate spheroidal wave functions, Fourier analysis and uncertainty III. Bell Syst. Tech. J. **41**, 1295–1336 (1962)
Lemoine, F., Kenyon, S., Factor, J., Trimmer, R., Pavlis, N., Chinn, D., et al.: The development of the joint